McCON—A GENERAL CONTOURING PROGRAM FOR PERSONAL COMPUTERS

by

John B. Palmerton

Geotechnical Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199

September 1992

Final report

Approved For Public Release; Distribution Is Unlimited

92-30205

Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000
**Title:** McCON—A General Contouring Program for Personal Computers

**Author:** John B. Palmerton

**Performing Organization:** USAEWES, Geotechnical Laboratory, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199

**Supplementary Notes:** This report is available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

**Abstract:** This report includes a description of a set of topographic contouring computer programs which are operational on DOS-based personal computers. A discussion of the mathematical procedure used to generate the contour maps and a detailed User’s Guide is included. The programs were developed so that columnar types of data, often generated by various database software packages, could be directly accessed and manipulated by users. These computer programs also provide for the inclusion of internal data discontinuity boundaries such as geologic faults or groundwater flow barriers. Contour drawing may also be excluded within selected zones. Following the generation of a contouring mesh, profiles (or cross-sections) may be drawn along any plan orientation. The output of the plots may either be sent to the video screen or to a variety of pen plotters.

**Subject Terms:** Contouring, Topography, Personal computers

**Security Classification:** UNCLASSIFIED
Preface

This report describes the concepts and the development of a topographic contouring program which is suitable for use on personal computers. A User's Guide is included along with a diskette which contains the executable program code. The computer program described herein is the result of developments during the course of numerous projects and studies performed over the past few years for a variety of sponsors. Development of this report was funded by Headquarters, US Army Corps of Engineers (HQUSACE) as part of the Civil Works-Materials (Rock) Research and Development Program, under Work Unit No. 31700, "Special Studies for Civil Work Rock Problems." Mr. Jerry S. Huie (CEWES-GS-R), US Army Engineer Waterways Experiment Station (WES) was the Principal Investigator. Mr. Lewis A. Gustafson (CECW-EG) was the Technical Monitor for this study. Publication of this report was funded by the Computer Applications in Geotechnical Engineering (CAGE) project that is sponsored by the Headquarters, US Army Corps of Engineers. Mr. Earl V. Edris, Jr. (CEWES-GS-S) WES, is the CAGE Principal Investigator. The USACE CAGE Technical Monitor is Mr. Art Walz (CECW-EG).

This report was prepared by Mr. John B. Palmerton (CEWES-GS-R) under the direct supervision of Mr. Jerry S. Huie, Chief, Rock Mechanics Branch, and under the general supervision of Dr. Don C. Banks, Chief, Soil and Rock Mechanics Division (S&RMD), and Dr. William F. Marcuson III, Director, Geotechnical Laboratory (GL), WES.

At the time of publication of this report, Director of WES was Dr. Robert W. Whalin. Commander and Deputy Director was COL Leonard G. Hassell, EN.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Uniqueness of Contours</td>
<td>5</td>
</tr>
<tr>
<td>Mesh Generation</td>
<td>6</td>
</tr>
<tr>
<td>Data Input Enhancements</td>
<td>10</td>
</tr>
<tr>
<td>Contour Drawing</td>
<td>12</td>
</tr>
<tr>
<td>Noncontoured Region</td>
<td>15</td>
</tr>
<tr>
<td>Using Successive Shells to Improve Efficiency</td>
<td>17</td>
</tr>
<tr>
<td>Contouring Nonconvex Areas by Composite Fragments</td>
<td>19</td>
</tr>
<tr>
<td>Handling Discontinuity or &quot;Fault&quot; Boundaries</td>
<td>21</td>
</tr>
<tr>
<td>Profiles</td>
<td>22</td>
</tr>
<tr>
<td>Appendix B: FORTRAN Listing for McCON</td>
<td>B1</td>
</tr>
<tr>
<td>Appendix C: FORTRAN Listing for McGRID</td>
<td>C1</td>
</tr>
<tr>
<td>Executable Programs on Diskette</td>
<td></td>
</tr>
</tbody>
</table>
McCON - A GENERAL CONTOURING PROGRAM FOR PERSONAL COMPUTERS

Introduction

1. The need to prepare contour maps and profiles along cross-sections often arises during the course of engineering and geologic investigations. Although topographic applications are the most widely known, contour maps and plots are prevalent for stress analysis, groundwater analysis, and for almost any type of data for which trends are to be examined, estimated, or predicted. Over the past twenty years, numerous computer programs have been developed on numerous electronic platforms to aid in the preparation of contour maps. The user of any of the various programs (including the one described herein) is cautioned to be aware of the application for which the program was written and to be aware of the methods employed to estimate the contourable values in the vicinity of the given data sets. The recent advent of high quality graphics capabilities on personal computers and workstations has resulted in the availability of a large number of contouring programs at reasonable cost from a variety of suppliers. The program McCON was developed such that the "raw" data could be obtained either from self-prepared data files or from data files commonly produced by other computerized applications. It was also developed to permit the inclusion of zones across which the data is discontinuous and to permit the preparation of profiles along cross-sections at any orientation.

2. McCON is a general purpose contouring program which executes rapidly on IBM compatible personal computers. The minimal data needed to execute the program is a free field data file of the location coordinates \((x,y)\) and the contour values \((z)\) of the data points. As configured, the program will accept up to 999 data \((x,y,z)\) triplets. The data points do not need to follow any regular spacing. Negative values of \(x, y,\) or \(z\) are permitted and the \((x,y)\) coordinates may have an arbitrary range of values. The coordinate ranges are internally examined and scaled to yield a contour plot which fills the video screen.

3. Various Options are provided by McCON:
   a. A noncontoured subregion (i.e., an area within which the user does not wish contour lines to be drawn) may be specified provided that this region is of a convex shape.
   b. Contouring may be restricted to be entirely contained within a
given convex subregion.

c. Contouring of areas having physical boundaries that are not convex may be prepared by subdividing the overall area into convex "composite" regions.

d. Internal boundaries across which contours are discontinuous (e.g., geologic fault lines, seepage barriers, etc.) may be included.

e. "Templates" (sets of isolated or connected lines) to be drawn onto the contour plot to enhance the artwork of the drawing are included.

f. Profiles, or cross-sections, along any plan orientation may be created and separately plotted.

4. McCON requires 590K of RAM memory for the executable program. The source language is Microsoft (R) FORTRAN. MicroGCS (the graphics compatibility system developed by the US Army Corps of Engineers) is used to support the graphics. All calls to subroutines beginning with the letter "U" in the accompanying source listing (Appendix B) are MicroGCS subroutines. The graphical output may also (by option) be sent to a variety of pen-type flatbed plotters via routines provided by MicroGCS.

5. The program generates nonintersecting triangles which connect each and every data (node) point. Triangle generation ceases when all of the node points are connected to at least one triangle and the resulting mesh of triangles encompasses all of the node points in a convex fashion (i.e., the outer edges of the triangle mesh form a convex shape). The resulting mesh will contain no areas that are not included within a triangle (i.e., the mesh will contain no "holes"). A scheme was developed to create the mesh in a one-pass "spawning" process; therefore, as each triangle is created, sufficient information exists to immediately draw the contours within the new triangle. Because of this one-pass triangle generation process, the execution time is quite rapid. Typically, a set of 100 nodes (on an 82386 machine with math coprocessor and EGA card) will require approximately 10 sec to generate the triangle mesh; a set of 400 nodes, 56 sec; and 900 nodes, 165 sec. The time devoted to contour line drawing is dependent upon the resulting number of contour lines to be drawn. A relatively heavy set of contour lines may require 30-45 sec of execution time for the contours to be drawn on the screen; however, the execution time is not significantly related to the
complexity of the contours nor to the spatial distribution of the nodes. Considerably more time for drawing is required if the output is sent to a plotter.

6. The contour lines are drawn as a series of connecting straight line segments and circular segments. This combination gives an aesthetically pleasing appearance to the resulting contour plot.

Uniqueness of Contours

7. The process of defining the locus of points of equal contour value (the contour line) is always dependent upon the manner in which the desired contour values are established (i.e., interpolated) between the given data points. Figure 1 shows two differing manual interpretations as regards contouring the elevations given at 4 points. Figure 1a shows the result obtained by interpolating the elevation data along the right descending dashed line; whereas Figure 1b is the result of interpolation along a right ascending line. It is obvious that the two schemes give radically different interpretations.

8. However, neither interpretation for establishing triangles is inherently superior. In this case, more data are required to properly establish the character (or trend) of the contours. If it were established that the elevation at the mid-point was equal to 40, then the interpretation shown in Figure 1a would be proper. Figure 1b would be applicable if the mid-point elevation was equal to 65. Resorting to polynomial, power, or spline fits does not necessarily improve the interpretation; each technique would only lead to a different estimate. All contouring techniques rely on estimating the behavior of the contourable value within the neighborhood of a few (typically 3 or 4) given field values. Within that neighborhood, many interpretive schemes are possible. However each scheme implies a knowledge of the character of the "landform." The interpolation scheme used for McCON is quite similar to the manual interpolation methods discussed above (i.e., a linear interpolation of the contour value gradient from node to node along triangle edges). The way and order in which the triangles are generated may influence the contour plot. In any case, one interpretation is as proper as any other in light of the lack of other data. There is no good reason to
a. Result obtained by interpolating elevation data along dashed lines

b. Result of interpolation along a different line

Figure 1. Converting elevations to contours

state that an alternate interpretation is better unless additional information is known.

Mesh Generation

9. The scheme to generate the triangle mesh is summarized as follows:
   a. Create a single "seed" triangle from any three suitable node points (i.e., the seed triangle must not contain any other node points within its boundaries).
   b. Spawn additional triangles from each edge of the seed triangle such that the spawning and the spawned triangle share the common edge. The criteria for determining which node point to use to create the new triangle are given below.
   c. In the order in which they were created, use the previously spawned triangles to spawn new triangles. After each triangle is spawned, all of the previously spawned triangles are examined to see if the new triangle is sharing an edge other than the edge of the parent triangle. Each newly spawned triangle will, at most, form two new edges (the third edge is already shared with its parent) from which to subsequently spawn additional triangles. The new triangle may furnish only one (or no) new edge for further spawning since the new triangle may also share edges with previously
generated triangles. As triangles are spawned, a spawning sequence number is assigned. Triangles may be spawned only from edges that are not already shared by two triangles.

d. Step c is repeated until after some spawn (which will be the last spawn) it is detected that the order-of-sequence number of the spawning triangle is equal to the total number of generated triangles. When this occurs, the mesh of generated triangles will connect and encompass all of the node points. The result will be a triangle mesh with its outer edges forming a convex shape. Spawning will continue as long as the order-of-sequence number of the spawning triangle is less than the number of generated triangles.

10. Figure 2 shows the order in which the 10 triangles connecting the 9 node points were spawned. Triangle 1 (the seed triangle) spawned triangles 2, 3, and 4. Triangle 2 spawned triangles 5 and 6. Triangle 3 spawned triangles 7 and 8. (Triangle 7 also shares another of its edges with the by then existing triangle 4.) Triangle 4 only spawned triangle 9 since triangle 7 had already been spawned by triangle 3. Triangle 5 spawned triangle 10. Triangles 6, 7, 8, 9, and 10 spawned nothing. The outer edge of the mesh (described by node points 2, 6, 8, 7, 9, and 4) is convex.

11. All triangles (except the seed triangle) are spawned by an edge of a parent triangle. The parent triangle provides two vertices for the next formed triangle. There are three criteria to decide which node to select as the third vertex of the new triangle.

**CRITERION A.** Every node point which lies on the side of the spawning edge which is opposite to the side containing the remaining vertex of the spawning triangle becomes a candidate node.

This criterion prevents new triangles from intersecting (overlapping) their parent. As triangles are formed, the node point numbers of the vertices of the triangle are stored in an ordered array. This array is ordered such that the vertices are listed in a counterclockwise fashion about the triangle. For example, referring to Figure 2, the vertex array for triangle 1 must be {3,1,5} (or {1,5,3} or {5,3,1}) since these numbers go counterclockwise about the triangle. Similarly, triangle 8 is described as {5,4,9} or {9,5,4} or {4,9,5}. The formula for the area, A, of a triangle in terms of its vertex coordinates is:
A = \{ x_1 (y_2-y_3) + x_2 (y_3-y_1) + x_3 (y_1-y_2) \} / 2

or symbolically,

\[ A = \text{Area}(1,2,3) \]

where \( x_1 \) and \( y_1 \) are the coordinates of vertex 1, etc. This formula yields a positive value if the points 1,2,3 are ordered in a counterclockwise sense about the triangle and a negative value if ordered clockwise. Therefore, knowing the fashion in which the spawning edge is ordered (say 3,5 for triangle 1) only those nodes which yield a positive (counterclockwise) area (i.e. Area (3,5,6), Area (3,5,8), or Area (3,5,7)) are on the proper side of spawning edge 3,5. All other nodes lie on the same side of the spawning line as the spawning triangles remaining vertex (node 1) and the application of the area formula gives negative or zero areas for these nodes. Equation 1 is applied often during the course of mesh generation. A mesh formed by 250 nodes typically will require over 60,000 area calculations; 999 nodes, over 225,000 calculations.

12. For each node point, if any, that passes Criterion A, the following criterion must be met.

**CRITERION B.** The angle at the vertex of the "trial" new triangle which is opposite the spawning edge must be the largest angle formed by the set of candidate nodes from Criterion A. (Provided that the candidate node also passes Criterion C below.)

This criterion will ensure that no other node points may lie within the new triangle. Referring to Figure 2, presume that triangle 1 is attempting to spawn a triangle from edge 3,5. In order, the candidate nodes would be 6,7, and 8. (Node 9 is rejected since Area (3,5,9) is zero.) Node point 8 supplies the largest vertex angle. If it is imagined that some additional node point lies within triangle 2, it is obvious (for either acute or obtuse triangles) that the imaginary point subtends a larger angle than the angle 3-8-5 and, as such, would be a better choice to spawn a triangle.

**CRITERION C.** Reject all candidate nodes that yield a negative (clockwise) area (per the formula given in Criterion A) when associated with any line previously emanating from the first end of the spawning edge and reject those candidate nodes which yield a positive area when
associated with any line emanating from the second end of the spawning edge.

The area formula (Equation 1) is applied such that \(x_1, y_1\) is the coordinate of the spawning edge endpoint under consideration, \(x_2, y_2\) is the coordinate of the node point at the end of the line(s) emanating from the end of the spawning edge, and \(x_3, y_3\) is the coordinate of the candidate node point. For the purposes of applying this criterion, it is only necessary to perform checks on lines which emanate above the horizon of the spawning edge. The purpose of this criterion is to ensure that no triangles intersect each other.

13. To illustrate Criterion C, examine the conditions shown in Figure 3. Suppose that the triangles shown by the solid lines were created in some proper fashion and that the calculations have proceeded to the point where side 1-2 of triangle P is attempting to spawn. According to Criterion A, nodes \(a, b, c, d, e,\) and \(g\) are all possible candidates because they lie above the 1-2 "horizon" (i.e., Area (2,1,?) is positive for all labeled nodes except \(f\)). By inspection, it is seen that only choices \(a, e,\) or \(g\) would result in a non-intersecting new triangle. However, applying Criterion B, the maximum apex angle would occur for node \(c\) (and the next best angle at node \(b\)). Criterion C will reject the candidacy of nodes \(b, c,\) and \(d\) because they form clockwise ordered triangles with the line emanating from node 1 to node \(a\). That is, Area (1,a,b), Area (1,a,c), and Area (1,a,d) are all ordered.

Figure 2. Spawning triangles

Figure 3. Criterion C illustration
clockwise. Nodes g and e are not rejected since Area (l,a,g) and Area (l,a,e) are ordered counterclockwise. Area (l,a,a) yields an area of zero (which, for the purposes of Criterion C, is non-negative) and is retained as a candidate node. For the example shown, the line emanating from the opposite end of the spawning edge, line 2-g does not cause any rejections because all node points under consideration give clockwise triangles in accordance with the second half of Criterion C. Therefore, node a (which yields the maximum vertex angle) will become the correct choice to spawn a new triangle. During the course of the calculations a linked list of all of the node points which are connected (via an edge of a triangle) to a given node is maintained. This list provides the means for applying Criterion C.

14. The node point which passes all of the criteria (and there is at most one node that can pass) is used to form a new triangle. In the event that no node passes the criteria, no triangle is spawned from the edge under consideration and the calculations proceed to the next edge to be considered. Since Criterion B guarantees that no nodes may be trapped within a triangle, then each node must be used to form at least one triangle simply because all nodes must lie outside at least one of the edges of every triangle (except the nodes which form that given triangle). Therefore, the spawning process cannot stop until all nodes have been used. The completed mesh of triangles must form a unique convex shape about all of the nodes since this is the only shape for which no other nodes will be available for spawning. Voids, or interior areas not included within a triangle, are not possible since the void would be adjacent to two or more triangles with unshared edges. During the spawning process these edges would have had an opportunity to spawn. The criteria for spawning require that the spawning edge always spawns unless no node exists outside of that edge or if the spawn would result in an intersecting triangle.

**Data Input Enhancements**

15. McCON provides several means to "massage" the input data. (Appendix A gives a more complete discussion of the capabilities for data enhancement.) In general, the input data file may consist of up to 20 columns of data values. The user identifies the columns which contain the x and y coordinates and the z (contourable) value. The z value may be the sum or difference of any of the columns of the data file. Therefore, a data file may
contain sufficient information to yield many contour plots.

16. The data input activity of McCON permits the user to identify the elements of the original input data file and to enhance the data in the following ways:

a. "As is." Contouring of the original data points will take place without any data smoothing and without any "border" data point creation.

b. "Original points with boundary." Additional data points are created which form a rectangular boundary around the original points. The contourable values (elevations) of the boundary points are computed by Equation 2 (given below).

c. "Grid points with or without apron." A grid which overlays the original data points is created. The contourable values (elevations) of the grid points are computed by Equation 2 (given below). The original data points are optionally superimposed on the grid points.

d. "Triple triangle refinement." Following the generation of a mesh of triangles, an option will subdivide each "old" mesh triangle into three sub-triangles. The contourable value (elevation) at the added point, (mid-point of the old triangle), may be computed by Equation 2 or by linear interpolation.

The last three data enhancements listed above will generally result in a smoothing of the contour plots; however, increased contour lines "waviness" may result. The contourable values (elevations) at "created" points are computed by the following equation for inverse power distance averaging:

\[ Z_i = \frac{\sum_j Z_j / (d_{ij})^Q}{\sum_j (1 / (d_{ij})^Q)} \quad \text{for all } d_{ij} \neq 0 \]

\[ Z_i = Z_j \quad \text{for any } d_{ij} = 0 \]

where \( Z_i \) = the elevation at the created point(s) \( i \),

\( Z_j \) = the elevations at the given original points, \( j \),

\( d_{ij} \) = the distance between point \( i \) and \( j \),

\( Q \) = an exponent to be chosen (usually 2).

Equation 2 will give a distance weighted average for the elevations to be assigned. Gridding, which always requires inverse power distance averaging,
will generally result in contour lines that are more circular in shape than "as is" contours and which often isolate local highs and lows with closed contours. An excessive number of grid points may result in undue contour line waviness. If no gridding is used, the contours generally exhibit more linear trends. "Triple triangle refinement" provides the only method for input data enhancement for "as is" contouring. Again, inverse power distance averaging for "triple triangles" may result in many closed contours (isolation of local highs and lows). Linear averaging for "triple triangles" is generally recommended.

17. The results from various data input manipulations are shown in Figure 4. The location and contourable value (elevation) of 32 nodes is shown in Figure 4a. The "as is" contour plot is shown in Figure 4b. The results of "gridding" the data and not including the "original" data points are shown in Figure 4c. Notice that the contours are more circular. When the original data points for "gridded" data are honored, the results are shown in Figure 4d. "Triple triangle" plots are shown in Figure 4e and in Figure 4f.

**Contour Drawing**

18. Since the spawning process outlined above is a single-pass process, the contour lines can be drawn as the triangles are spawned. Figure 5 shows a triangle being spawned. The numbers at the nodes represent contourable value (or elevation). Presume that it is required to draw the 10 and 20 contours. Regardless of the particular values of the elevations at the nodes surrounding any triangle, any given elevation for which a line is to be drawn either does not intersect the triangle at all or it intersects two sides of the triangle. (The special case where two nodes of the triangle have equal values results in the contour line being that triangle’s edge.) The contour lines are drawn as follows as the triangles are spawned.

a. In the order of all contour lines to be drawn, determine whether the line will pass between the ends of the shared edge (as do the 10 and 20 contour lines). If not, do not draw lines; if so, go to the next step.

b. Compute (by linear interpolation) the locations where the contour line intersects the shared edge and also where it intersects the second edge of both the spawning and the spawned
Figure 4. Contours resulting from various input data enhancements
Mathematically construct two lines connecting these points. Then compute the location of the mid-point (designated by the "+" symbols in Figure 5) of each line in each triangle. If the contour line is to be drawn with "sharp" corners, the two lines going from the mid-points to the shared edge are the contour lines to be drawn on the screen (or plotter). The other "half" of the contour line will not be drawn until a triangle is spawned sharing the now "unshared" edge. The reason for leaving half of the triangle unfinished is to accommodate the introduction of circular contour line segments which will smooth the contour lines. In the event that an unfinished side does not share an edge, (as would be the case for all triangles adjacent to the outer convex edge of the mesh) the program will, after all spawning is complete, seek all triangle edges that are unshared and finish drawing the contour lines out to the mesh boundary.

19. Circular segments (usually desired) for the contour lines are drawn in the vicinity of the shared edge. Figure 6 shows a detail of the 10 contour line. The points o, a, and b are the same points as shown in Figure 5. All circular segments are constructed by calculating the radius and center coordinates of the circle S that will be tangent to the lines oa and ob at a distance of q from the point o. The program initializes q to a value of 25 screen units (the width of the entire contour plotting area is 1040 screen units) and does not permit the value of q to exceed 25 screen units. The variable q is reduced to d (where d is the length of the shorter of oa or ob) if d is less than 25 screen units. The distance q is also reduced sufficiently to assure that the circle to be drawn crosses the shared edge. Figure 7 illustrates a situation where the contour line would, without a reduction of
Figure 6. Circularly segmented contours.

Figure 7. Nonpermitted circular segment.

The edges of the triangles which form the convex outer shell of the noncontoured region is maintained (i.e., a list of edges that are not shared) for later use. No contour line drawing is done during this stage. Stage two is accomplished by allowing all of the remaining nodes to be used for spawning additional triangles. The spawning is renewed by beginning with the first triangle (the seed triangle) created at stage one.
and stepping through each triangle in order of its creation. No new triangles will be spawned until a triangle with an unshared edge is reached (i.e. the triangles along the noncontoured boundary). The new triangles (which will all be outside the convex shell from stage one) will then spawn other triangles to encompass all of the nodes. Contour lines are drawn during stage two. Following the spawning of all the triangles, edge contours are drawn within triangles having an unshared edge. The unshared edges exist for all triangles along the second stage outer convex shell and the convex shell from stage one.

21. The user must specify the nodes that describe the noncontoured region. Regardless of the location of the specified nodes, a first stage convex shell will be formed which is composed of only these nodes. Referring to Figure 8, suppose the noncontoured region is specified by nodes 5, 6, 2, 9, and 7. (These nodes were identified during the course of program execution by repeatedly positioning the screen "cursor" in the vicinity of each specified node on the noncontoured boundary.) Stage one mesh generation will form triangles using only the nodes specified as describing the noncontoured region plus any nodes lying on the lines which connect the specified nodes (e.g., node 8 lies on line 7-5).

22. The specification of a noncontoured region will not yield the same result as simply not drawing (or erasing) the contour lines that would otherwise be within the region. The data points within the noncontoured region are not used for any purpose; therefore, they are not used to form mesh triangles or for contour interpolation.

23. An example of a contour plot with a noncontoured region is shown in Figure 9. Six points were required to define the region.

![Figure 8. Describing a non-contoured region](image1)
![Figure 9. Example of a non-contoured region](image2)
Using Successive Shells to Improve Efficiency

24. McCON was originally written for mainframe processing. After the program was installed on a personal computer, it was found that excessive execution time occurred for problems with large numbers of nodes. For example, approximately 30 min (1800 sec) was required to generate a mesh for 400 nodes. This execution time was reduced to 56 sec by using the successive shell procedure to be discussed. The reason for the excessive execution time was the application of the three criteria to choose which node to use when spawning triangles. As each triangle is spawned, all nodes must be evaluated as candidates for each new triangle. This will cause the execution time to increase as the square of the number of nodes. In addition, following each successful spawn, it is necessary to check each and every previous triangle to see if the new triangle is sharing an edge other than the edge of the parent triangle. The time for this check also increases as the square of the number of generated triangles. Therefore, a procedure to reduce the number of nodes to be considered as candidate nodes and to reduce the number of triangles to be compared after a new spawn would hold the promise of substantially improving execution time.

25. The procedure of successive shells is basically an extension of the concepts employed to accommodate noncontoured regions. Imagine some (any) closed convex shape to be superimposed onto a set of nodes. If only the nodes which lie within or on this convex figure are used to form a triangle mesh, it is assured that the convex shell around the resulting mesh will be entirely enclosed by the imagined convex shape. It is also assured that none of the remaining nodes will lie between these two convex boundaries. Then imagine another convex shape that surrounds the first to be superimposed over the nodes. If triangles are then spawned from the first shell (exactly as was done for expanding the mesh about a noncontoured region) the result will be a second convex shell that is encompassed by the newly imagined convex shape. This procedure may be repeated with successively larger imagined convex shapes until all of the nodes are encompassed. Figure 10 illustrates the concept. The triangles labeled "a" were spawned within boundary 1. The heavy line is triangle shell #1. The triangles labeled "b" were spawned from shell #1 and produced shell #2. It is obvious that the procedure may be repeated to include all nodes.
26. McCON is programmed to use the procedure of successive shells whenever the total number of node points is in excess of 180. (A mesh formed with 180 nodes requires approximately 45 sec for triangle generation.) Concentric circles are used as the (imagined) convex shape to build successive shells. The centers of these concentric circles are located at the mid-point of the minimum and maximum coordinates of the nodes. The radius of each successive shell is determined by requiring that at least a certain number (currently set to 60) of nodes are contained within each shell. If a non-contoured region is also specified, the center of the concentric circles is located at the mid-point of the nodes specifying the noncontoured region; the distance from the center to the furthest specified node on the noncontoured boundary is computed, and the first successive shell contains all of the nodes lying within that computed distance from the center.

27. The procedure of successive shells gave a tremendous reduction in execution time for two reasons. First, the number of nodes to be examined as candidate nodes was reduced to approximately 60. Second, it was no longer necessary to examine every previously spawned triangle to check for shared edges. Since each successive shell is convex, it is assured that as triangles are spawned within any given shell, that edges can only be shared with triangles within or outside of the previous shell. The successive shell procedure resulted in an execution time that is proportional to the number of nodes rather than to the square of the number of nodes.

28. The successive shell procedure does have the disadvantage that "peculiar" triangles may be generated at the shell boundaries. This may result in some unwarranted "waviness" in contour line drawing. Therefore, in the event that the contour plots are being directed to a plotter, the number
of nodes per spawning cell is set such that it never exceeds one-half of the number of nodes available for triangle generation. This increase in the number of nodes per cell will lead to an increase in execution time. However, since most plotters draw so slowly (compared with drawing lines on the screen), the execution time increase will not be particularly noticeable. Contour plots sent to a plotter will (for large numbers of nodes) generally be of a higher quality than screen plots.

Contouring Nonconvex Areas By Composite Fragments

29. The triangle mesh generation criteria will always result in the data points being surrounded by a unique convex boundary. This boundary is generally adequate for topographical applications. However, if contour plots are desired for applications such as stress analysis, the regions in which meaningful contours may be drawn are often limited by the physical shape of the object being stressed. These physical regions often do not yield a convex shape which conforms to the physical boundaries. Any nonconvex shape may be subdivided into two or more shapes which are convex. McCON provides an option to restrict contour drawing to an internal convex region. This option is similar, but opposite, to the option to prevent drawing in a noncontoured area.

30. Figure 11a shows the contours of the shearing stresses within an embankment and its foundation. (The small crosses or tics on the drawing represent the location of the data points. The outline of the embankment and foundation was drawn using the "template" option). Contour lines are drawn outside the embankment boundaries because the generated triangle mesh (shown in Figure 11b), which must result in a convex shape which encompasses all of the data points, contains triangles beyond the physical boundaries. It is obvious that the contour lines drawn through "space" are not desirable. Figure 11c shows the contours within the foundation. These contours were created after restricting contour drawing to the convex region formed only by the data points within the foundation. (The use of the option to restrict contouring to a specific region required that the user, by use of the movable screen "cursor", identify the four data points at the corners of the foundation.) The triangle mesh is shown in Figure 11d. Figure 11e shows the contours which resulted from restricting contour drawing to be entirely within
a. Noncomposite contours

c. Lower composite

e. Upper composite

g. Combined composites

b. Generated triangle mesh

d. Triangle mesh for foundation portion

f. Triangle mesh for embankment portion

h. Composite mesh

Figure 11. Preparing a "composite" contour plot
the upper portion. The triangle mesh for the embankment portion is shown in Figure 11f. A composite drawing, as shown in Figure 11g, may be produced by first contouring the lower portion and then, without removing the paper from the plotter, contouring the upper portion.

Handling Discontinuity Or "Fault" Boundaries

31. The preceding discussion demonstrates that the parent region, which is defined by all of the data points, may be broken into any number of convex regions simply by identifying which internal data points are along convex boundaries of the subregions. A composite contour drawing may then be produced. McCON also provides a variation of the above option for specifying the location of connected lines which indicate where a contour discontinuity is expected. For example, if the trend of a geologic "fault" is known, then it would be expected that the elevation of the top of a given formational unit (e.g. top of limestone) would be discontinuous across the "fault" (i.e., contour elevations should not be interpolated between data points that are on opposite sides of the "fault"). The use of this option requires that a data file be previously prepared. This data file must contain x,y pairs (one pair per data file line) which define the "fault" location. If there are more than two x,y pairs (more than one line), then the lines which connect the x,y pairs (in the order given) must form a convex shape always turning in the right-hand direction. (The "fault" line may not have "switchbacks.")

32. Additional nodes (which are displayed on the screen) are created at regular intervals along each "fault" line segment. The contour plot may then be created as a "composite" plot. First, invoke the option that contours are restricted to be within a convex area. Only the nodes situated on the right-hand side of the x,y pairs (from the "fault" data file) will be used to define the contour plot. The elevations of the additionally created nodes along the "fault" line(s) are computed using Equation 2 (with the Z restricted to points within the contoured area). Then, a second contour plot is prepared by invoking the "noncontoured region" option. The non-contoured area will consist of the nodes on the right-hand side of the "fault." The elevations of the additional nodes are computed with the Z restricted to points outside the non-contoured area.
33. Figure 12 contains a series of drawings which demonstrate contouring in the vicinity of a "discontinuity" boundary. The data represent water table elevations on either side of a vertical impermeable boundary. Since water cannot flow across an impermeable boundary, interpolation of elevations across the boundary should be avoided. Figure 12a shows "all" of the nodes used to form the triangle mesh (original plus "created" nodes). The solid, bent line (which was drawn via the "template" option) is located along the impermeable boundary. Figure 12b shows the contour plot which would result if no "fault" boundary was prescribed. The contour lines are continuous across the "fault" boundary. Figure 12c shows the triangle mesh which resulted from not including the "fault." The contour lines are discontinuous at the "fault" boundary, but are continuous beyond the ends of the "fault" line as shown in Figure 12d. The mesh for the "fault" boundary case is shown in Figure 12e.

Profiles

34. After a triangular mesh has been formed, it is a relatively simple process to create an "elevation" profile along any section of the mesh. The x,y location coordinates and the z value (elevation) for the vertices of each triangle is then known. After the user specifies the beginning and ending locations of the ends of the profile section, McCON computes the x,y location of numerous (one location every 5 screen units) profile points along the section. Equation 1 is applied to locate the triangle in which the profile point lies. The elevation, z, to assign to each profile point is computed by assuming that the elevation of any point within any given triangle is given by:

\[ z = ax + by + c \]  

(3)

where the constants a,b, and c are found by substitution of the known values of x,y, and z at the three vertices of the triangle in which the profile point lies. The profile drawings may either be superimposed directly on the contour plots as shown in Figure 13a, or the (same) profile may be produced as a separate drawing as shown in Figure 13b. Several profiles may be produced at one time via the "GRID" option of McCON. Examples of multiple profiles are shown in Figure 14a and Figure 14b.
Figure 12. Contours across a "fault" boundary
35. The user may also create a "mesh" file following completion of a triangle mesh. The "mesh" file may be used to create additional contour plots and profiles without having to again go through the process of mesh generation. The user's guide (Appendix A) gives complete instructions for the use and application of McCON.
a. Horizontal profiles

b. Vertical profiles

Figure 14. Multiple profile plots
APPENDIX A - USER'S GUIDE FOR MCCON

A floppy diskette containing the executable (MICRO1.EXE and MCGRID.EXE) and Fortran source language (MICROI.FOR, MICRO2.FOR, MICRO3.FOR, MICRO4.FOR, and MCGRID.FOR) programs is attached to the inside back cover of this report. A batch file (MCCON.BAT) which calls the programs into execution and a demonstration data file (MCDEMO.DAT) is also included. To utilize the diskette, follow these steps:

1. At the C: prompt create a new directory (e.g., enter MD\CONTOUR).
2. Get into the new directory (i.e., enter CD\CONTOUR).
3. Insert the diskette into the floppy drive (presumably the A drive).
4. Enter COPY A:*.*
5. Enter INSTALL (the programs will be installed on the hard drive).
6. Remove the diskette and return it to the storage envelope.
7. Enter MCCON to execute the programs.

Note: The demonstration data file, MCDEMO.DAT, may be executed by simply entering a carriage return at all program prompts (with the exception of entering a "Y" when requested at the outset of the actual contouring activity).

The contouring programs are brought into execution by entering "MCCON". The MCCON command first issues the command, MCGRID, which requests the name of the (separately prepared) data file which contains the pertinent information and performs certain enhancements, if requested, on the data. A file named MAP.DAT is always created upon the exit from MCGRID. The MCCON command then issues the command MICRO1 which accesses the MAP.DAT file and performs the actual contouring functions. You may (at the C: prompt) enter MCGRID (to prepare a new MAP.DAT file) or MICRO1 (to operate on the current MAP.DAT file) directly from the keyboard if desired.

All of the program queries listed below may be answered by a default (DEF) response, (i.e., simply by a carriage return). The default for ALL Y/N queries is N (no). The following instructions give the default response, where applicable, for other than Y/N queries. Audible beeps issued by the program indicate that a carriage return must be entered to cause the computations to proceed.

The numbers in parentheses (e.g. (1.01)) refer to the questions asked during execution of the contouring programs.

0 DATA INPUT ACTIVITY..(1.01 THRU 1.16)...

(1.01) THRU (1.06) RELATE TO THE MANNER IN WHICH THE INPUT DATA FILE IS TO BE INTERPRETED. THE MCGRID PROGRAM ISSUES THESE PROMPTS.

(1.01a) ENTER (ORIGINAL) DATA FILE NAME (DEF=MCDEMO.DAT): Execution of the programs requires that an ASCII data file containing the x and y coordinates and the z (elevation) values of each of the points be separately prepared.
This file may be created via an editor or by another program. Each line of
this file must contain (as a minimum) one set of x, y, and z. The x and y
coordinates may be given in a random order. Multiple z values may be included
on a line. The order of the x, y, and z values is arbitrary; however, they must
be in the same order on each line within a given file. The default response
will access an example data file (MCDEMO.DAT). This file may be treated
either as a "free-field" file or as a "columnar" file (see (1.01b)).

(1.01b) IS FILE "FREE-FIELD" OR "COLUMNAR" (F(DEF) OR C ?:
The data file
structure may either be "free-field" or "columnar". If the file is
"columnar", the data columns must be "right-justified". The program will ex-
amine the input data file and produce a series of "X's" to denote the columns
as shown in the following example.

(1) (2) (3) (4) (5)
XXXXXXXX XXXXXX XXXXX XXXXXX XXXXXXX (computer made heading)
1043.2 653.4 198.4 156.43 234.54 (1st line in data file)
986.3 1785.3 67.4 1245.52 (2nd line in data file)
e tc. .. ... ... ... ... (etc. .. .. )

Presume that Column (1) is x, Column (2) is y, and Columns (3), (4), and (5) are
z values. If a value is missing (e.g. 2nd line Column (4)), then any contour-
ing operation (see (1.07) thru (1.13)) which involves such a missing value
will cause the entire line to be ignored (i.e. the information in the line
will be used provided that the operations do not involve a missing field).

"Free-field" files are characterized by separating the numbers by one or more
spaces or by commas or by /'s. The sample data file above, could, in free-
field format be represented as:

1043.2,653.2, 198.4, 156.43,234.54
906.3,1785.3,67.4,,1245.52

or by

1043.2 653.2 198.4 156.43 234.54
906.3 1785.3 67.4,,1245.52

If any data element is missing in a free-field file (signified by (,,)) then
the element is interpreted as "0.0"; - not as a "missing" bit of information.
ALL data in a "free-field" data file MUST BE NUMERIC.

A "columnar" data file is recommended since this form provides for data
manipulations that are not possible with a "free-field" file.

(1.02) WANT INVERSE POWER GRIDDING ? (Y/N):
This option is provided so that
the (random) input data points may be manipulated to create a regular rectan-
gular grid of data points. A 13X13 (15X15 if the border described under
(1.05) is retained) point grid is created. The z values (or elevations) of
the grid points are computed by the formula:
\[ Z_i = \frac{\sum Z_j / (d_{ij})^Q}{\sum 1 / (d_{ij})^Q} \text{ for all } d_{ij} \neq 0 \]

\[ Z_i = Z_j \text{ for any } d_{ij} = 0 \]

where \( Z_i \) is elevation at the grid point \( i \), 
\( Z_j \) are the elevations at the given original points, \( j \), 
\( d_{ij} \) is the distance between point \( i \) and \( j \),
and \( Q \) is an exponent chosen by you (usually 2).

The effect of imposing inverse power gridding is to smooth the data. This option should probably be used for cases where the original data points tend to cluster. If good spatial coverage exists, this option is not recommended. Also see (2.32) below for an alternate method for data smoothing.

1.03 WANT ORIGINAL DATA POINTS "AS IS"? (NO FILLING OUT TO EDGE OF PLOT) (Y/N): The default (N) will cause a rectangular border (which just surrounds the original data points) of points to be created. The elevations (z) of the points that are created and lie on the border are computed as discussed in (1.02) above. Generally, the addition of these fictitious points will cause a more pleasant appearance to the contour plot. A "Y" response will result in leaving the data points "as is", without any border enhancements. The query for this option is not in effect when inverse power gridding is invoked.

1.04 ENTER VALUE FOR Q (POWER FOR INVERSE FITTING) DEF=2: The Q value defined under (1.02) is to be supplied to this query. This query is invoked only when inverse power gridding is selected. Higher Q values will cause the elevation at a grid point to be more closely related to its nearest (original data point) neighbor. The closer neighbors should indeed have more weight in determining the assignment of a grid point elevation, but within reasonable limits; therefore a good value to select is "2" (the default response).

1.05 REMOVE APRON FROM AROUND DATA POINTS? (Y/N): The default (N) will cause a rectangular apron to be placed around the original (or, when invoked, inverse power gridded) border (1.03). No original data points will then lie on the extreme edge of the contour plot. The elevations of the fictitious border points are computed as described under (1.02). A "Y" response will cause the border to NOT be generated.

1.06 DO NOT HONOR ORIGINAL DATA POINTS? (Y/N): The default (N), will, when inverse power gridding is selected, retain the original given data points in addition to the gridded points. The default response is usually recommended since inverse power gridding tends to "chop-off" the peaks and valleys. A "Y" response will not add the original data points to the gridded points (i.e., the original data points are deleted).
The following discussion refers to the sample "columnar" data file below.

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
</tr>
<tr>
<td>24-A1</td>
<td>6226</td>
<td>3756</td>
<td>5150.23</td>
<td>D</td>
<td>14.54</td>
<td>13.50</td>
<td>23.22</td>
<td></td>
</tr>
<tr>
<td>24-A2</td>
<td>6444</td>
<td>3890</td>
<td>5160.34</td>
<td>D</td>
<td>15.43</td>
<td>23.45</td>
<td>15.34</td>
<td></td>
</tr>
<tr>
<td>24-A3</td>
<td>6489</td>
<td>3967</td>
<td>5170.33</td>
<td>A</td>
<td>17.34</td>
<td>24.44</td>
<td>13.23</td>
<td></td>
</tr>
<tr>
<td>24-A1</td>
<td>6226</td>
<td>3756</td>
<td>5150.23</td>
<td>D</td>
<td>23.22</td>
<td>12.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-A1</td>
<td>6422</td>
<td>4201</td>
<td>5180.33</td>
<td>A</td>
<td>14.54</td>
<td>12.32</td>
<td>15.44</td>
<td>13.34</td>
</tr>
</tbody>
</table>

... etc.

Column 1 may be thought of as the borehole label, Column 2 as the y coordinate, Column 3 as the x coordinate, Column 4 as the ground surface elevation, Column 5 as the aquifer identifier, and Columns 6-9 as depths to the water surface on different dates.

**(1.07) WHICH COL IS THE "IDENTIFIER" (0 IF NONE, DEF=0):** The computer screen will show the first few lines (as above) of the input data file. You may select the column, if any, which is to be used as an identifier (i.e. the borehole label column 1 above). If there is no column containing such information, enter 0. If you do select a column as being the identifier column, then in the event that the same identifier number is repeated within the file, only the applicable information contained in the last occurrence is retained. For example, borehole 24-A1 is repeated. Therefore, the information associated with the second entry will be used for any operations involving Columns 8 or 9 and the first entry will be used for any contouring operations restricted to Columns 1 thru 7. This option applies only to "columnar" data files. Alphabetic data may be contained in the "identifier" column.

**(1.08) WHICH COL IS THE "SELECTOR"? (0 IF NONE, DEF=0):** A data file may be prepared that contains multiple sets of data to be separately contoured provided that the file contains a column that identifies the different sets. In the above sample data set, Column 5 is the "selector" column. If one were contouring the ground surface elevation, then a choice of "0" to this query would be proper. If one were wishing to contour the depth to water within a single aquifer, then the proper response would be "5". This option applies only to "columnar" data files.

**(1.09) ENTER SELECTOR "KEEP VALUE" (DEF=0):** This query appears only if you have responded with a non-zero answer to (1.08) above. Enter the characters which identify which data set you wish to contour (i.e., for the example, enter a "D" or an "A").

**NOTE:** The next four queries (1.10 thru 1.13) have default responses that anticipate that the original data file has only one x, y, and z triplet (in that order) entered as the first items on each line.
(1.10) WHICH COL IS "X"? (DEF=1): Enter the number of the column which contains the x coordinate (i.e. "3" in above example).

(1.11) WHICH COL IS "Y"? (DEF=2): Enter the number of the column which contains the y coordinate (i.e., "2" in above example).

(1.12) WHICH COL IS Z1? (-) SIGN IF DESIRED (DEF=3): You may contour the value of a single column in the data file or the sum or difference of any columns. A (-) sign before the column number indicates a subtraction operation. Suppose that it is desired to contour the elevation of the water table surface on the date associated with Column 7. Then a response of "4" is proper at this time as this will initially set z to the ground surface elevation.

(1.13) WHICH COL IS Z2? (FOR Z1+ OR -Z2) OR 0(DEF) WHEN DONE: Enter the number of the column (preceded by a "-" sign if necessary) which is to be added or subtracted to Z1. A "0" response signifies that no further data manipulations are wanted. To contour the water table surface on the date associated with Column 7, the proper response would be "-7" (i.e. contour (4)-(7)). This query will keep repeating (up to 5 times) until a "0" response is entered.

(1.14) ENTER NEW CHECKING DISTANCE IF DESIRED (DEF=7): The data are checked to see if any points are at or near the same coordinate locations. If any points are closer than (what will ultimately be) the specified (default=7 screen units) checking distance, the points are combined into a single data point at the average elevation of the affected points. Points which have identical coordinates are combined regardless of any specified checking distance. The default value of 7 is recommended.

(1.15a) ENTER MULTIPLIER FOR Z (RETURN IF HAPPY): Following all data calculations, the minimum and maximum z values are displayed on the screen. You may multiply all of the z values by the constant you supply. The default is 1.0.

(1.15b) ENTER MULTIPLIERS FOR X AND Y (RETURN IF HAPPY): The minimum and maximum x and y values (and their differences) are displayed on the screen. To change these values, enter a pair of constants (positive or negative, separated by a comma or space). All x and y coordinates will be multiplied by these constants. A negative multiplier will invert the (indicated) axis. The default response (1.0,1.0) will cause a file named MAP.DAT (which contains the x,y, and z values of the points to be contoured) to be created (or replaced) and execution to be transferred to the contouring activity.

(1.16) WANT NORMAL(N) OR FINE(F) INVERSE POWER DISTANCE GRIDDING (DEF=N)? (F OR N): The default (N) response will generate a 13X13 point rectangular grid of points (see 1.02). The elevations of those points are computed by inverse power distance averaging. A response of "F" will cause a finer mesh to be created. The square of number of points on an edge of the grid plus the number of original points (if retained) will be set up to, but not exceeding, 999 points.
(2.01) CONTOURING ACTIVITY..(2.01 THRU 2.44).. PROGRAM MICRO1 ISSUES THESE PROMPTS.

(2.01) THRU (2.20) REQUEST INFO TO PREPARE THE CONTOUR PLOT

(2.01) ENTER DATA FILE NAME. FIRST DEFAULTS TO 'MAP.DAT' THEN TO LAST FILE NAMED.
    OR ENTER 'TRIPLE' TO RE-CONTOUR A TRIPLE TRIANGLE MESH
    OR ENTER '*D(raw)' TO DRAW A PREVIOUSLY SAVED PROFILE
    OR ENTER '*M(ake)' TO CREATE A PROFILE FROM A PREVIOUSLY
        GENERATED AND SAVED MESH
    OR ENTER '*G(rid)' TO DRAW GRID BOX PROFILES (A MESH MUST
        HAVE BEEN PREVIOUSLY SAVED)
    OR ENTER '*R(econ)' TO REDRAW THE CONTOURS FROM A
        PREVIOUSLY SAVED MESH.
    OR ENTER 'S' TO STOP: Enter a file name or one of the following.
    (1) A carriage return will default to the file MAP.DAT which is normally
        created during the previous phase of program execution. (2) If you are re-
        contouring a "triple triangle mesh" (see 2.32) enter "TRIPLE". (3) A response
        of *D (see 2.23) will enable you to re-draw a previously generated and saved
        profile (see 2.26). (4) A response of *M will enable you to create new
        profiles from meshes that have been previously saved (see 2.21) without having
        to re-generate the contours. (5) A response of *G (see 2.41) will provide a
        means to generate multiple profiles (horizontally and vertically) on the same
        drawing provided you have previously saved a mesh. (6) A response of *R (see
        2.24a) will redraw the contours provided you have previously saved a mesh (see
        2.21). The contour intervals may be changed. (7) An "S" response exits
        McCON.

(2.02) ENTER COLOR FOR LINE WORK (DEF=YELLOW): The default color for line
drawing on the screen is yellow. Permissible responses are 'RED',
'BLUE','WHITE','CYAN','GREEN','YELLOW','BLACK', or 'MAGENTA'.

(2.02) ENTER BACKGROUND COLOR (DEF=BLACK): Same as above for background color.

(2.03) ENTER 1 OR 2 WHERE 1 IS FOR SMOOTH CONTOURS (DEF=1)
        AND 2 IS FOR SHARP CONTOURS: The default is 1. A
response of 2 prevents smoothing of the contours. This response would be ap-
licable for contouring graded terraces, or embankments and the like.

(2.04) ENTER 1 FOR SCREEN. 2 FOR PLOTTER (DEF=1).: The output may be directed
either to the screen or to a plotter.

(2.05) ENTER SIZE REDUCTION FACTOR (DEF=100)
        (NUMBER BETWEEN 1-100): Drawings directed to the plotter may be
        reduced in size as a proportion to the number selected. A response of 100
        will give full sized page drawings.

(2.06) PAPER SIZE ? ('A'=SMALL, 'B'=LARGE) (DEF=A): This program is designed
to drive small flatbed plotters. An "A" response means to plot on 8.5 inch by
11 inch size paper. "B" indicates 11 inch by 17 inch paper. Depending on the
configuration of your plotter, you may also need to set plotter "switches" for
different paper sizes.

(2.07) WANT PLOT IN SINGLE COLOR ONLY (Y/N): Multiple-pen plotters may be set
to draw using different pen colors. However, if you are preparing plots to be
reproduced in monochrome, you may wish to respond "Y".

(2.08) THE ABOVE ARE USED TO CHOOSE A SCALE TO FILL THE SCREEN.
      WANT TO CHANGE THESE VALUES ? (Y/N): The program computes (and
displays) the minimum and maximum x and y coordinates of the data points. A
scale is then automatically computed such that the resulting contour plot will
"fill" the screen. A response of "Y" to this query will enable you to change
the scale. This option may be important if you are planning to produce
several drawings which you wish to be to the same scale. If you respond "Y",
you will be queried to supply new minimum and maximum x and y limits.

(2.09) DO YOU WANT TO DESCRIBE AN INTERNAL CONVEX AREA
      TO CONTAIN ALL CONTOURING ? (Y/N): A "Y" response indicates that you
want to restrict the area in which contours will be drawn. The location of
all of the original plus grid plus border points will appear on the screen.
The following question (2.10) will then be asked.

(2.10) ENTER # OF POINTS TO DESCRIBE AREA (DEF=0)
      OR ENTER NAME OF A "FAULT" FILE: Three or more points are required to
describe the area. The default "0" will skip to query (2.12). A numeric non-
zero response will cause query (2.11) to be asked. If you respond with a file
name (of a previously prepared "fault" file) that file must consist of a
series of x,y pairs (in the same units as the original data) which define a
boundary to "cut" through the original (plus gridded) points. The order of
the x,y pairs (one pair per data file line) must be such that the original
(plus gridded) points one wishes to retain are on or are to the right-hand
side of all the lines connecting (in order given) the x,y pairs. The bound-
aries defined by the x,y pairs are not required to "close". New "fictitious"
points are generated (and displayed on the screen) at regular spacing between
the x,y pairs. The elevation (z) of these "fictitious" points are computed
via the inverse power equation (1.02), but only the "retained" points are
used to make the estimate. The normal use for specifying a "fault" file is
to break the contour plot into two areas. The first area is contoured by
specifying that there is an "area to contain all contouring" (2.09) and then
using a "fault" file to define this area. The contour plot will then consist
of contour lines drawn up to the boundary defined by the x,y pairs. The
elevations at the "fictitious" points along the boundary are estimated using
only the elevations of the points within the "area to contain all contouring".
The output should be sent to the plotter. Then the second area is contoured
by specifying a "non-contoured area" as described by (2.12) below. Specify-
ing the same "fault" file will result in a another contour plot in which the
contour lines are only drawn outside of the "non-contoured area" (i.e., within
the region excluded from the first area). The elevations along the (same)
boundary points are estimated using only the elevations of the points outside
of the "non-contoured area". Again, the output should be sent to the plotter,
but do not replace the paper. Generally, the composite contour plots will be discontinuous across the "fault" boundaries.

(2.11) DESCRIBE AREA BY POSITIONING CURSOR AT LOWER LEFT OF POINTS (IN A COUNTERCLOCKWISE MANNER) AND PRESSING "A". (CURSOR IS MOVED WITH ARROW KEYS OR -,<, AND > KEYS. --- THE -,<, AND > KEYS GIVE FINE MOVEMENT): Proceed around the required area in a counterclockwise fashion by placing the cursor just to the lower left of each of the points which will define the area. You will see a cross appear at the selected points after striking "A" at each point.

(2.12) IS THERE A NON-CONTOURED AREA (Y/N): A "Y" response will let you proceed to describe an internal convex area in which no contours will be drawn. Queries (2.10) and (2.11) will follow a "Y" answer.

(2.13) ENTER BEGINNING CONTOUR, CONTOUR INTERVAL, HEAVY LINE INTERVAL & NO. CONTOURS. (0 MEANS SPAN RANGE OF ELEVATIONS): OR 'N' FOR NO CONTOURING (DEF=AUTO) (H FOR HELP) -->: The minimum and maximum elevations are shown on the screen. You choose the beginning contour elevation, the contour interval (either as a positive or negative value), the "heavy line" interval, and the number of contour lines to be drawn. The "heavy line" interval must be a multiple of the contour interval. These "heavy" lines will plot in a different color on the video screen or be drawn as thick lines on the plotter. A zero entry for the number of lines to be drawn is interpreted to mean that contour lines will continue to be drawn starting at the beginning contour value, incrementing by the contour interval until the range of elevations is spanned. For example, presume that the screen shows that the minimum elevation is 138 ft. and the maximum elevation is 243 ft. A response of 140,10,50,0 would cause the 140, 150, 160, .. 230, and 240 (ft.) contours to be drawn. The 150 and 200 ft. contours would be "heavy." A response of 154,2,10,10 would result in the 154, 156, 158, .. 170, and 172 (ft.) contours to be drawn with the 160 and 170 ft contours to be "heavy." A response of 200,2,2,1 would result in only the 200 (ft.) contour to be drawn. The default response will result in automatic scaling. The contour interval will be set to be evenly divisible by 2,4,5, or 10 and the number of contours not to exceed 13. The default response would cause the 140, 150, .. 240 (ft.) contours to be drawn.

(2.14) LEAVE BOUNDARY OF CONTOURED REGION UNDRAWN ? (Y/N): The default (N) will cause the lines which define the convex outer boundary of the data points to be drawn. All contour lines lie within this boundary. A "Y" response will result in no drawn outer boundary and the contour lines will be left "dangling" at edges.

(2.15) DO YOU WISH TO DRAW A TEMPLATE ? (Y/N): This option provides a way to enhance the resulting contour plot with some "art-work". If a "Y" response is given, the following query will appear.

(2.16) ENTER TEMPLATE DATA FILE NAME (DEF=TMPLT.DAT): Enter the name of the file which contains the information to draw the "art-work". This file (which
must be previously prepared) consists of a set of x and y coordinates (in the same units as the original contour data) along with an indicator to raise and lower the pen. If the file looked like:

```
100 100 1
100 150 1
200 150 -1
300 200 1
300 500 -1
```

a line would be drawn from (100,100) to (100,150). Another line would be drawn from (100,150) to (200,150). The pen would then be raised (the -1 means go to the next coordinate with pen up; a 1 means pen down) and positioned at (300,200). A line would then be drawn from (300,200) to (300,500). This option provides a way to add physical features (such as building locations) to the drawing.

(2.17a) WANT TO SEE ELEVATIONS PLOTTED? (Y/N): A "Y" response will cause the z values (elevations) to be printed on the drawing. If the data points are close together, the drawing may become cluttered when this option is used.

(2.17b) WANT LABELS PLOTTED? (Y/N): A "Y" response will cause the labels (or identifiers), if any, of the original data points to be printed on the drawing. Again, the drawing may become cluttered if the data points are close together.

(2.17c) LEAVE X-Y AXES ANNOTATIONS OFF? (Y/N): The X and Y axes may be labeled with tic marks and coordinate values if desired. The coordinate values which are printed are automatically calculated as "nice" numbers.

(2.18) LEAVE ELEVATION TIC'S AT DATA PTS OFF TOO? (Y/N) (ENTER 'A' TO INCLUDE GRID POINTS ALSO): The default (N) causes small crosses (tic's) to be drawn at the location of the ORIGINAL data points. A "Y" response will cause the tic's to remain undrawn. A response of "A" will cause the ORIGINAL PLUS ADDED DATA POINTS tic's to be drawn.

(2.19) ENTER TITLE OF PLOT-- (IF NONE, HIT RETURN): The title of the plot (if any) will be printed on the drawing.

(2.2) SKIP LEGEND? (Y/N): The contours lines a-e labeled "A", "B", etc.. The legend (which will be printed on the right margin of the contour plot) lists the numerical value of the labels.

(2.21) THRU (2.44) RELATE TO OPERATIONS FOLLOWING THE COMPLETION OF THE TRIANGLE MESHES AND THE PRODUCTION OF THE CONTOUR PLOTS.

(2.21) DO YOU WANT TO SAVE THIS MESH (SO THAT YOU MAY LATER DRAW OTHER PROFILES)? (Y/N): This query occurs after the contouring is completed. A response of "Y" will let you save certain results of
the mesh generation which will enable you to later rapidly create profiles along any section of the mesh. If "Y" is the response the following question appears.

(2.22) ENTER A FILE NAME IN WHICH TO STORE THE MESH.
(DEF=M.MSH): Choose a file name in which the mesh generation data will be stored. The default file name is M.MSH.

(2.23 WANT TO DRAW A PROFILE ? (Y/N): A "Y" response indicates you wish to create a profile of the elevation data from some given point to another point. This query will occur following the completion of a contour plot or as a result of the *M(ake) option discussed under (2.01). Further questions about the nature of the profile will ensue.

(2.24a) WANT TO REDRAW THE CONTOURS ? (Y/N): A "Y" response will permit the contours to be redrawn (provided a "mesh" was previously saved). The contour interval may be changed if desired.

(2.24b) WANT THE VOLUME UNDER THE MESH ? (Y/N): The volume of the space contained between some reference elevation (next query) and the contoured elevation for the entire mesh area may be computed. The units of the volume are in terms of the units implied in the original data file.

(2.24c) ENTER REFERENCE 'BASE' ELEVATION (DEF=0.0): The minimum elevation of all data points is shown on the screen. Enter the elevation you wish to be used as the reference for volume calculations. The default response will cause an elevation of 0.0 to be used as the reference. After the profile is drawn, the volume will be printed at the bottom of the drawing as will the area of the mesh and the average elevation (the average elevation is the elevation which, if used as the reference elevation, would give a volume of zero).

(2.25a) ARE PROFILE LINE ENDPOINTS TO BE LOCATED WITH THE CURSOR (C)
OR FROM A NAMED FILE (F) ? (DEF=C): The endpoints of the profile line may be located either by positioning the crosshair cursor or by supplying the (x,y) coordinates of the two ends. If the response is "C" (for cursor), queries (2.25b), (2.25c) and (2.25d) will be issued. If the response is "F" (for file), you will be asked the name of a one line file containing the x-y coordinates of the endpoints (i.e., x1,y1 - x2,y2) or you may respond that the file name is CON (for console) and directly type in the endpoint coordinates.

(2.25b) WANT CURSOR TO INDICATE NEAREST (TO UPPER RIGHT)
DATA NODE POINT ? (Y/N): The data node points will be displayed on the screen. A "Y" response will cause the location of profile endpoints to be positioned exactly on the node point located closest (to the upper right) of the crosshair cursor. A "N" response will position the profile endpoints at the position of the crosshair cursor.
(2.25c) POSITION CURSOR AT 1ST END OF PROFILE LINE AND HIT RETURN

**...USE <,>, , AND 6 KEYS TO 'FINE' POSITION CURSOR:** Position the cursor at one end of the line defining the section along which you want a profile. Move the cursor with the "arrow" keys or, for fine movement, with the <,>, , or 6 keys. Strike the "return" key when cursor is properly located.

(2.25d) POSITION CURSOR AT 2ND END OF PROF. LINE AND HIT RETURN: Same as above except to locate other end of profile line.

(2.26) Want profile (D)rawn on this plot, sent to (F)ile or (B)oth ? (Enter D,F or B) DEF=D: When this question is asked, a plot of the data points will be visible on the screen. A response of "D" will cause the profile to be drawn (after a question regarding the scale of the drawing) directly on the current plot skewed along the positions indicated via query (2.25a). A response of "F" will send the profile data to a file (which is named by (2.27)). Profiles saved via the "F" response may be drawn later by responding *D(raw) to query (2.01). A response of "B" will cause both actions ("D" and "F") to occur.

(2.27) ENTER A FILE NAME (DEF=D.PRO): Enter a file name in which to store the profile data. The file name entered here will be required by the *D(raw) response to query (2.01).

(2.28) ENTER A ONE LINE DESCRIPTION: The description you enter will be retained in the file named above (2.27) and will, when the profile is redrawn via the *D(raw) response to (2.01) be printed at the top of the drawing. If no description is desired press the "return" key.

(2.29) ENTER SECTION LABEL (LIKE A FOR A-A): The section label you enter will be retained in the file named by (2.27). This label will also be placed at the ends of the profile section on the drawing showing the data point locations.

(2.30) ENTER Y-AXIS LABEL (DEF=ELEVATION): When the profile is redrawn via the *D(raw) response to (2.01) this description will be printed as the label of the "elevation" axis.

(2.31) ENTER YOUR MIN AND MAX PROFILE ELEVATIONS (DEF=MIN AND MAX ELEVATION OF ALL POINTS ON PLOT): The minimum and maximum elevations for the all data points within the entire mesh and along the profile line are shown on the screen. The values you chose will influence the "scale" of the resulting profile.

(2.32) WANT TO GENERATE A TRIPLE TRIANGLE MESH ? (Y/N): This option provides yet another method to "massage" the original input data (i.e. an alternate to the inverse power scheme discussed under (1.02)). All contouring is accomplished by creating a mesh of triangles which connect every data point (original plus grid points if any). The effect of a positive response to this query: to add a (fictitious) data point at the centroid of each of the "old" triangles. A file named "TRIPLE" which consists of the old plus the
"fictitious" points is created. When the program cycles back to (2.01) you must respond with "TRIPLE" to create a new contour plot that considers the new points. This option may also be used to refine "inverse power" grids. If the response is "Y" the following question is asked.

(2.33) **LINEAR OR INVERSE SQUARE RULE? (L OR I)** DEF=L: The elevation of the centroid of the "triple triangles" may be computed in two ways. A "L" response will compute the elevation, $Z_c$, of the centroid via the equation, $Z_c = a + bx + cy$, where the constants $a$, $b$, and $c$ are determined by substitution of the $x$, $y$, and $z$ values at the nodes of the triangle containing the centroid. A response of "I" will cause the centroid elevation to be computed via the "inverse power" equation given under (1.02) using only the three points which define the triangle. The default is "L".

(2.34) **DO YOU WANT TO DRAW THE MESH OF TRIANGLES? (Y/N):** A response of "Y" will cause the triangle mesh to be drawn.

(2.35a) **WANT TO DRAW A PREVIOUSLY SAVED PROFILE? (Y/N):** This is the query you will get by using the *D(raw) response to (2.01). Answer "Y" to create page size profiles of "saved" profiles.

(2.35b) **1ST PROFILE OF A SERIES ON SAME PLOT? (Y/N):** More than one profile may be drawn on the same plot. This option will provide a means to retain the same scaling parameters for multiple profiles. Answer "Y" for the first profile to be drawn; "N" responses from that time on will cause subsequent profiles to be drawn on the same screen (or page).

(2.36) **ENTER SAVED DATA FILE'S NAME (OR "Q" TO QUIT) DEF=D.PRO:** Enter the name of a previously saved "profile" data file or enter "Q" to return to query (2.01).

(2.37) **WANT THE PROFILE STRETCHED TO FILL THE PAGE? (Y/N):** The default response (N) will let you draw this and subsequent profiles to the same horizontal scale, i.e., if one profile is longer than another, the drawings will also be of different widths. A response of "Y" will cause each profile to fill the page width.

(2.38) **ENTER YOUR ZMIN AND ZMAX (DEF=AUTO):** The minimum and maximum elevations encountered along the profile section are shown on the screen. Enter the values you wish to see as the minimum and maximum values for plotting purposes. The default response will cause automatic scaling.

(2.39) **HOW MANY Z UNITS PER PLOT INCH (DEF=AUTO):** Enter the number of vertical (elevation) units per (approximate) plot inch. The resulting plot may not be more than 5 inches tall. An entry which would result in a plot in excess of 5 inches will be rejected and this query will again appear. The default response will cause automatic scaling.

(2.40) **ENTER NAME OF "MESH" FILE (DEF=M.MSH OR "Q" TO QUIT):** This query will result when responding either *M(ake) or *G(rid) to (2.01). Enter the name of
the file in which mesh data was previously stored (2.22). The default will assign "M.MSH" as the file name.

(2.41) WANT TO DRAW GRID BOX PROFILES? (Y/N): This option permits the simultaneous creation and drawing of multiple profiles located within a rectangular grid box of your choosing. A response of "Y" will cause a request for the "MESH" (see 2.40) file. You will also be asked (see 2.41a) to describe the box dimensions in which the profiles are to reside.

(2.41a) ENTER PROFILE GRID BOX ORIGIN OF COORDINATES, ANGLE OF BOX TO HORZ, LENGTH OF BOX, & HEIGHT. (X,Y,ANG,L,H)
OR ENTER 'S' FOR SAME AS LAST TIME (DEF=AUTO): Enter the coordinates of the lower left corner of the box in which the grid profiles are to be drawn (in terms of original data coordinates), the angle the box makes with the horizontal (usually 0), the length of the box and the height of the box. Enter 'S' to use same grid box as previous. The default response will generate a box which fits inside the span of the data.

(2.42) WANT PARALLEL (TO BASE) OR NORMAL GRIDS?
Input P or N (DEF=P): Respond appropriately for the required set of profiles. A response of P will cause the profiles to be drawn parallel to the base of the grid box. A response of N causes the profiles to be drawn normal to the base of the grid box.

(2.43) HOW MANY PROFILES ON THE PAGE? (DEF=11): Enter the number, n, of profiles to be created. The profiles are evenly spaced with the 1st and last profiles being drawn just slightly inside the appropriate edge boundaries of the mesh. The default response of "11" will result in 11 profiles drawn (horizontally or vertically) centered on the middle of the region. A series of 'n' files will automatically be created (for further processing under the *D main menu (see 2.01) option). These files are named with the same root as the ??..MSH file (see 2.41) with the extension .OP, .1P, etc., (for profiles drawn parallel to the base of the grid and .ON, .1N, etc. for profiles drawn normal to the base of the grid). That is, if the mesh file is named M.MSH and 5 profiles parallel to the base of the grid box are to be created, the profile files will be named M.0P, M.1P, M.2P, M.3P, and M.4P. These files may then be selected (see 2.36) to prepare individual profiles under the *D option (see 2.01).

(2.44) ENTER SCALE EXAGGERATION FACTOR (DEF=1.0): The default response (1.0) will result in the 'height' of the profiles to be about 1/2 inch (maximum). The value of your response is used to cause an exaggeration to the height of the profiles.
APPENDIX B. FORTRAN LISTING FOR McCON

*** ((( McCON ))) ***

A CONTOURING PROGRAM...

WRITTEN BY
JOHN B. PALMERTON
GEOTECHNICAL LABORATORY
WATERWAYS EXPERIMENT STATION
U.S. ARMY CORPS OF ENGINEERS
VICKSBURG, MISSISSIPPI
AUG 1991

THIS PROGRAM REQUIRE A SET OF (N) X-Y-ELEV DATA POINTS.

A GRID IS CREATED WHICH CONNECT THE DATA POINTS INTO TRIANGLES.

THESE TRIANGLES WILL COMPLETELY OCCUPY THE CONVEX REGION DEFINED

BY THE EXTERIOR POINTS. THE GRID IS CREATED BY:

1. CREATING A SEED TRIANGLE WITH EDGES 1-2-3 LABELED COUNTERCLOCKWISE.

AND FORMS THE LARGEST ANGLE WITH THAT EDGE.

2. IN ORDER OF EDGES, FORMING NEW TRIANGLES BY SELECTING THE POINT

WHICH LIES OUTSIDE OF THE EDGE OF THE GENERATING TRIANGLE

3. REPEATING THIS PROCESS FOR ALL EDGES OF ALL TRIANGLES UNTIL

EDGES ARE SHARED BY TWO TRIANGLES OR A CONVEX BOUNDARY IS

FORMED.

CONTOURING IS DONE BY ASSUMING A LINEAR INTERPOLATION OF ELEV.

BETWEEN TRIANGLE VERTEXES. AS EACH NEW TRIANGLE IS FORMED THE

CONTOUR LINES WHICH CROSS THE SHARED EDGE ARE ARE DRAWN AS EITHER

STRAIGHT LINES OR CIRCULAR SEGMENTS. IF THE USER WISHES TO

INSURE THAT ALL CONTOURS ARE STRAIGHT LINES BETWEEN INTERPOLATION

THIS SET THE VARIABLE OSP=1. THIS IS USEFUL FOR REPRESENTING

SITUATIONS INVOLVING GRADED LANDFORMS.

DEFINITION OF SOME PARAMETERS

X(I),Y(I),Z(I) ---- COORDINATES AND ELEVATION VALUES OF DATA POINTS

VERTEX(I,J) ------ POINTS(I) AT VERTICES OF TRAINGLE(J) NUMBERED

COUNTERCLOCKWISE

NODE(I) ---------- LINKED LIST OF THE NUMBERS OF THE ENDPOINTS

WHICH HAVE THE OTHER END AT POINT I.

COMMON/CONTOUR/NC,CL(500),IAANNS,FACDPX,PY,PZ,CLS
COMMON/CRAP/DEL,MN
COMMON/POW/POW
COMMON/ARCO/LCON(999),MED,MCON
COMMON/REV/REVX,REVF,PREX,PREF(8)
COMMON/COLOR/CL06,CL02,CL03,CL04
COMMON/WIND/WMIA,WMIN,YMIN,YMAX,FSOFT
DIMENSION XX(999),YY(999),ZZ(999)
CHARACTER FNME(12),BNME(12),FMRE(12)
CHARACTER COL(10),CL06*4,CL02*4,CL03*4,CL04*4
DATA COL/'YELL','BLAC','RED','CYAN','GRED','WHITE','MAGE','BLUE'/
DATA FMRE/1,
FNME='MAP.DAT'

5933 PRINT, ('A11','ENTER A "Y"')
READ, (I1,FNME)
CONTINUE
IF (I1.EQ.0) GO TO 10
IF (I1.EQ.1) PRINT, 'SET CAPSLOCK !!!
GO TO 5933

CONTINUE
88 CONTINUE
88 0=1
88 CONTINUE
PRINT: "PLEASE NOTE:"
PRINT: "FOR QUESTIONS THAT ARE TO BE ANSWERED, "Y" OR "N" 
PRINT: "ARE RESPONSES, INCLUDING A CARRIAGE RETURN) OTHER THAN:"
PRINT: ""NO" MEANS "Y", A CARRIAGE RETURN TO CONTINUE."
PRINT: "!
PRINT: "ENTER DATA FILE NAME. FIRST DEFAULTS TO "MAP.DAT" 
PRINT: "OR ENTER "TRIP". TO USE TRIANGLE MESH. 
PRINT: "OR ENTER "DIR". TO DRAW A PREVIOUSLY SAVED PROFILES. 
PRINT: "OR ENTER "RAKE". TO CREATE A PROFILE FROM A 
PRINT: "PREVIOUSLY GENERATED AND SAVED MESH."
PRINT: "OR ENTER "GRID". TO DRAW GRID BOX PROFILES (A MESH)."
PRINT: "OR ENTER "RECON". TO REDRAW THE CONTOURS, FROM A 
PRINT: "PREVIOUSLY SAVED MESH."
PRINT: "OR ENTER "S", TO STOP"
READ (A12). (FILE,.S ) THEN 
READ 09
CLOSE 09
FILE='MAP.DAT', STATUS='OLD'.ERR=1881)
CLOSE 09. STATUS='DELETE'
1881 OPEN 09,FILE='ED'. STATUS='OLD'.ERR=1882)
1882 OPEN 09,FILE='SKIN'. STATUS='OLD'.ERR=1883)
1883 CLOSE 09, STATUS='DELETE'
1884 OPEN 09,FILE='SAVCON'. STATUS='OLD'.ERR=1884)
1885 CLOSE 09, STATUS='DELETE'
1886 OPEN 09,FILE='OUTSIDE'. STATUS='OLD',ERR=1886)
1886 CLOSE 09, STATUS='DELETE'
1887 CONTINUE
STOP
END
IF(NCOLOR.EQ.0) THEN 
NCOLOR=1.
PRINT: "COLOR CHOICES ARE: RED, YELLOW, BLUE, MAGENTA, CYAN"
PRINT: "BLACK, GREEN, AND WHITE"
PRINT: "A"
READ (A). COLOR
IF(COLOR.EQ.'RED') COLOR='RED'
IF(COLOR.EQ.'GREEN') COLOR='GREEN'
IF(COLOR.EQ.'BLUE') COLOR='BLUE'
IF(COLOR.EQ.'MAGNET') COLOR='MAGNET'
IF(COLOR.EQ.'CYAN') COLOR='CYAN'
IF(COLOR.EQ.'YELLOW') COLOR='YELLOW'
IF(COLOR.EQ.'WHITE') COLOR='WHITE'
READ (A) COLOR
IF(COLOR.EQ.'BLACK') COLOR='BLACK'
CLRBCK=COLOR
IF(CLRBCK.EQ.'RED') CLRBCK='RED'
IF(CLRBCK.EQ.'GREEN') CLRBCK='GREEN'
IF(CLRBCK.EQ.'BLUE') CLRBCK='BLUE'
IF(CLRBCK.EQ.'YELLOW') CLRBCK='YELLOW'
IF(CLRBCK.EQ.'WHITE') CLRBCK='WHITE'
IF(CLRBCK.EQ.'CYAN') CLRBCK='CYAN'
DO 8904 I=1,8
8904 CONTINUE
DO 7701 J=1,J,J+8
7701 CONTINUE
DO 7702 I=1,I,I+8
7702 CONTINUE
DO 7703 I=1,8
7703 CONTINUE
DO 7704 I=1,8
7704 CONTINUE
7705 COLOR=COL4
CONTINUE
IF(IEWME.EQ.'1') FMME=IEWME
IF(IEWME.EQ.'R') GO TO 2777
IF(IEWME.EQ.'W') GO TO 2777
IF(IEWME.EQ.'E') GO TO 2777
IEWME=IEWME
OPEN 1,FILE=IEWME,STATUS='OLD',ERR=567)
GO TO 568
CONTINUE 568 READ 9,(AIJE) IEWME
READ 9,IS,SS
READ 9,0
OPEN 9,FILE='MCINFO.DAT')
WRITE 9,(AIJE) IEWME
DO 189 IS=1,5,1
CLOSE 9
IF(IEWME.EQ.'0') THEN
PRINT 'THIS PROGRAM EXPECTS UP TO 999 NODES IN FILE: MAP.DAT.'
PRINT 'MAP.DAT IS A FREE FIELD FILE OF X,Y,2 VALUES--1 SET,'
PRINT 'PER LINE. THE POINTS WILL BE SCALLED TO FIT SCREEN.'
PRINT 'TEMPLATES MAY BE DRAWN ON THE CONTOUR MAP. THE TEMPLATE'
PRINT 'DATA MUST BE IN A SEPERATE FILE: --A FREE FIELD FILE.'
PRINT 'A NEGATIVE PEN VALUE INDICATES,'
PRINT 'THE END OF A TEMPLATE (IE PEN IS UP TO NEXT POINT),'
PRINT '{USE SAME COORDINATE SCALE AS SCALE FOR MAP.DAT}')
END IF
IF(IEWME.EQ.'0') THEN
PRINT 'DATA FILE ',IEWME,' NOT FOUND--
GO TO 5
END IF
IF(IEWME.EQ.'0') THEN
PRINT 'DATA FILE IS ',IEWME,' 
END IF
IF(IEWME.EQ.'TRIPLE') IPOW=1
CONTINUE 876 JNC=1
NC=0
IF(IEWME.EQ.'1') NC=1
IF(IEWME.EQ.'4') NC=3
IF(IEWME.EQ.'0'.OR.NC.EQ.2)) Go To 877
CONTINUE 877 QDEL=25(JNC).GT.1) QDEL=1.
REWIND 09
CONTINUE 777 PRINT 'FOR THE OUTPUT............'
PRINT '(A1)'
READ 09,(A1) JNC
PRINT ';............WAIT'
MM=0
IF(IEWME.EQ.'1') MM=1
IF(IEWME.EQ.'4') MM=2
IF(IEWME.EQ.1) DEV=1BMH
IF(IEWME.EQ.3) DEV=PS
REWIND 09
PCY=1.0
PCX=1.0
IF(IEWME.EQ.) THEN
PRINT '(2.05) ENTER SIZE REDUCTION FACTOR (DEF=100)'
PRINT '(A1) [MUR. BETWEEN 1-100] FOR PLOTTER )'
READ 'A10',PERC
IF(PERC.EQ.10.) PERC=100.
IF(PERC.EQ.0.) THEN
OPEN 1,FILE='MCINFO.DAT')
WRITE 14,(AIJE) PERC
END IF
B3
READ *) ERR=500, QS.J, FCY.
CLOSE(3, STATUS='DELETE')
ENDF
IF (PCY.EQ.0.) PCB=100.
PCX=PCY
PRINT '(A):'
+ 2.06 PAPER SIZE? ('A'=SMALL,'B'=LARGE) (DEF=A)'
READ *) PSIZE
IF (PSIZE.EQ.0.) PCB=825*PCY
PRINT '(A): WENT PLOT IN SINGLE COLOR ONLY? (Y/N)'
READ *) ANS
IF (ANS.EQ. 'Y') THEN
GO TO 804
PRINT: 'PEN1 IS USED TO DRAW CONTOURS AND LEGEND.'
PRINT: 'PENS IS IS DATA FILES, POINTS & ELEVATIONS.'
PRINT: 'PENS IS IS TEMPLATES.'
PRINT: 'PENS IS IS HORIZONTAL (PARALLEL) GRID PROFILES'
PRINT: 'PENS IS IS VERTICAL (NORMAL) GRID PROFILES'
PRINT: 'SO MAKE SURE YOU KNOW WHAT IS IN THE PEN HOLDER!'
CALL UBE
CALL UPAUSE
END
ENDIF
IF (ANS.EQ. 'Y') THEN
GO TO 804
804 PCB=I=1.8
PRINT: 'PEN1 IS использует контур и легенду.'
PRINT: 'PEN1 IS использует данные файлы, точки и высоты.'
PRINT: 'PEN1 IS использует шаблоны.'
PRINT: 'PEN1 IS использует горизонтальные (параллельные) сетки профилей.'
PRINT: 'PEN1 IS использует вертикальные (нормальные) сетки профилей.'
PRINT: 'Так что убедитесь, что вы знаете, что находится в стенде для стержней!'
CALL UBE
CALL UPAUSE
END
ENDIF
802 PCB=I=1.8
PRINT: 'PEN1 IS used to draw contours and legend.'
PRINT: 'PEN1 IS used to draw data, points, and elevations.'
PRINT: 'PEN1 IS used to draw templates.'
PRINT: 'PEN1 IS used to draw horizontal (parallel) grid profiles.'
PRINT: 'PEN1 IS used to draw vertical (normal) grid profiles.'
PRINT: 'So, make sure you know what is in the pen holder!'
CALL UBE
CALL UPAUSE
END
ENDIF
IF (ANS.EQ. 'Y') THEN
GO TO 804
FORMAT (A1)
IF (MM.EQ.2.) PCELL=999
CALL DPAWCON(IPCELL)
CALL PEND
IF (MM.EQ.3) PRINT*, 'ENTER EPRINT TO SEND TO PLOTTER'
STOP
CLOSE(IO9)
GO TO 3
STOP
C
SUBROUTINE CONTER(ZINT)
COMMON (ZINF, 1) NNLK
COMMON (0D, 59)
COMMON/ORCC(NCC(999), X(999), Y(999), Z(999), CX(3), CY(3), CZ(3),
+ 1, VERT(2000), SIDE (2000), TINS, FAX) PK, PY, PZ, CLS
COMMON/BOUND/MCON, LCON(999), MED, MCON
COMMON/PIES/NUM(1000), API(100), IRD(100), KRD, AVX, AVY
COMMON/MINS, XMAX, XMIN, YMAX, YMIN
COMMON/MASK NUM(1000), MOMB
COMMON/NODE/NDWRITE(1000), NAV, NMAX
COMMON/DEV/DEV, PCY, PCB, PSIZE, CLRBCK, PSIZE, LPEN(8)
COMMON/COLOR/FLAGS, COLS, COLS, COLS
COMMON/0X/0X, 0XMAX, 0XMIN, YMAX, YMIN
DIMENSION LED(1000), SELF(1), TP(12)
DIMENSION (X, Y, Z, N, M)
CHARACTER COL*4, COL*4, COL*4, COL*4
CHARACTER DEV*1, COLOR*4, LRBC*4, PSIZE*1
OPEN(5, FILE = 'SAWCON '
DO 12 I=1,1000
   VERTEX(I:3)=0
  12 SIDE(I,J)=0
DO 820 J=1,11,MP
   NODE(I)=0
   NAV=NP+1
   MP=1
   MB=1
   MSTART=1
   MBNUM(I)=0
   DO 950 J=1,3, KRD
      NP=RN(JJ)
      IF (NP.EQ.2) THEN
         GOTO 952
      ELSE IF (NP.EQ.3) THEN
         GOTO 952
      ELSE IF (NP.EQ.2) THEN
         GOTO 952
      ELSE IF (NP.EQ.3) THEN
         GOTO 952
      END IF
   952 CONTINUE
   IF (NP.EQ.2) CALL SEED
   CONTINUE
   DO 200 L=1,3
      IF (NP.EQ.1) THEN
         NP=NP+1
      ELSE IF (NP.EQ.2) THEN
         NP=NP+1
      ELSE IF (NP.EQ.3) THEN
         NP=NP+1
      END IF
   200 CONTINUE
   IF (SIDE(I,J).LT.0) GO TO 200
   IF (SIDE(I,J).GT.0) GO TO 250
   D1=0.000000
   D2=SQR(D1)
   CONTINUE
   IF (SIDE(I,J).EQ.0) GO TO 708
   CONTINUE
   LLIN=0
C SET MIN ANGLE EQUAL TO 15 DEGREES
   AS=30.0
   CX1=VERTEX(I,J,IS(1))
   CY1=VERTEX(I,J,IS(1))
   CX2=VERTEX(I,J,IS(1))
   CY2=VERTEX(I,J,IS(1))
   SD1=(CX1-CX2)**2+(CY1-CY2)**2
   DO 200 J=1, NP
      IL=NR(J)
      IF (SIDE(I,J).GE.1) THEN
         GOTO 707
      IF (SIDE(I,J).NE.0) THEN
         GOTO 707
      END IF
   707 CONTINUE
   CONTINUE
   LLIN=0
C................. CRITERIA A
   A=CI1*(CY2-V(I1))**2+(V(I2)-V(I1))*X(I)**2(CY1-CY2)
   IF (A.EQ.0.0) THEN
      60 TO 250
   IF (A.LT.0.00001) GO TO 250
   C................. CRITERIA B
   LLIN=1
   SD2=CX1**2+(CY1-V(I1))**2
   IF (DASH.EQ.1) AD=SD2+SD3
USE STATEMENT ABOVE TO MINIMIZE PERIMETER OF TRIANGLE

IF AD<GE ASM GO TO 250

C USE STATEMENT BELOW TO MAXIMIZE VERTEX ANGLE OF TRIANGLE

IF AD=GE ASM GO TO 250

C CRITERIA FOR MAXIMUM ANGLE OF TRIANGLE

C IF THE ANGLE IS GREATER THAN 100 DEGREES FIND MINIMUM PERIMETER

IF (AD<GE.qo AND AD.LT.-1.73) THEN
  IDASM=1
ENDIF

IF AD=40 THEN
  NU=1
ENDIF

250 CONTINUE

IF (P.EQ.MP AND NCRIC.EQ.0) THEN
  IF (NU.EQ.0) GO TO 260
  NCRIC=1
  I=NU
  GO TO 8816
C NOW IS THE PARENT TRIANGLE
C NOW = MB
    VERTEX(MB, 1) = NW
    VERTEX(MB, 2) = VERTEX(MB, [1])
    VERTEX(MB, 3) = VERTEX(MB, [2])
    IF(I = 2) THEN
        CALL WORD(1, X(VERTEX(MB, 1)), Y(VERTEX(MB, 1)))
    ENDIF
    CALL WORD(1, X(VERTEX(MB, 2)), Y(VERTEX(MB, 2)))
    CALL WORD(1, X(VERTEX(MB, 3)), Y(VERTEX(MB, 3)))
    CALL WORD(1, (X11 + X12 + X13)/3., (Y11 + Y12 + Y13)/3.)
    MB = MB
    CALL UPRTN1(I XMB, 'INTE')
    CALL AREA(I X11, X12, X13, A)

C IF A.T.O. Y(1) > Y(2) > Y(3) THEN
C
    DO 290 J = 1, 3
    290 P(J) = VERTEX(MB, J)
    CALL ORDERING(P(J), P(1))
    SIDE(P(1), P(2), MB = NS)
    CALL ORDERING(P(2), P(3))
    SIDE(P(3), P(1), MB = NS)
    IF( (MPAREN, IPAREN) = IABS(SIDE(MPAREN, IPAREN)) ) THEN
        KN = NODE(NW) / 100000
        IF(KN.LT.0) GO TO 310
    ENDIF
    CALL IPRNT1(X(MB), 'INTE')
    CONTINUE
    IF(KN.GE.0) GO TO 311
    EN
    IF( IF > 0 ) THEN
        IF(M2 = M2) THEN
            S = S1
            M = M2
        ELSE
            S = S1
            M = M1
        ENDIF
        IF(S1.EQ.M) THEN
            IF(M1.EQ.M) THEN
                IF(S1.EQ.M) THEN
                    CALL IPRNT1(X(M2), 'INTE')
                ELSE
                    IF(M1.EQ.M) THEN
                        IF(S1.EQ.M) THEN
                            CALL IPRNT1(X(M2), 'INTE')
                        ELSE
                            CALL IPRNT1(X(M1), 'INTE')
                        ENDIF
                    ELSE
                        CALL IPRNT1(X(M1), 'INTE')
                    ENDIF
                ENDIF
                CONTINUE
            ELSE
                CALL IPRNT1(X(M1), 'INTE')
                CONTINUE
            ENDIF
        ELSE
            CALL IPRNT1(X(M2), 'INTE')
            CONTINUE
        ENDIF
    ENDIF

C THE TRIANGLE(S) WHICH SHARES NODE NW IS M2
C M2 = LNW - (LP * 100000)
C LP = NODE(I M2)
C L2 = NODE(I LP)
C X2 = (SIDE(M2, M1)) .EQ. IABS(SIDE(M2, M3))) GO TO 350
C 310 CONTINUE
C 320 CONTINUE
C 330 CONTINUE
C 340 CONTINUE
C 350 CONTINUE
C 360 CONTINUE
C 370 CONTINUE
C 380 CONTINUE
C 390 CONTINUE
C 400 CONTINUE
C 410 CONTINUE
C 420 CONTINUE
C 430 CONTINUE
C 440 CONTINUE
C 450 CONTINUE
C 460 CONTINUE
C 470 CONTINUE
C 480 CONTINUE
C 490 CONTINUE
C 500 CONTINUE
C 510 CONTINUE
C 520 CONTINUE
C 530 CONTINUE
C 540 CONTINUE
C 550 CONTINUE
C 560 CONTINUE
C 570 CONTINUE
C 580 CONTINUE
C 590 CONTINUE
C 600 CONTINUE
C 610 CONTINUE
C 620 CONTINUE
C 630 CONTINUE
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMMON/contour/nc,cl(1000),ans,incident,px,py,ptz,cls
C
C subroutine nodes
COMM
SUBROUTINE DLAREA(A, A2, A3, B1, B2, B3, A)
  D = SQRT(MAX(\(A2-A3)^2, (A2-B1)^2, (A2-B3)^2\)) \cdot MAX(\(A2-A1)^2, (A2-B1)^2, (A2-B3)^2\)) \cdot MAX(\(A2-A1)^2, (A2-B1)^2, (A2-B3)^2\))
  AREA = \frac{1}{2} \times D \times (A1-A2) \times (B1-B2)
  IF(D .LT. 0.0000001 AND A .GT. 0.0000001) A = 0.
  RETURN
END

SUBROUTINE OEXPR1
  COMMON/COR/COMP, X(100), Y(100), Z(100), CX(5), CY(5), CZ(5)
  CONTINUE
  RETURN
END

SUBROUTINE ORDER(N, M, I)
  N = N+1000
  I = I+1000
  RETURN
END

SUBROUTINE ORCHIT(M, MB, ZZ, I1, I2, I3, I4, K)
  COMMON/STUFF(Z), Y(100), Z(100), CX(3), CY(3), CZ(3)
  COMMON/COR/COMP, X(100), Y(100), Z(100), CX(5), CY(5), CZ(5)
  RETURN
END
C FIND CIRCLE WHICH INSCRIBES 1-2-3....A1 IS ANGLE 1-2-3
D=SQRT( (X1-X2)**2+(Y1-Y2)**2 )
D2=SQRT( (X2-X3)**2+(Y2-Y3)**2 )
SL=ABS( (SL-D2)/(SL-D3) )
IF SL>D2.CT.0.01 THEN
KNEW=X2
YNOW=Y2
IF YNOW-Y1.EQ.0.0.AND.XNOW-X2.EQ.0.0) ANG=0
ELSE ANG=A2( YNOW-Y1, XNOW-X1 )
CALL USEP(TKNO+M+FBL*STG(ANG), YNOW-FBL*COS(ANG))
GO TO 400
END IF
TANA=SL/(SL-D2)
SIG=A*TAN(TANA)
C DEL IS THE DELTA OF THE CURVE 1-2-3
DEL=1.570796-AI*16*57.29578
SIG=DEL
578 CONTINUE
C MOVE POINT 1 SUCH THAT THE DISTANCE BETWEEN THE PT & THE CURVE C IS LESS THAN OR EQUAL TO DEL (SET DEL=1 FOR SHARP CONTOURS).
D2=Y2-X1**2+(Y2-Y1)**2
DCH=DEL*STG(1.570796-AI)/(1.-COS(1.570796-A1))
D2=DCH**2
IF (DCH.0.0.D2.0.05) THEN
C NEW COORDINATES FOR POINT 1 (IF NECESSARY)
TO 378

C

IF

C

MAKE
SURE
THAT
THE
CURVE
CROSSES
THE
COMMON
EDGES
OF
THE
TWO

C

IF
NOT,
MOVE
THE
PT
POINT
1)
CLOSER
to
POINT
2.

837

838

834

C
REDUCE
DEL.
IF
NECESSARY

935

934

CONTINUE

GO
TO
837

END

C

SIMILAR
TRIANGLES

R=SQRT(CHORD**2/2(CHORD))

C
AND
Y
COORDINATES
OF
CENTER
OF
CURVE
OF
RADIUS

XC=X-R*SIN(\theta)

YC=Y+COS(\theta)

RAD=\theta-R*(XC-XC)**2

C

MAKE
SURE
THAT
THE
CURVE
CROSSES
THE
COMMON
EDGES
OF
THE
TWO

C
TRIANGLES.

835

QDEL=PDEL

ANG=ATAN2((Y1-Y2)*(X1-X2),(X1-X2)**2+(Y1-Y2)**2)

CALL
UPFNL\[X3+FB\*\cos(ANG),Y3-\cos(ANG),DEL\]

ENDIF

400

CONTINUE

CALL
UPFNL\[X3+FB\*\cos(ANG),Y3-\cos(ANG),DEL\]
IF X(i) .GT. 0.1 VSIGN=-30
CALL UMVEC(X1,Y1,X3,X5*F5:T1)/2,1Y10W-Y3/2.
IF(X(i).LT.2.IBX=2
IF(IBX.GT.2)IBX=1
IF(IBX. LT.51IBX=53
IF(IBX.IX,YF).NE(0) GO TO 890
DO 893 X=1,IBX+1
893 CONTINUE
CALL USEC ('INTE')
CALL USEC (COL2)
IF=26.
IF(KF.LE.26)KF=26
IF(KF.EQ.1.AND. IBOLD.EQ.2) FBL=FSOFT*0.75
 IF(KF.EQ.2) FBL=FSOFT*0.75
 CALL UPRNT1(LET(Y10WK3), 'TEXT')
 CALL USEC (COLOR)
 CALL USEC ('INTE')
END
763 CONTINUE
CALL USEC (COL1)
RETURN
450 CONTINUE
DO 766 J=1,IBOLD
766 CONTINUE
CALL USEC (COL1)
RETURN
C
SUBROUTINE LOC(ZZ,Z1,Z2,XP,YP,JP, ISCC)
INTEGER SIDE, VERTEX
COMMON/COORDC2999, Y(999), Z(999), CX(3), CT(3), CZ(3),
* VERTEX(2000,3), SIDE(2000,3), NP
ISCC= 0
C=MAX(1,21)
IF(ZZ .LE. 0 .OR. ZZ.GT.C2) RETURN
CM=C2-ZZ
XP=CM**3+CM**2+CM+1
YP=CM**3+CM**2+CM+1
ISCC=3
RETURN
END

SUBROUTINE PSTART(M)
COMMON/VICE,DEVC,PCX,PCY,PSFT, COLOR, CLRBCX, PSIZE,LPE(N)
COMMON/SINDO,YMIN,XMIN,YMIN,XMAX,YMAX
COMMON/AREA,YMIN,XMAX,YMIN,YMAX,F5:T
CHARACTER DEVC*,
CHARACTER COLOR*, CLRBCX*, PSIZE*,
IF(M .LE. 1 .Or. M .EQ. 2) Then
C . . OUTPUT TO SCREEN OR PLOTTER...
DEV=DEVC
DEVP=DEVC
IF(M .EQ. '10') CALL UDEVICE ( 'EG4' )
IF(M .EQ. '16') CALL UDEVICE ( 'EG4' )
IF(M .EQ. '12') CALL UDEVICE ( 'EG4' )
IF(M .EQ. '10') CALL UDEVICE ('NORMAL')
IF(M .EQ. '16') CALL UDEVICE ('NORMAL')
CALL USEC (COLOR)
CALL USEC (COLOR)
CALL USEC (COLOR)
CALL USEC (COLOR)
CALL USEC (COLOR)
END

B14
CALL UP$, SET VERT, 40, PSFT)
CALL UP$, SET HOR, 201
CALL SET, 'HARD')
END IF
IF (IM .EQ. 3) THEN
OUTPUT TO PRINTER
DEV = DEV
CALL DEVC (DEV)
CALL OPSON ('FX80', 1.)
CALL USAP
CALL SET, 'PERCENT')
CALL UDATA(I0, TP, 10, 50.)
CALL WINDOW(XMIN, XMAX, YMIN, YMAX)
CALL SET, 'SOFT')
CALL UP$, SET VERT, 30
CALL UP$, SET HOR, 2:
RETURN
END IF
RETURN
END
SUBROUTINE PEND
CALL UFLUSH
CALL UEND
RETURN
END
SUBROUTINE PUAIN
CHARACTER ID*
IF (ID .EQ. 2) RETURN
CALL UDEF
CALL UMOVE(S00., 500.)
CALL UCOPY(X, Y, IP)
IF(IP .EQ. 'E') THEN
CALL UMOVE(00., 00.)
CALL UPRINT('"+TEXT")
CALL UERASE
RETURN
END IF
RETURN
END
SUBROUTINE INF (NC)
CHARACTER FILE*4, BORD*2, HONOR*2, TYP*4
CALL UMOVE(810., 500.)
X=220.
IF (NC .GT. 26) X=810
IF (NC .LT. 26) X=NC*25.+100.
IF (X .LT. 550.) X=550.
CALL UMOVE(X, Y)
CALL UPRINT('"CONTOURS","TEXT")
RETURN
END
SUBROUTINE PROFILE(ZMX, ZMN, MM, XMINX, XMAXX, YMINY, YMAXY, SM, ROASK)
INTERFACE VERTER
COMMON/COLOR/MX(999), Y(999), X(999), Z(999), CX(3), CY(3), CZ(3),
+ COMMON/DEV/DEV, PCY, PCT, PSF, COLOR, CLRBC, PSIZE, LPEN(8)
COMMON/TRL/RBL, MB
COMMON/CO/L2, COI, CO2, CO4, CO8, CO2, CO4, CO8, CO2, CO4, CO8
COMMON/WINDW/XMIN, XMAX, YMIN, YMAX
COMMON/DATE/ID, XMIN, YMAX, YMIN, YMAX, FSOFT
DIMENSION XI(260), YI(260), ZI(260)
CHARACTER COLn1, COLn2, COLn3, COLn4
CHARACTER DEV*, COLOR*, CLRBC*, PSIZE*, I
CHARACTER MINMAX*, PFILE*, DF
FUNCTION IMAX
IMAX = 1
IF (IP .EQ. 'A') IMAX = 1
IF (IP .EQ. 'B') IMAX = 2
RETURN
END
PRINT*, 'Profile line endpoints to be located:',
PRINT*, 'AY AY THE CURSOR (I) OR FROM A NAMED FILE (*).',
READ(A1, JASK)
IF (JASK .EQ. 'A') JASK = 'C'
B15
PRINT 'ENTER NAME OF FILE WITH X1,Y1,X2,Y2'  
READ 'A'(FILE,PFIL,ERR)  
PRINT 'FILE EQ. 'CON'T THEN  
G0744  
OPEN 10,FILE=PFIL,ERR=3334,STATUS='OLD')  
READ X1,Y1,X2,Y2  
Y1=Y1-SUB(Y1)/SM+100.  
X2=X2-SUB(X2)/SH+100.  
Y2=Y2-SUB(Y2)/SM+100.  
CALL UCRCLE(X1,Y1,5.)  
CALL UCRCLE(X2,Y2,5.)  
GO TO 7755  
    CALL CLEAR  
    PRINT '(AT) WANT CURSOR TO INDICATE NEAREST (TO UPPER '  
    PRINT 'RIGHT) DATA NODE POINT? (Y/N)')  
    CALL CLEAR  
    PRINT** 'POSITION CURSOR AT 1ST END OF PROF. LINE & HIT RETURN'  
    USE (...) AND 6 KEYS TO 'FIND' POSITION CURSOR.  
    CALL UNOVEV100.100.)  
    CALL UBELL  
    CALL KURIN(X1,Y1,IP)  
    CALL UCLOSE(X1,Y1)  
    CALL CLEAR  
    PRINT** 'POSITION CURSOR AT 2ND END OF PROF. LINE & HIT RETURN'  
    CALL USE(COL1)  
    CALL UCRCLE(X1,Y1,5.)  
    CALL UFLUSH  
    CALL UBELL  
    CALL KURIN(X2,Y2,IP)  
    CALL UCLOSE(X2,Y2)  
    CALL UFLUSH  
CONTINUE  
CALL CLEAR  
PRINT 'X1.E0.X2.AN0.Y1.E0.Y2) GO TO 7902  
CONTINUE  
    IF ABS(X2-X1).GE.0.1 THEN  
        IF (Y2.LT.X1) THEN  
            Y2=Y2+0.1  
            X1=X1  
            Y1=Y1  
            GO TO 123  
        ELSE IF (ABS(X2-X1).LT.0.1.AND.Y2.LT.Y1) THEN  
            X2=X2  
            X1=X1  
            Y1=Y1  
            ENDIF  
        ENDIF  
    CONTINUE  
    IF (X1.LT.1041.2  
        IF (Y1.EQ.0.0.AND.X1.EQ.0.0) GO TO 335  
        IF (Y2.EQ.0.0)  
            IF (X2.EQ.0.0)  
                IF (X1.EQ.0.0) THEN  
                    X1=1+X2  
                    Y1=1+Y2  
                    ELSE IF (X2.EQ.0.0)  
                ENDIF  
            ENDIF  
        ENDIF  
    ENDIF  
B16
READ '(!AI)\',FILEN.
IF(FILEN.EQ. 'NO_FILE') THEN
OPEN(IUNIT,FILE=FILEN,ERR=610)
GO TO 90
37 PRINT*,'FILEN ALREADY USED-- HIT RETURN, THEN TRY A NEW NAME'
   CALL UBEll
   CALL UPAUSE
   GO TO 90
89 CALL CLEAR
   PRINT '(',2,36)',ENTER A ONE LINE DESCRIPTION (OR RETURN)'
   READ '(!AI)\'.DESC
   CALL CLEAR
   PRINT '(A1)', '(2.29) ENTER SECTION LABEL (LIKE A FOR A-A)'
   WRITE(IP, '(2A1)') IF,SLASH
   PRINT '(',2,36),ENTER Y-AXIS LABEL (DEFAULT=ELEVATION)'
   READ '(!AI)',YLABEL
   WRITE('(',2,36),YLABEL
   VX='FIRST, IF (X-Y),TX(1),TY(1),ZT(1)
   WRITE(4, '(*16X)') VX,VY
   CALL MOVE(X1-20,Y1)
   CALL UPRNT1(IP,TXT)
   CALL MOVE(X2+20,Y2)
   CALL UPRNT1(TPP,'TEXT')
   CALL MOVE(X1-10,Y1)
   WRITE(4,'(A70)') DESC
   WRITE(4,'(A40)') YLABEL
   DO 605 = 1,N
605 WRITE(4,'(12X)XT(I),YT(I),ZT(I)
   IF(TASK.EQ.'F') GO TO 7902
END CALL CLEAR
12 FORMAT(8H,MIN EL,F8.2,8H,MAX EL,F8.2,8H-ALL POINTS ON PLOT.
+ / / BH
6634 PRINT '*', '(2.31) ENTER YOUR MIN AND MAX PROFILE ELEVATIONS.'
   PRINT '(A1)', 'DETERMINE MIN AND MAX ELEV OF ALL POINTS ON PLOTS')
   READ('(!AI)',MINMAX)
   IF(MINMAX.EQ.'F') THEN
      ZMIN=ZMN
      ZMAX=ZMX
   IF(MINMAX.EQ.'M') THEN
      OPEN(IUNIT,FILE='M44243'
   WRITE(4, '(A40)') MINMAX
   READ(4,'(43,434)') ZMIN,ZMAX
   CLOSE(43,STATUS='DELETE')
   END IF
   CALL CLEAR
   7902 CONTINUE
   IF(MM.EQ.2) THEN
   CALL FORM.
   DEVC,HP
   CALL PSTART(2)
   CALL UERASE
   CALL UERASE(4)
   CALL UAREA(10.'PCX,75.'PCX,10.'PCY,75.'PCY)
   IF(TASK.EQ.'Y') THEN
   CALL USEI(COL8)
   CALL UEPEN(1)
   CALL UEPEN(2)
   CALL UEPEN(3)
   CALL UEPEN(4)
   RASK= 'W'
   CALL USEI(COL1)
   ENDF
   IF(X.EQ.Y2.AND.Y1.EQ.Y2) RETURN
   CALL UMOVE(Y1-20.Y2)
   CALL UPRNT1(TPP,'TEXT')
   CALL UMOVE(X2+20.Y2)
   CALL UPRNT1(TPP,'TEXT')
   IF(IP.NE. 1) CALL UPRNT1('\'','TEXT')
B18
CONTINUE
XP=XNOW
YP=YNOW
COLOR=CLR
CALL USET(COLOR)
RETURN
ENDIF
COLOR=CLR
CALL USET(COLOR)
RETURN
END
SUBROUTINE CONS(TXX,YY,ZZ)
INTEGER SIDE, VERTEX
COMMON/COORD,NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
* SIDE(3000), S(3000), C(3000), X(3000), Y(3000), Z(3000),
* C(3000), XY(3000), XZ(3000), YZ(3000), CX(3), CY(3), CZ(3),
* XCY(3), YCZ(3), ZCX(3), XZC(3), YZC(3), C3, X3, Y3, Z3,
* XY3, YZ3, ZX3, XCY3, YCZ3, ZCX3, XZC3, YZC3)
C(3000), XY(3000), XZ(3000), YZ(3000), CX(3), CY(3), CZ(3),
* XCY(3), YCZ(3), ZCX(3), XZC(3), YZC(3), C3, X3, Y3, Z3,
* XY3, YZ3, ZX3, XCY3, YCZ3, ZCX3, XZC3, YZC3)
RETURN
END
SUBROUTINE OUTPRO(MM)
COMMON/CONT,DEV,PCX,PCY,PSFT,COLOR,CLRBACK,PSIZE,LPEN(8)
CHARACTER FILEN(12),ASK(1),VIEW(8)
DATA OLDPRO,D.PRO
PRINT(A1),
+ (2.35) WANT TO DRAW A PREVIOUSLY SAVED PROFILE ? (Y/N) 
READ(A1), JASK
CALL CLEAR
SUBROUTINE FINPRO(MAX,MM)
COMMON/CONT,DEV,PCX,PCY,PSFT,COLOR,CLRBACK,PSIZE,LPEN(8)
CHARACTER FILEN(12),ASK(1)
PRINT(A1),
+ (2.35) WANT TO DRAW A PROFILE PLT ? (Y/N) 
READ(A1), JASK
RETURN
END
SUBROUTINE INPRO(MAX,MM)
COMMON/CONT,DEV,PCX,PCY,PSFT,COLOR,CLRBACK,PSIZE,LPEN(8)
CHARACTER FILEN(12),ASK(1),VIEW(8)
DATA OLDMESH,M.PRO
READ(A1), JASK
CALL CLEAR
PRINT(A1),
+ (2.23) WANT TO DRAW A PROFILE ? (Y/N) 
READ(A1), JASK
CONTINUE
IF(JASK.EQ.'Y') THEN
PRINT(2.40) ENTER NAME OF "MESH" FILE (DEF= '.OLDMESH')
ENDIF
RETURN
END
PRINT (',A1','); OR 0 TO QUIT
READ (',A12',')',FIMESH=0,EOF=0,OPTION=0)
READ (',A12',')',FIMESH=OLDMESH
OPEN 1, FILE=',FIMESH', STATUS='OLD', ERR=100
OLDMESH=,FIMESH
GO TO 103
100 PRINT*, 'FILE NOT FOUND. TRY AGAIN'
GO TO 102
103 READ (,111), SUBX, SUBY
READ (,112), FACX, FACY
READ (,113), YMIN, YMAX, YMIN, YMAX
READ (,114), XMAX, ZMIN, XMIN, XMAX
READ (,115), YMIN, YMAX, SM
READ (,116), NP, MD, MBL
DO 106 I=1, NP
105 READ (,111), X(I), Y(I), Z(I), NCC(I)
DO 106 I=1, NP
106 READ (,111), (VERTEX(I,J), J=1,3)
PRINT (',A1') + '2.4a) WANT TO REDRAW THE CONTOURS ? (Y/N)'
READ (,A1), RDASK
IF RDASK EQ 'Y' THEN CALL CLEAR
PRINT (',A1') + '2.4b) WANT THE VOLUME UNDER THE MESH ? (Y/N)'
READ (,A1), FASK
IF FASK EQ 'Y' THEN PRINT (',A4')
4455 PRINT (',A5') + '2.4c) ENTER REFERENCE "BASE" ELEVATION (DEF=0.0)
READ (A10), REFEL
IF REFEL.LE.0. THEN REFEL=0.0
WRITE (43, (A10)) REFEL
READ (43, (A10), ERR=4455) REFEL
CLOSE (43, STATUS=DELETE)
ER0=0.0
VOL=0.0
DO 540 I=MBL, MB
540 CONTINUE
CALL UDASM
CALL USTART(1)
CALL UDASE(ClRBCX)
C
DRAW POINT
CALL USET(COL2)
CALL UDUM
DO 1931 =1, NP
1931 CONTINUE
CALL USET(COL2)
CALL USET(VECT, 30.)
CALL USET(MODE, 15.)
CALL XYAS(XMIN, XMAX, YMIN, YMAX, SM, 1)
CALL USET(COLOP)
IF RDASK EQ 'Y' THEN CALL ROCON(ZMAX, ZMIN, 1, XMINX, XMAX, YMIN, YMAX, SM, X)
B21
READ *.AASK
GO TO 2302
ENDF
GO TO 2303
2302 CLOSE(F3,STATUS='DELETE')
GO TO 2301
2303 CONTINUE
CLOSE(F3,STATUS='DELETE')
7503 CONTINUE
CALL CLEAR
PRINT '(A1)',PAS
+ '(2,17) WANT TO SEE ELEVATIONS PLOTTED? (Y/N) ' 
READ *(A1),HELV
CALL CLEAR
PRINT '(A1)',PAS
+ '(2,17) LEAVE X-Y AXES ANNOTATIONS OFF (Y/N) ' 
READ *(A1),PAS
PRINT *
+ '(2,18) LEAVE, LOCATION,TICKS AT DATA PTS, OFF TOO? (Y/N)' 
READ *(A1),PAS
PRINT
IF(HELV .GT. 0) THEN
CALL UCHPEN(PEN(0))
DO 1923 I=1,MP
IF(PASC(NE) .NE. 'A') GO TO 1923
IF(NCC(I).LT.1000) GO TO 1923
NCC(I)=1000
CALL USET(COL(I))
IF(NCC(I).GT.1000) CALL USET(COL(I))
CALL UMOVE([1.0],Y(I),*FSOFT)
CALL UMOVE([1.0]-3.*FSOFT,Y(I))
PRINT '(A1)',PAS
ENDF
1923 CONTINUE
IF(FLAG.EQ.1) THEN
CALL UCOMDO(1.0.40.0,1280.0,0.0,PCX,75.*PCY)
1979 CONTINUE
CALL USET(COL(I))
IF(NCM.GT.52.AND.HEAVY.EQ.0.) NCM=52
IF(NCM.GT.52.AND.HEAVY.NE.0.) NCM=52*HEAVY/ZINT
IF(NCM.EQ.0.AND.HEAVY.EQ.0.) NCM=52
IF(NCM.GT.500) NCM=500
NCM=ZMINT
C(NC)=CL(NC)+ZINT*(NC-1)
IF(ZINT.GT.0.0.AND.CL(NC).GT.ZMAX) GO TO 778
IF(ZINT.LT.0.0.AND.CL(NC).LT.ZMIN) GO TO 778
NC=NC+1
GO TO 777
778 NC=NC+1
IF(NCM.EQ.1.AND.CL(1).GT.ZMAX) NC=8
IF(HEAVY.NE.0.) THEN
NC=NC+1
ENDF
7608 CONTINUE
GO TO 7607
7607 NC=NC+1
IF(NCM(1).NE.0) GO TO 159
CONTINUE
HEAVY=O.
ENDIF

READ ('(A20)', 'DEF:NONE')
THEN
CALL UCHEPEN(LPEN(3))
OPEN(12,FILE=TMPLT,STATUS='OLD',ERR=159)

READ(*,END=59) X7,Y7,IPN
X7=(X7-XMIN)/SM
X7=(X7-XMIN)/SM
Y7=(Y-YMIN)/SM
OPEN(12,FILE=TMPLT,STATUS='OLD',ERR=159)

JPN=-1
REAK (12,*)
*DE
X7=(X7-XMIN)/SM
Y7=(Y-YMIN)/SM

CALL UWHERE(X7,Y7)
ANG=ANG(X7-X7-227)

IF(ANG.LT.0) CALL UMOVE(X7,Y7)

CALL UMOVE(X7-1.*F0S01*SM(ANG),Y7-1.*F0S01*CO(ANG))
CALL UMOVE(X7-1.*F0S01*SM(ANG),Y7-1.*F0S01*CO(ANG))
CALL UMOVE(X7-1.*F0S01*SM(ANG),Y7-1.*F0S01*CO(ANG))
CALL UMOVE(X7-1.*F0S01*SM(ANG),Y7-1.*F0S01*CO(ANG))
CALL UMOVE(X7-1.*F0S01*SM(ANG),Y7-1.*F0S01*CO(ANG))
IPN=IPN
GO TO 49

IF(FLAG.EQ.1) THEN
CALL UAREA(XMIN,XMAX,YMIN,YMAX)
ENDIF
CALL UPAPE
CLOSE(12)
ENDIF
CALL UCHEPEN(LPEN(1))
XSF=FPT
YSF=FST
IF(W.EQ.1) XSF=1.
CALL USE('SOFT')
CALL USE('VERT',999,1,1,FST,FST)
IF(XMIN.NE.WP) CALL UMERE(XMIN,YMIN,YMAX,SM,1)
IF(XMIN.EQ.2.AND.IV.EQ.0) CALL USE('HARD')
READ(81) NUMCON
DO 744 I=1,NUMCON
READ(81) X1,Y1,X2,Y2
CALL UMOVE(X1,Y1)
CALL UMOVE(X2,Y2)

CONTINUE
HEAVY=O.
ENDIF

READ ('(A1)', 'DEF:NONE')
THEN
CALL CLEAR

READ ('(A1)', 'DEF:NONE')
THEN
CALL CLEAR

READ(*,END=59) M1,M2,M3,M4,M5,M6
*ZT1,ZT2,ZT3,ZT4
DO 543 I=1,NC

CONTINUE
GO TO 548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548

CONTINUE
GO TO 1548
1550 CONTINUE
1549 CONTINUE
CLOSE(61)
CALL CLEAR
PRINT "(Y/N)"
READ ('A1') ,IDRAW
IF (IDRAW.EQ. 'Y') THEN
CALL CLEAR
PRINT "(Y/N)"
READ ('A1') ,ISKIP
IF (ISKIP.EQ. 'Y') THEN
CALL UERASE
CALL UBACKEX(CLRBK)
ENDIF
DO 510 I=MBL,MB
10 = I
CALL COROR(I)
CALL UMOVE(I),CY(I)
CALL UPPEN(I),CY(I)
CALL UPPEN(I),CY(I)
510 CONTINUE
ENDF
....
DRAWING TITLE AND LEGEND........
C
IF(MM.EQ.1.OR.MM.EQ.2) THEN
CALL UDATA(10,0.,PCX,100.0.,PCY,100.0.,PCY)
CALL USETSET("SOFTWARE")
CALL USETSET("HORI",0.,PCY)
IF(MM.EQ.1) CALL USETSET("VERT",50.)
IF(MM.EQ.2.AND.MM.EQ.0) CALL USETSET("HARD")
ENDIF
IF(MM.EQ.3) THEN
CALL UDATA(10,1.,PCX,100.0.,PCY)
CALL USETSET("VERT",20.)
CALL USETSET("HORI",30.)
ENDIF
CALL USETSET("COL4")
CALL UMOVE(100)
WRITE(TITLE(1),*'(8AI)') (TITLE(I),I=1,48)
WRITE(TITLE(1),*'(8AI)') (TITLE(I),I=1,48)
CALL UPRNTITITLE, TEXT')
CALL CLEAR
PRINT "(Y/N)"
READ ('A1') ,ISKIP
IF (ISKIP.EQ. 'Y') THEN
GO TO 162
ENDIF
IF(ISKIP.EQ. 'Y') THEN
GO TO 162
ENDIF
YP=YP+2
CALL UMOVE(100)
IF(NUMH.EQ.1) NUMH=NUMH+1
ENDIF
8900 CONTINUE
IF(IPOW.GT.0) IPOW=IPOW
CALL UMOVE(0)
IF(IPOW.EQ.0) IPOW=IPOW
DO 8900 IF(IPOW.EQ.0) NUMH=NUMH-1
IF(IPOW.EQ.1) NUMH=NUMH-1
IF(IPOW.EQ.2) NUMH=NUMH-1
IF(IPOW.EQ.3) NUMH=NUMH-1
ENDIF
CALL UMOVE(0,0)
IF(IPOW.EQ.0) IPOW=IPOW+1
IF(IPOW.EQ.1) IPOW=IPOW+1
IF(IPOW.EQ.2) IPOW=IPOW+1
IF(IPOW.EQ.3) IPOW=IPOW+1
8900 CONTINUE
IF(IPOW.GT.0) IPOW=IPOW
CALL UMOVE(0,0)
IF(IPOW.EQ.0) IPOW=IPOW+1
IF(IPOW.EQ.1) IPOW=IPOW+1
IF(IPOW.EQ.2) IPOW=IPOW+1
IF(IPOW.EQ.3) IPOW=IPOW+1
8900 CONTINUE
IF(IPOW.GT.0) IPOW=IPOW
CALL UMOVE(0,0)
IF(IPOW.EQ.0) IPOW=IPOW+1
IF(IPOW.EQ.1) IPOW=IPOW+1
IF(IPOW.EQ.2) IPOW=IPOW+1
IF(IPOW.EQ.3) IPOW=IPOW+1
ENDIF
ENDC
END
DATA COLOR('YELL', 'RED', 'GRNE', 'BLUE', 'YAN', 'MAGE', 'WHIT';
OPEN(1, FILE='MCIF.0AT')
   FORMAT(*4I5, A10)
   00 134. 15.50
   00 135. 1.999
   534 MARK(I) = 0
   00 333 = 1
   READ(9, *A10) IP
   READ(*, FACY, FACX)
   READ(*, SUBX, SUBY)
   CLOSE(8)
   NMPA=0
   INPUT=1

   300 CONTINUE
   91
   MIN=.9F30
   YMIN=.9F30
   XMIN=.9F30
   YMIN=.9F30
   YMAX=.9F30
   YMAX=.9F30
   READ(*, A12) 'FACY, FACX
   READ(*, A12) 'SUBX, SUBY
   END=DOE(I), 9(I), 2(I), NCC(I), ASPACE, ALABEL(I)
   IF(I(I).LT.9) THEN
     IF(X(I).GT.XMAX) XMAX=X(I)
     IF(Y(I).GT.YMAX) YMAX=Y(I)
     IF(X(I).LT.XMIN) XMIN=X(I)
     IF(Y(I).LT.YMIN) YMIN=Y(I)
   ELSE
     IF(900) THEN
       7779 CONTINUE
       PRINT *, 'MORE THAN 999 NODES.. IGNORING EXTRA NODES.. Hit RETURN'
       CALL UBELL
       CALL UPAXE
       GO TO 200
       END IF
     END IF
   END IF

   200 IP=CELL-NP/(2+10)
   IF(IP.EQ.999) IP=999
   PRINT *, 'IP', IPCELL=999
   PRINT *, '# NODES= ',NP, ' ......PRESS RETURN'
   CALL UBELL
   CALL UPAXE

   7789 CONTINUE
   GO TO 7779
   PRINT *, 'XMIN=', XMIN/FACX*SUBX, 'XMAX=', XMAX/FACX*SUBX
   PRINT *, 'YMIN=', YMIN/FACY*SUBY, 'YMAX=', YMAX/FACY*SUBY
   READ(*, A12) XMIN, XMAX, YMIN, YMAX
   PRINT *, 'ENTER NAME OF FILE GIVING XMIN,XMAX,YMIN,YMAX'
   READ(*, A12) FILEX
   IF(FILEX.NE.'Y') THEN
     IF(FILEX.EQ.'Y') THEN
       PRINT *, 'ENTER NEW XMIN AND XMAX'
       READ(*, A12) XMIN, XMAX
       PRINT *, 'ENTER NEW YMIN AND YMAX'
       READ(*, A12) YMIN, YMAX
     END IF
     GO TO 7779
   END IF
   OPEN(44, FILE=FILEX, STATUS='OLD', ERR=7789)
   READ(44, 7789) XMIN, XMAX, YMIN, YMAX
   CLOSE(44)
   XMIN=XMIN/SUBX*FACX
   YMIN=YMIN/SUBY*FACY
   XMAX=XMAX/SUBX*FACX
   YMAX=YMAX/SUBY*FACY

   7779 CONTINUE
   SY=(XMAX-XMIN)/900.
   SY=(YMAX-YMIN)/640.
   SM=SY
   SM=SM
   IF(M.EQ.1) CALL pstart(1)
   IF(M.EQ.2) DEV='EPS'
   IF(M.EQ.1) DEV='BMAH'
   B28
IF IX. EQ. 0. THEN
XH = IX - XTM(IK)**2 + YTM(IK)**2 / 20.
ENDIF
DO I = 1, IH
IF XH - XTM(IK)**2 - YTM(IK)**2 < 0. THEN
YH = YTM(IK)
ELSE
YH = YTM(IK) + 1
ENDIF
DO J = 1, JH
IF YH - YTM(IK)**2 - XTM(IK)**2 < 0. THEN
XH = XTM(IK)
ELSE
XH = XTM(IK) + 1
ENDIF
WRITE (19, *) XH, YH, Z(IK), NCC(I)
GO TO 1520
519 CONTINUE
WRITE (19, *) X(I), Y(I), Z(I), NCC(I)
GO TO 520
520 CONTINUE
END FILE 19
1817 READ (19, *, END=1818) X(NP), Y(NP), Z(NP), NCC(NP)
NP = NP + 1
GO TO 1817
1818 NP = NP + 1
WRITE (19, *) X(NP), Y(NP), Z(NP), NCC(NP)
GO TO 1817
NPP = NPP + 1
CALL USET(OL2)
DO J = 1, JH
NPP = NPP + 1
WRITE (19, *) X(NP), Y(NP), Z(NP), NCC(NP)
GO TO 1817
1201 I = I + 1
IF I EQ. 0. THEN
CONTINUE
ENDIF
1200 CONTINUE
CALL UFLUSH
CALL UAE LL
CALL UPAUSE
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL U?¶OVE (300., 300.)
DO 986 X=1, JCON
CALL U?PLL
CALL KURSAN (XP, YP, IP)
CLOSE (I28, STATUS='DELETE')
GO TO 1523
ENDIF
PRNT
LR
2
PRNT, (CURSORS_10áD)
WRITH
ROW
END
KEYS.
CALL CLEAR

* 70 ENTER # OF POINTS TO DESCRIBE AREA--(DEF=0) *'
PRNT(4,5)

* OR ENTER NAME OF A 'FAULT' FILE *'
READ (44, 'CONVAL ', IF CONVAL.EQ.1 . THEN OPEN(44, FILE=,'M033344')
WRITE(44, '(A40)', CONVAL,
READ (44, 999, ERR=6603) NFAL
CLOSE (44, STATUS='DELETE')
ENDF

6602 NCON=0
CLOSE (4, STATUS='DELETE')

6605 CONTINUE
IF (NCON.EQ.1 . OR NCON.EQ.2) GO TO 6601
IF (NCON.EQ.0) THEN OPEN (4, FILE=CONVAL, STATUS='OLD', ERR=6601)
NFAL=1
II=1
70 READ(44, END=71) XTM(II), YTM(II), IAKSHN(II)
IF (XTM(II).EQ.0.0 .AND. YTM(II).EQ.0.0) GO TO 70
XTM(II)=XTM(II)-SUBT*FACT
YTM(II)=YTM(II)-SUBT*FACT
IF (II.GT.11) YTM(II)=SHN+180:
II=II+1
GO TO 71

71 IEL=1
OPEN15, FILE='SKUNK')
NCON=1
DO 1404 I=1, IEL-1
IF (EO(II).EQ.0.1) IN=1
TH=IN
CALL USE(COL2)
DO 1405 I=1, TH
WP=NP
Z(NP)=Z(NP)+((YTM(I)-YTM(I+1)**2)/(20.)*TH(I))**2/20.
NC(NP)=NP
CALL UMOVE (NP, Y(NP))
CALL UMOV (NP, Y(NP)+1)
IF (I.AN.D. IAKSHN(II).EQ.0) MARK(NP)=1
IF (II.GT.1) AND IAKSHN(II).EQ.1 MARK(NP)-1
1405 CONTINUE
CALL USE(COL3)
1404 CONTINUE
CALL UDESH
CALL UPAUS
DO 521 I=1, NCON-1
DO 522 J=I+1, NCON
CALL AREA(1, XTM(I), YTM(I), XTM(J), YTM(J), A1)
IF (A1.GE.0.000001) GO TO 521
522 CONTINUE
WRITE (19, I1) I
521 CONTINUE
DO 734 I=NNPP, NP
ST=0.0
DO 620 J=1, NNPP
A2=0.0
IF (I.EQ.NP.OR.I.EQ.NP) GO TO 1411
DO 620 IF (I.EQ.1) . THEN IN=-1
DO 620 IF (I.EQ.1) . THEN IN=1
IF (MARK(I).EQ.0) 100=1
IF (MARK(I).EQ.1) 100=1
6302 CONTINUE
IF (I.GT.0.1) GO TO 1411
DO 620 IF (I.GT.0.1) . THEN IN=-1
CALL DLAREA(XTM(I), XTM(I+1), Y(J), YTM(J), YTM(I+1), Y(J), A1)
IF (A1.GT.0.0) GO TO 1411
B32
1400 CONTINUE
GO TO 1401
1401 CONTINUE
DO=111,1,5 (--*111+Y(1)***2)
IF (DO, EQ, 0.) THEN
Z(I)=Z(I)
GO TO 1407
ENDIF
ST=ST+Z(I)/DD
SB=SB.Z(I)/DD
1406 CONTINUE
Z(I)=ST/SS
1407 CONTINUE
GO TO 956
ENDIF
NUMC=NUMC
CALL CLEAR
PRINT *, 'AREA BY POSITIONING CURSOR AT LOWER LEFT OF:
PRINT *, 'POINTS (IN A COUNTERCLOCKWISE MANNER) AND PRESSING CR':
PRINT *, 'CURSOR IS MOVED WITH ARROW KEYS OR 6,( AND ) KEYS.
PRINT *, 'THE (.) KEYS GIVE FINE MOVEMENT':
CALL MOVE(300,300)
DO 2086 I=I,NUMC
CALL KURST(NPX,YP,IP)
DCL=999.630
DO 2089 J=1,NP
IF (J.LT.1) THEN
IF (IP(J).GT.60) THEN
DO 2089 J=1,NP
DCL(J)=DCL(J-1)
ELSE
DO 2086 I=I,NUMC
2087 CONTINUE
CALL USE(COLL)
CALL USEV((LCON(I)+1),X(LCON(I)),Y(LCON(I+1)),COLOR)
CALL USEU(X(LCON(I)),Y(LCON(I+1)),COLOR)
CALL USEU(X(LCON(I+1)),Y(LCON(I+1)),COLOR)
CALL USEU(X(LCON(I+1)),Y(LCON(I+1)),COLOR)
2086 CONTINUE
DO 2089 J=1,19,NUMC
CALL DRIVE(LCON(J),X(LCON(J+1)),X(I),Y(LCON(J)),Y(LCON(J+1)),I)
IF (ABS(A).LT.0.001) GO TO 952
954 CONTINUE
WRITE('18',*) 1
952 CONTINUE
END IF 18
957 CONTINUE
READ(19,*) END=957
OPEN(19,FILE='SKUNK')
REBND 19
976 READ(19,*,END=958) I
DO 958 J=I,NUMC
CALL DRIVE(I,LCON(J),X(LCON(J+1)),X(I),Y(LCON(J)),Y(LCON(J+1)),I)
IF (ABS(A).LT.0.001) THEN
WRITE('19',*) 1
GO TO 976
ENDIF
957 CONTINUE
GO TO 976
958 CONTINUE
REBND 19
952 READ(19,*,END=954) I
LCON=LCON+1
NUMC=NUMC+1
GO TO 1952
1954 NUMC=NCON-1
REBND 19
CLOSE 11
END IF
OPEN 11,FILE='TAP.DAT'.STATUS='UNKNOWN'
READING
DO 779 1=1,NP
779 WRITE(11),X(I),Y(I),Z(I),NCC(I),ASPACE,ALABEL(I)
XPL=XMIN
YPL=YMIN
IF J MP .EQ. 1 .THEN
CALL CLEAR
PRINT '(A)'. ' WANT TO (C)lip OR (Z)oom AREA (DEF=N) ? (C OR Z) ' 
READ (A).ZOOM
IF ZOOM .EQ. 'C'.OR.ZOOM.EQ.'Z' .THEN
PRINT '(A)'. ' WINDOW FROM (F)ILE OR (C)ursor (DEF=C)' 
READ (A),A
IF PT=NE .F.' GO TO 8366
CALL CLEAR
PRINT '(A)'. ' FILE ',FORCFILE,' NOT FOUND..PRESS RETURN ' 
READ (A),A
GO TO 9367
9367 CONTINUE
READ(43,),XPL,YPL,WIDT
CLOSE(43),YPL=(YPL-SUBY-YMIN)/(SM)+100.
WIDT=WIDT/SM
YPR=YPL+WIDT
GO TO 8346
8346 CONTINUE
PRINT 'USE CURSOR TO LOCATE LOWER LEFT CORNER OF AREA'
CALL UMOVE(100.,100.)
CALL KURSIN(XPL,YPL,IP)
CALL UMOVE(XPL-10.,YPL)
CALL OPEN(XPL+10.,YPL)
CALL UPIN(XPL,YPL+10.)
CALL CLEAR
PRINT 'USE CURSOR TO INDICATE WIDTH (OR HEIGHT) OF AREA'
CALL UMOVE(XPL,YPL)
CALL KURSIN(XPR,YPR,IP)
8345 CONTINUE
IF ABS(YPR-YPR).LT.0.001 . YPR=YPR+1039./1039.
IF ABS(XPR-XPR).LT.0.001 . XPR=XPR+1039./1039.
UMAX=XPR
UMIN=XPR
VMAX=YPR
VMIN=YPR
ENDIF
CALL UEPASE
CALL UCLEAR
CALL UEND
ENDIF
2301 CONTINUE
ZMIN=ZMAX
DO 2307 I=1,NP
IF (Z(I).LT.ZWMAX.AND.Y(I).LT.YWMIN.AND.
+ X(I).LT.XWMAX.AND.Y(I).GT.YWMIN.AND.
+ Z(I).LT.ZWMIN).THEN
PRINT '(2.13)'.
2307 CONTINUE
ZMIN=ZMAX
END IF
PRINT 'INPUT .ZMIN,.MAX EL.=',ZMAX
PRINT ', OR NO CONTOURING (DEF=AUTO) (H FOR HELP)' 
READ (440),CONVAL
IF (CONVAL.EQ.'NONE' .OR.CONVAL.EQ.'N') CONVAL='999999.1.0.0'
IF (CONVAL.EQ.'H') THEN
PRINT 'FOUR NUMBERS ARE TO BE ENTERED(OR ENTER RETURN FOR AUTO)'
PRINT*, 'THE 1ST IS THE BEGINNING CONTOUR VALUE. THE 2ND IS A - OF -'
PRINT*, 'CONTOUR INTERVAL
PRINT*, 'THE "J" OK CAUSES ALL CONTOUR LINES WHICH ARE MULTIPLES.'
PRINT*, 'OF THIS VALUE TO BE DRAWN HEAVY (PLOTTER) OR ANOTHER.
PRINT*, 'COLOR (SCREEN). THE 4TH IS THE NO. OF LINES TO BE
PRINT*, 'DRAWN. 10 TIMES THE 4TH MEANS SPAN THE RANGE OF DATA.
PRINT*, 'E.L. 5020, 2, 10, 6 MEANS START AT 5020 AND DRAW THE'
PRINT*, '5020, 5018, 5016, 5014, 5012, AND 5010 CONTOURS WITH THE'
PRINT*, '5020, AND 5010 MULTIPLES OF 10). DRAWN BOLD.'
PRINT*, (AT, 'PRESS RETURN TO CONTINUE')
READ (14), 'PRRT' GO TO 2301
ENDIF
IF (CONVAL EQ ' ') THEN
   DO 230 2(ZMIN-ZINT)/13.
      BLOG=LOG10(DJ)
      LOG=LOG10(BLOG+1).
      IF (LOG.LT.0.0) LOG=LOG-1.
      JDEL=10.
   IF (JDEL.EQ. 5.) JDEL=10
   IF (JDEL.EQ. 1.1) JDEL=6.
   ZMIN=ZINT+0.0001*ZMIN-ZINT
   ZMIN=12.*ZINT+ZINT
   NCM=0
   GO TO 7503
ENDIF
OPEN (43,FILE='MT456P')
WRITE (43,'(A40)') CONVAL
READ (43,ERR=2302) ZMIN, ZINT, HEAVY, NCM
IF (HEAVY.EQ.0) GO TO 2303
IF (ZINT.EQ.0) THEN
   HEAVY LINE MULTIPLE MUST BE GREATER OR = TO INTERVAL'
2302 CLOSE (43,STATUS='DELETE')
GO TO 2301
2303 CONTINUE
CLOSE (43,STATUS='DELETE')
7503 CONTINUE
IF (EQ.EQ.1) MED=0
IF (MED.EQ.1) GO TO 7564
PRINT*, (AT, 'LEAVE BOUNDARY OF CONTOURED REGION UNDRAWN ? (Y/N)')
READ (14), 'PRRT' JASK
PRINT*, (AT, 'LEAVE LABELS PLotted? (Y/N)')
READ (14), 'PRRT' JASK
IF (JASK.EQ. 'Y') JELV=1
PRINT*, (AT, 'LEAVE X-Y AXES ANNOTATIONS OFF ?(Y/N)')
READ (14), 'PRRT' JASK
IF (JASK.EQ. 'Y') JLABEL=1
PRINT*, (AT, 'LEAVE LOCATION, TIC'S AT DATA PTS, DEF TO ? (Y/N)')
READ (14), 'PRRT' JASK
IF (JASK.EQ. 'Y') JELV=0
IF (JELV.EQ.1) JELV=1
IF (JELV.EQ.1) JELV=1
3200 CONTINUE
IF (JTEST.EQ.1) THEN
   PRINT*, (AT, 'WANT LABELS PLotted? (Y/N)')
   IF (JASK.EQ. 'Y') JLABEL=1
   ENDIF
PRINT*, (AT, 'LEAVE X-Y AXES ANNOTATIONS OFF ?(Y/N)')
READ (14), 'PRRT' JASK
IF (JASK.EQ. 'Y') JELV=0
IF (JELV.EQ.1) JELV=1
IF (JELV.EQ.1) JELV=1
B35
PRINT', '(C,19) ENTER TITLE OF PLOT--IF NONE, HIT RETURN'
READ', 'A48') TITLE
PRINT', '--PRESS RETURN AT BEEPS'
CALL UBEEL
CALL UPAUSE
CALL PEND
CALL PSTART
CALL USET('CD2)
CALL UDOOL,'Z') CALL UWINO(XPL,XPR,YPL,YPR)
FSOF=1
IF JNEW EO 0) THEN
CALL UHPIO(PEN(2))
DO 1929 I = 1, N
IF PAS(I) E . T E 1. THEN
IF NCC(I) LT.1000) GO TO 1929
ENDIF
CALL UMOVE(X(I-3,FSOFT,Y(I))
CALL UPEW(X(I+3,FSOFT)
CALL UMOVE(X(I),Y(I)+3,FSOFT)
1929 CONTINUE
CALL UHOME
CALL UBEEL
CALL UPAUSE
END
CALL USET('SOFT'
CALL UPSET('VERT',30.*PSFT*FSOFT)
CALL UPESET('HORI',15.*PSFT*FSOFT)
IF NASK.NE. 0) CALL XYAXIS(XMINX,XMAXX,YMINY,YMAXY,SM,O)
IF JNEW EO 1) THEN
DO 1929 I = 1, N
IF PAS(i) E . T E 1. THEN
IF NCC(I) LT.1000) GO TO 1929
ENDIF
CALL UMOVE(X(I)+6,FSOFT,Y(I)+6,FSOFT)
CALL UPEW(T['REAL')
1929 CONTINUE
CALL UBEEL
CALL UPAUSE
END IF
IF LABEL.EQ.1) THEN
DO 1223 I = 1, N
IF NCC(I) LT.1000) GO TO 1223
IF ALAB(EQ.1) THEN
BLAB(I) = READ('LABEL(I)', '(10A1)') (BLAB(J),J=1,10)
DO 1223 J=1,10,1 THEN
WRITE('LABEL(I)', '(10A1)') (BLAB(J),J=1,10)
GO TO 1229
ENDIF
1229 CONTINUE
READ LABEL(I)', '(10A1)' (BLAB(J),J=1,10)
GO TO 1226
LBLAB(I)
1226 GO TO 1227
1229 CONTINUE
IF MM.EQ.2) THEN
CALL UPESET('VERT',18.*PSFT*FSOFT)
CALL UPESET('HORI',15.*PSFT*FSOFT)
ENDIF
CALL UPRNT1(Label,'TEXT')
1223 CONTINUE
IF MM.EQ.2) THEN
CALL UPESET('VERT',30.*PSFT*FSOFT)
CALL UPESET('HORI',19.*PSFT*FSOFT)
ENDIF
CALL UBEEL
CALL UPAUSE
ENDIF
CALL USET(COLOR)
IF MM.EQ.11) THEN
CALL UCHEQ(PEN(3))
CALL UUH
!FN
AN
G=ATAN2Y
I
-X7-X77
XY'
I1JPN.LI
a
CALL UIOVI
!FN
IF
JPNGE:
CALL UPEN1ý'7Y
'CAL UOE
0
;77+1.XFSOF
*SifINING).
.Y77-1
+FSOFT'CSjANG11.O
IF3jLT
J
CALL UM2R
X
7
+1
'FSOFjS
JAN~
4
Y7-61 SOJCO
ALL UOTh7I.
FAL
0P
T
S
1
N
(C
I
G

JPN.1049
CALL UlLL
CALL PASSE
159 CO IND
UE
1
INUeI~T.
52AND.HEAVY.E0.O.1 NCM=52,
FNCI¶
61.82.AND.HEAVY
NE 0 )NCM:52
HEAVY/ZINT
FNCM.0.
AND. HEAVY.N
b
NCM=52,
IF
NCM.EQ.
0AN.EVNE.J
NCM=52
HEAVY/ZINT
LI
=:I
PEL
777 CLNC)-CL11]+ZINT*(NC-1)
IF
CL11.CL0.0.AND.CLNC).LT.ZMIN) 60 TO 778
IF
NC-1.ECNCH) 60 TO 778
NC=NC
778 IF
NC=NC-1
IF
NC=NC-1
IFNCN.T)
XLIAX=-99.E30
06 3
0:
FXLCON
T J 1TXtlIN
XMIN=X(ILCON II
ýF
XICON
I H.T.XMAX JXMAX=X
ILCONI
IF
YJLCON
LI[ YMIN YMIN:Y
LCON
IJ
iF
Y(LCON
(I T. YMAXJ YMAX=Y
LCON
Ii
693 CONTINUE
AVX= 'VII N+XMAX1 /2.
JID~o
CONTINUE
AVY= 'VII N+YMAX1 /2.
JID~o
DO 661 =1.NC
CONTINUE
HEAVY=0.
END IF
1159 CONTINUE
C *...BUILD CONVEX CIRCULAR CELLS..ABOUT NPCELL POINTS PER CELL.
IF
NCN.T) THEN
XMIN=99.E30
YMIN=99.E30
YMAX=99.E30
DO 693 =1.NC
IF(X(LCON[1].LT.XMIN) XMIN=X(LCON[1]
IF(Y(LCON[1].LT.YMIN) YMIN=Y(LCON[1]
IF(Y(LCON[1].GT.YMAX) YMAX=Y(LCON[1]
693 CONTINUE
AVX= 'VII N+XMAX1 /2.
AVY= 'VII N+YMAX1 /2.
JID~o
DO 661 =1.NC
G=X(LCON[1]-AVX)**2+(Y(LCON[1])-AVY)**2
T=20.ID J1D J1D=ID
661 CONTINUE

B37
COUNTER END

END

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

C

SAVE A MESH FILE IN FILE 'MESH.DAT'

C

GO TO 8881

CONTINUE

PRINT *(1,2.21) DO YOU WANT TO SAVE THIS MESH (SO THAT YOU MAY'

PRINT *(1,2.21) DO YOU WANT TO SAVE THIS MESH (SO THAT YOU MAY'

READ (A1,A2) TASK

READ (A1,A2) TASK

IF(TASK.EQ.'V') THEN

IF(TASK.EQ.'V') THEN

PRINT *,(2,22) ENTER A FILE NAME IN WHICH TO STORE THE MESH.'

PRINT *,(2,22) ENTER A FILE NAME IN WHICH TO STORE THE MESH.'

READ (A43) FILE

READ (A43) FILE

IF(FILE.EQ.'DEF.MESH') THEN

IF(FILE.EQ.'DEF.MESH') THEN

WRITE (*,*) 'DEF MESH:'

WRITE (*,*) 'DEF MESH:'

SIGNED = 'M...'...

SIGNED = 'M...'...

PRINT *,(2,23) PRINT 'IF YOU WISH TO EXIT THE PROGRAM, TYPE 'E'.'

PRINT *,(2,23) PRINT 'IF YOU WISH TO EXIT THE PROGRAM, TYPE 'E'.'

IF(TASK.EQ.'E') STOP

IF(TASK.EQ.'E') STOP

READ (A81) FILE

READ (A81) FILE

WRITE (*,*) 'EXIT...' FILE=FILE

WRITE (*,*) 'EXIT...' FILE=FILE

WRITE (81,FAC,FACY)

WRITE (81,FAC,FACY)
WRITE 15, (X(I),Y(I),Z(I),NCC(I))
CONTINUE
DO 3356 3357
WRITE 15, (VERTEX(I,J), J=1,3)
CONTINUE
OPEN 65,FILE='EDGECON',FORM='FORMATTED')
WRITE 65, (8) NNU
READ 65, END=1868) X1,Y1,X2,Y2
WRITE 65, X1,Y1,X2,Y2
CLOSE 65, STATUS='DELETE')
READ 65, (RA), END=1823) ABC
IF (ABC EQ 'AB') THEN
CLOSE 65, STATUS='DELETE')
GO TO 1824
READ 65, END=1859
REWIND 65
READ 65, (NA), END=1828) DUM
GO TO 1827
READ 65, (RA) IKM
GO TO 1842
READ 65, END=1843) X1,Y1,X2,Y2
WRITE 65, X1,Y1,X2,Y2
GO TO 1842
CLOSE 65, STATUS='DELETE')
READ 65, FILE='SAVCON')
REWIND 65
READ 65, END=1768)MN1,MN2,MN3,MN4,MN5,MN6
WRITE 65, MN1,MN2,MN3,MN4,MN5,MN6
GO TO 1767
CLOSE 65, END IF
CALL CLEAR
GO TO 8882
DRAWING TITLE AND LEGEND...........
IF (MM.EQ.1 OR MM.EQ.2) THEN
CALL UNWINO (1,100.0,980.)
CALL UNAREA(10,PCX.100.*PCY.100.*PCY)
CALL UPSET ('SOFTWARE')
CALL UPSET ('VER 30.*PSF')
CALL UPSET ('HORI 15.*PSF')
IF (MM.EQ.2) THEN CALL UPSET ('HARD')
ENDIF
ENDIF
IF (MM.EQ.3) THEN
CALL UNAREA (1,100.0,67.)
CALL UPSET ('SOFT')
CALL UPSET ('VERT 20.')
ENDIF
ENDIF
CALL UPSET ('COL4')
READ 15, (4BAI ) (TITTLE(I),I=1,48)
PRINT (4B7) TITTLE
WRITE 15, (4BI ) TITTLE
CALL UPON (TITTLE, TEXT)
CALL CLEAR
IF (AEGEQ) THEN
CALL CLEAR
ENDIF
IF JASK.EQ.'Y') GO TO 162
ICM=NC
KPOW=0
DO 8900 1=NC
IF (NUMH(I).GT.0) KPDP=KPDP+1
CONTINUE
IF (KPDP.GT.0) KPDP=KPDP+1
8900 CONTINUE
IF (KPDP.GT.0) KPDP=KPDP+1

CALL INF1POW
PS=1.25
IF (POW, L I, 26) YF= IPOW*25. + 50.
IF (YF.LT.500.) YF=500.
CALL UPRNT1(INF=1, 'TEXT')
CALL UPRNT1(INF=1, 'REAL')
INC=1
DO 500 IC=1, NC, INC
IF (NUMH J1 IC, EQ.0. AND. HEAVY.NE.0.) GO TO 500
YF=YN
PX=YN
ENDF
ACL=IC(IC)
CALL DMOVE(PX, YF)
KF=-26
IF (IC*IC.GT.26) KF=26
IF (HEAVY.EQ.0.) CALL UPRNTI(LET=IC+KF, 'TEXT')
IF (HEAVY.NE.0.) CALL UPRNTI(REAL, 'TEXT')
CALL UPRNTI(LET=IC+KF, 'TEXT')
CALL UPRNTI(REAL, 'TEXT')
ENDF
YF=YF-25.
500 CONTINUE
CALL UPRNT1
162 CONTINUE
GO TO 8881
8882 CONTINUE
C............CREATE PROFILES.................
IF (IC.EQ.1) DEV=100;
IF (IC.EQ.2) DEV=104;
CALL PSTART(MM)
CALL USET(COLOR)
CALL UFLUSH
CALL CLEAR
C............DRAW PROFILES...................
CALL USET(COLOR)
CALL UFLUSH
956 CONTINUE
CEND=09
CALL CLEAR
PRINT(99, A)
+"? 30 WANT TO GENERATE A TRIPLE TRIANGLE MESH ? (Y/N) ?
READ(222), JASK
IF (JASK.EQ.'Y') THEN
CALL CLEAR
PRINT(99, A)
+"? 31 LINEAR OR INVERSE SQUARE RULE ? (L OR I) DEF=L
READ(222), JASK
NPX=JASK
PRINT(99, A)
PRINT*,'NEW INPUT DATA FILE NAME IS 'TRIPLE'
PRINT*,...WAIT....
DO 516 I=MBl, MB
I=I
CALL CORCOR(I)
CALL DAREA(CX(1), CX(2), CY(1), CY(3), A)
IF (ABST=15) GO TO 956
VX=VX(CX(1)-CX(3))/3.
VY=VV(CY(1)-CY(3))/3.
SV=VX-CX(2)*VX+CY(1)*VV**2
SV=VX-CX(2)*VX+CY(1)*VV**2
SX=VX(1)-VX(2)+VX(3)-VX(1)
SX=VX(1)-VX(2)+VX(3)-VX(1)
IF (NPX.EQ.1) CALL CONS(VX, VV, V2),
WRITE(13, 1) VX, VV, VZ
516 CONTINUE
ENDFILE 11
REWIND 11
CLOSE 109
OPEN(13, FILE='TRIPLE')
WRITE(13, 10) A141
WRITE(13, 1) \"FACX,FACY
\"
READ (1,1)(FX,0.10,FX,XMIN)
READ (3,1)(FY,0.10,FX,YMIN)
WRITE (1,1)(FX,0.10,FX,YMAX)
DO 10 519
10 CONTINUE
END

READ (1,1)(FX,0.10,FX,XMIN)
READ (3,1)(FY,0.10,FX,YMIN)
WRITE (1,1)(FX,0.10,FX,YMAX)
DO 10 519
10 CONTINUE
END
iF \( x \leq 1.1 \) CALL UPRNT(1,10).
CALL UMOVE(1,30).
CALL USET('ENTER')
\[ \text{IF} \left[ \text{FACX} \leq 1.1 \right] \text{ CALL UPRNT(1,10).} \]
\[ \text{IF} \left[ \text{FACX} \geq 1.1 \right] \text{ CALL UPRNT(1,10).} \]
CALL UMOVE(X,770.)
CALL UMOVE(X,780.)
GO TO 70

CONTINUE
YMAX=(YMAX-YMIN)/SM+100.
NUM=YMAX/NUM
BLOG=LOG10(2D)
LOG=LOG
IF(DEL=.GT.5) ID=10
IF(DEL=.EQ.5) ID=4
YINT=IDEL/10. * LOG
YES=IDEL/5
YMM=YN INT/6+0.0001*YMIN/YINT
YY=ZMIN-YINT
40 YY+Y+YMM+100.
Y.Y.L.I.50 GO TO 40
\[ \text{IF} \left[ \text{FACX} \geq 1.1 \right] \text{ CALL UPRNT(1,10).} \]
\[ \text{IF} \left[ \text{FACX} \leq 1.1 \right] \text{ CALL UPRNT(1,10).} \]
CALL UMOVE(1,20.)
GO TO 40

CONTINUE
\[ \text{IF} \left[ \text{FLAG} .GT. 1 \right] \text{ THEN} \]
CALL UDAREA(XUMIN,YUMIN,YUMAX)
CALL UDELETE(XUMIN,YUMAX)
\[ \text{ENDIF} \]
\[ \text{RETURN} \]
\[ \text{END} \]

C SUBROUTINE DRWRF(MM)
COMMON/DEVICE/DEVC,PCY,PSP,PSFT,COLR,CLBK,PSIZE,LPEN16
CHARACTER COLOR, CLBK, SIZE, SEC2, DESC2, YLABEL2, YLAB2, X
CHARACTER DESCRIPTION2, SEC, SIZE, DESC, YLABEL, YLAB
CHARACTER IDF2, IDF, PNAME2, PNAME, NAME, NAME2, NAME3
CHARACTER PNAME2, PNAME3, PNAME2, PNAME3, NAME2, NAME3, NAME4
CHARACTER PNAME2, PNAME3, PNAME2, PNAME3, NAME2, NAME3, NAME4
DIMENSION X(260), Y(260), Z260, IBOX(80,80)
DATA X, Y, Z260, IBOX(80,80)
CALL UOPEN(LPEN11)
\[ \text{IF} \left[ \text{MM} .EQ. 0 \right] \text{ OR MM .EQ. 21} \text{ THEN} \]
\[ \text{CALL USPT('SOFT') \}} \]
CALL USET('VERT',30.*PSFT)
CALL USET('HORI',20.*PSFT)
\[ \text{ENDIF} \]
\[ \text{IF} \left[ \text{MM} .EQ. 2 \right] \text{ THEN} \]
\[ \text{CALL USPT('SOFT') \}} \]
\[ \text{NOTE} \]
\[ \text{END} \]
\[ \text{READ 1,1(VX1, Vy1) \}} \]
\[ \text{READ 1,1(A70,Y'DESC \}} \]
\[ \text{READ 1,1(A100)) \}} \]
\[ \text{WRITE DESC2, (A70,A1)) \}} \]
\[ \text{WRITE SEC2, (A1)) \}} \]
\[ \text{WRITE YLABEL2, (A1)) \}} \]
\[ \text{IMAX=-2*EDY} \]
\[ \text{ZMIN=-ZMAX} \]
\[ \text{YMIN=ZMIN} \]
\[ \text{ZMIN=ZMAX} \]
\[ \text{IMAX=ZMAX} \]
\[ \text{I=1} \]
\[ \text{READ (14,*,END=143)} \]
\[ \text{X(I),Y(I),I} \]
\[ \text{X(I)=X(I)} \]
\[ \text{Y(I)} \]

B42
IF(Y1=0.0) THEN
GO TO 141
ENDIF
IF(Z1.EQ.ZMIN) ZMIN=Z1
GO TO 141

CONTINUE
NJ=1
YSUB=0
CALL CLEAR
PRINT "FIRST PROFILE OF A SERIES OF PLOTS ? Y/N "
READ (A40) JASK
IF(JASK.EQ.'Y') THEN
GO TO 101
ENDIF
GO TO 141

101 BOX(I,J)=0
IF(JASK.EQ.'Y') THEN
CALL CLEAR
PRINT "ENTER DESCRIPTION (DEF=NO CHANGE) "
READ (A40) NEWLAB
IF(NEWLAB.NE.0) THEN
LABEL=NEWLAB
CALL CLEAR
ENDIF
IF(JASK.EQ.'Y') THEN
MSYM=MSYME+4
MSYM=MSYM+1
ENDIF
IF(MSYM.GE.16) THEN
CALL CLEAR
PRINT "THE PROFILE STRETCHED TO FILL PAGE ? Y/N "
READ (A11) JASK
ENDIF
CALL CLEAR
DO 201 I=1,NJ
200 Y(1)=Y(I)-YSUB*0.639
IF(JASK.EQ.'Y') XF=664.56/SORTI(X(NJ)-X(1))**2+(Y(NJ)-Y(1))**2
DO 201 J=1,NJ
201 Y(J)=Y(1)
CALL CLEAR
PRINT .SECT .SEC
PRINT 333,NJ,ZMIN,ZMAX
333 FORMAT(9H MIN EL= ,F8.2,H MAX EL= ,F8.2)
CONTINUE
IF(JASK.EQ.'Y') REFEL='S'
PRINT (A11) A
IF(JASK.EQ.'Y') THEN
" ENTER MIN AND ZMAX (DEF=AUTO) "
READ (A40) ,REFEL
XSET=200.
YSET=100.
YCAP=1.5
CALL CLEAR
IF(REFEL.EQ.'S') REFEL=OREFEL
AREFEL=REFEL
IF(REFEL.EQ.'B') THEN
BLOG=LOG10(2)
LOG=LOG10
LOG=LOG10(1.01) LOG=LOG(1)
DEL=IDEL
IF (IFGE 5) \( \text{ZGE} = 10 \)
IF (IFGE 0) \( \text{ZGE} = 0 \)
\( \text{ZINT} = \text{ZGE} \cdot 10 \cdot 10 \cdot 10 \)
\( \text{ZMIN} = \text{ZINT} \)
\( \text{ZMAX} = \text{ZMIN} + \text{ZINI} \)
IF (ZMAX.GT.ZMAX) GO TO 89
GO TO 88
ZMAX = ZMAX
ZPLOT = (ZMAX-ZMIN)/5.
GO TO 94
END IF
IF (RELATED.GE.1) THEN
OPEN (43. FILE = "MA4334"
WRITE (43. 'A40') REFEL
READ (43. END = 6772) ZMIN, ZMAX
CLOSE (43. STATUS = 'DELETE')
END IF
CONTINUE
IF (KASK EQ. 'Y') ZEFIL = 'S'
IF (KASK EQ. 'Y') PRINT (A40), ZEFEL
ENDIF
CALL CLEAR
IF (ZEFIL EQ. 'S') ZEFIL = 'ZEFIL
ZEFPL = 'ZEFPL
IF (ZEFIL EQ. 'S') THEN
ZPLOT = (ZMAX-ZMIN)/5.
GO TO 94
ENDIF
IF (ZEFIL EQ. 'S') THEN
OPEN (43. FILE = "MA4334"
WRITE (43. 'A40') ZEFIL
READ (43. END = 6772) ZMIN, ZMAX
CLOSE (43. STATUS = 'DELETE')
ENDIF
CONTINUE
IF (ZMAX-ZMIN)/ZPLOT.GT.5.05) THEN
CALL CLEAR
PRINT USING ',ZPLOT,' WILL PLOT TOO TALL. TRY A LARGER VALUE.'
GO TO 118
ENDIF
CALL CLEAR
IF (KASK NE. 'Y') THEN
PRINT (A1), 'WHAT PROFILE SHADING REMOVED Y/N ? (S FOR SAME)'
READ (A1), ISHADE
IF (ISHADE EQ. 'S') ISHADE = 0ISHADE
ISHADE = ISHADE
CALL CLEAR
IF (KASK NE. 'Y') THEN
PRINT (A1), 'LEVOFF = 'Y'
READ (A1), LEVOFF
ENDIF
IF (LEVOFF EQ. 'S') LEVOFF = 0LEVOFF
LEVOFF = LEVOFF
CALL CLEAR
8855 PRINT 'SYMBOL ON PROFILE LINE (DEF=NONE, S=SQUARE, C=CIRCLE'
PRINT (A1) , ' T=TRIANGLE, D=DIAMOND. ** )'
READ (A1), SYM
IF (SYM. EQ. ' ' OR SYM. EQ. 'S' OR SYM. EQ. 'C' OR SYM. EQ. 'T' OR SYM. EQ. 'D') GO TO 8866
CALL CLEAR
PRINT 'SYM, NOT A VALID SYMBOL....TRY AGAIN'
GO TO 8855
8866 CONTINUE
IF (SYM. EQ. ' ' ) THEN
CALL CLEAR
PRINT (A1), 'LEGOEND FOR SYMBOL ? (20 CHAR MAX)'
READ (A20), LEGSL
WRITE (A20, (A1)) LEGSL.
IF (KASK EQ. 'Y') THEN
CALL CLEAR
PRINT (A1), 'LEGEND FOR SYMBOL PRINTED ON LEFT (DEF) OR RIGHT ? L/R'

CALL CLEAR
ZDIF E0.0 RETURN IF (3KASi.EQ.0)'Y:' CALL UERASE
CAL LCLEAR IFL JKASK.EQ.0'Y: CALL UACKG (CLRBCK)
CALL UPEN (X11,Y11) =N
CALL UPEN (X12,Y12)
18=0 GO TO 400 IF (ZL.EQ.-999.OR.ZZ.EQ.-999) GO TO 402
SYM=SYM+1 IF (ZL.EQ.0)'Z: CALL UPEN^{-X12,Y12}+Z1+YSET)
G=I{G1G2SYM} G**2 S1**2 S1**2 G2**2 S1**2 S1**2 G2**2 S1**2 S1**2
CALL XPRNT (A.NREAL’) CALL UPEN (X11,Y11)
G=0.0 (G1+G1LAST)/2; (G2+G2LAST)/2; (SYM)
COUNT CONTINUE IF (S1.NE.’Y’) THEN CALL UQVE~(XI1+~)
COUNT CONTINUE IF (LEVOFF.NE.’Y’) THEN CALL UPEN (X11,Y11)
G=SORI(2*X(1)*Y(1)+Y(Y(1))**2)+S1**2 S1**2 G2**2 S1**2 S1**2 G2**2 S1**2 S1**2
CALL UMOVF (100, YSET)
CALL UPEN (X11,REAL’) CALL UMOVF (X12, YSET)
CALL UMOVF (X11,YSET)
CALL UPEN (X12,YSET) 25.0
500  CONTINUE
      CALL UMOVE(RI+60.,R2+20.)
      DPRINT(2)
      CALL UPRINT(DPRINT,'TEXT')
      DPRINT(2)
      CALL UMOVE(SORT(Y(1)**2+Y(1)**2)*XSET,YSET)
      CALL UWHERE(GI,G2)
      CALL UMOVE(GI-15.,G2-30.)
      IF(M.EQ.1) CALL UMOVE(GI-15.,G2-30.)
      IF(M.EQ.1) CALL UPRINT(LI,'TEXT')
      DPRINT(0)
      IF(Y(I-1).LE.100.) CALL UPRINT(DI,'INTE')
      DO 700 I=1,11
         CALL UMOVE(XSET+(I-1)*50/10.,YSET)
         CALL UMOVE(YSET+(I-1)*50/10.,YSET-10.)
         CALL UWHERE(GI,G2)
         IF(M.EQ.1) CALL UMOVE(GI-15.,G2-30.)
         IF(M.EQ.1) CALL UPRINT(LI,'TEXT')
      ENDF
      DPRINT(0)
      IF(Y(I-1).LE.100.) CALL UPRINT(DI,'INTE')
      DO 700 I=1,11
         CALL UMOVE(XSET+(I-1)*50/10.,YSET)
         CALL UMOVE(YSET+(I-1)*50/10.,YSET-10.)
         CALL UWHERE(GI,G2)
         IF(M.EQ.1) CALL UMOVE(GI-15.,G2-30.)
         IF(M.EQ.1) CALL UPRINT(LI,'TEXT')
      ENDF
      700 CONTINUE
      IF(Y(I).<=1.40) GO TO 850

500  CONTINUE
      I=40
      850 CONTINUE
      851 CONTINUE
      IF(Y(I-1).LE.100.) CALL UPRINT(DI,'INTE')
      CALL UMOVE(XSET+(I-1)*50/10.,YSET)
      CALL UMOVE(YSET+(I-1)*50/10.,YSET-10.)
      CALL UWHERE(GI,G2)
      IF(M.EQ.1) CALL UMOVE(GI-15.,G2-30.)
      IF(Y(I-1).LE.100.) CALL UPRINT(LI,'TEXT')
      IF(Y(I).<=1.40) GO TO 850
SUBROUTINE FENFILE(ZMN, ZMAX, X1, X2, Y1, Y2, PX, PSDX, ...)
    COMMON/GRAPH(X, Y, Z, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 VERTEX(2000), SIDE(2000), NP, 
                 COMMON/DEV, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
                 COMMON/GRAPH, X(99), Y(99), Z(99), CX(3), CY(3), CZ(3),
                 COMMON/GRAPH, VERTEX(2000), SIDE(2000), NP, 
                 COMMON/GRAPH, PCX, FSOFT, COLOR, CLRBACK, PSIZE, LPE
                 COMMON/GRAPH, COLO, COL2, COL3, COL4, CMAX, CMIN,
                 COMMON/GRAPH, YMIN, YMAX, YMIN, YMAX, FSQF, DIMENSION XT, YT, ZT,
\( Y[1] = -8; \)

\[ \text{IF} \ Y[1] \geq 0 \text{ THEN} \]
\[ \text{CALL AREA1(X[1], Y[1], Y[2]);} \]
\[ \text{CALL AREA2(X[2], Y[2], Y[3]);} \]
\[ \text{CALL AREA3(X[3], Y[3], Y[4]);} \]
\[ \text{IF} \ Y[1] \leq 0 \text{ AND } Y[2] \leq 0 \text{ AND } Y[3] \leq 0 \text{ THEN} \]
\[ Y[1] = -Y[1]; \]
\[ \text{ENDIF} \]

\[ \text{CALL USE(TCOL1)} \]
\[ \text{IF} \ MM \geq 1 \text{ THEN} \]
\[ \text{CALL UCIRCLE(X1, Y1, 5.0);} \]
\[ \text{CALL HPLUSH;} \]
\[ \text{IF} \ NJ \leq 0 \text{ THEN} \]
\[ \text{NJ} = 0; \]
\[ \text{GO TO 100;} \]
\[ \text{DO} \ I = 1, NJ \]
\[ \text{O1 = 1;} \]
\[ \text{O2 = 1;} \]
\[ \text{DO} 100 \]
\[ \text{O1 = 1;} \]
\[ \text{O2 = 1;} \]
\[ \text{DO} 210 \]
\[ \text{IF} \ LBB \geq LBJLBC \]
\[ \text{IF} \ ZT(1) \leq -999 \text{ THEN} \]
\[ \text{ZT(1) = -999; LFND = 1; GO TO 210;} \]
\[ \text{CALL UALPHA 180; \}} \]
\[ \text{DO} 200 \]
\[ \text{IF} IG = II.MB + II - 1 \]
\[ \text{IF} \ KB = KK \]
\[ \text{IF} \ KB = MB \]
\[ \text{CALL ECOMP1; \}} \]
\[ \text{CALL ARECI1(X1, Y1, Y2, Y3, Z[1], Z[2], Z[3]);} \]
\[ \text{CALL ARECl2(X1, Y1, Y2, Y3, Z[1], Z[2], Z[3]);} \]
\[ \text{CALL ARECl3(X1, Y1, Y2, Y3, Z[1], Z[2], Z[3]);} \]
\[ \text{IF} \ Y[1] \leq 0 \text{ AND } Y[2] \leq 0 \text{ AND } Y[3] \leq 0 \text{ THEN} \]
\[ \text{CALL UALPHA; \}} \]
\[ \text{IF} LFND \leq -998 \text{ AND } ZT(1) \leq -998 \text{ THEN} \]
\[ \text{LFND = 1; \}} \]
\[ \text{DO} 210 \]
\[ \text{IF} \ ZT(1) \leq -998 \text{ AND } ZT(1) \leq -998 \text{ THEN} \]
\[ \text{LBB = NO.; NJ = NO. \}} \]
\[ \text{GO TO 210; \}} \]
\[ \text{ENDIF \}} \]
\[ \text{IF} LTIME = 1 \text{ THEN} \]
\[ \text{IF} \ LFND \leq -998 \text{ THEN} \]
\[ \text{CALL CLEAR; \}} \]
\[ \text{IF} \ ISKIP = 1 \text{ THEN} \]

B48
DO 60 I=1,N2
   60 CONTINUE
   7902 CONTINUE

1900 CONTINUE
IF (MM.EQ.2) THEN
   CALL PENO,
   DEV='HP4'
   CALL PSTART(2)
   CALL UERASE
   CALL UBACKG
   CALL UCHPEN(I1PEN)
   CALL UAREA(XMIN,YMAX,YMIN,YMAX)
ENDIF
IF (MM.EQ.0) THEN
   CALL USECOLOR
   CALL UCHPEN(I1PEN)
   CALL UCHPEN(I1PEN)
   CALL UCHPEN(I1PEN)
   CALL UCHPEN(I1PEN)
   CALL USE('W')
   CALL USET(COL1)
ENDIF
CLOSE(91)
RETURN
C049
CALL USETCOLOR
RETURN
END

SUBROUTINE FENCE(XMAX,XMIN,XMIN,XMAX,YMIN,YMAX,SM,IW,KW)
INTEGER SI
COMMON/COUNT(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
+ VERTEX(2000,3),SIDE(2000,3),NP,
COMMON/DATA(3),M1,M2,M3,
COMMON/XMIN,XMAX,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
COMMON/GRID2,XMIN,YMIN,YMAX,
COMMON/AREA(XMIN,YMIN,100,0.0),FSTOP,
COMMON/SM,1W,KW,
COMMON/STAGE3,FACE,FACY,SUBX,SUBY,
IF (N.EQ. 1) FDASK='Y'
CALL CLEAR
PRINT '(A)', '2:24C'MAKE THE VOLUME UNDER THE MESH? (Y/N)'
READ (AIO), FDASK
IF (FDASK.EQ.'Y') THEN
CALL CLEAR
4455 PRINT '(A)', '##95455
*2:24C'ENTER REFERENCE "BASE" ELEVATION (DEF=0.0)'
READ (AIO), REFEL
IF (REFEL.NE.0.0) THEN
OPEN(43,FILE='M44554')
WRITE(43, (AIO)) REFEL
END
CLOSE(43,STATUS='DELETE')
END
ATOT=0.0
VOL=0.0
VOL2=0.0
DO 540 I=1,MBL_MS
CALL CORCOR(1)
CALL AREA(CX(1),CX(2),CX(3),CY(1),CY(2),CY(3),A)
XMIN=AY-MIN(A)/2.
ATOT=ATOT+A
ZAV=(CZ(1)+CZ(2)+CZ(3))/3.
VOL=VOL+(ZAV-REFEL)
VOL2=VOL2+(ZAV-ZMIN)
540 CONTINUE
REFE=REFE-(ZMAX-ZMIN)/(VOL2-VOL1)+ZMIN
END
NMAX=MAX(XMIN-ZMIN/ZAV,0.0)
YMIN=MIN(YMIN-ZMAX/ZAV,0.0)
YMAX=MAX(YMIN+100.0-ZAV,0.0)
OX=(XMIN-100.0-ZAV)*SM*SUBX+XMIN
OY=(YMIN-100.0-ZAV)*SM*SUBY+YMIN
OZ=(ZMAX-ZMIN)*SM
IF (FISHED. .NE.HIMESH) THEN
CALL UTRAN
CALL UBASE(CLRBCK)
HIMESH=FISHED
ENDIF
CALL UDAREA(XDMIN,XDMAX,YDMIN,YDMAX)
IF (N.EQ. 2) THEN
CALL PEND
ENDIF
CALL PSTART(1)
CALL UBASE(CLRBCK)
CALL USET(Color)
C
C END POINTS
GOT0 1938
1938 CONTINUE
CALL UBASE(CLRBCK)
CALL UDOUT
1938 CONTINUE
CALL USET(Color)
GO TO 1938
DO 1939 I=1,NP
CALL UMOVE(X(1:1)+3.*FSFHT,Y(1:1))
CALL UMOVE(X(1:1),Y(1:1)+3.*FSFHT)
1939 CONTINUE
CALL USET(Color)
XPL=XMN
YPL=YMN
IP=1
CALL CLEAR
IF (FZOOM.EQ.' ') THEN
PRINT \( 'A)!' \), \' WANT TO (C)LIPI OR (Z)OOM AREA (DEF=N) \( ?(C \text{ OR Z}) \) ?'
READ \( 'A)!' \), ZOOM
IF ZOOM.EQ. \( ' \text{C}' \) OR ZOOM.EQ. \( 'Z' \) THEN
CALL CLEAR
PRINT \( 'A)!' \), \' WINDOW FROM (F)ILE OR (C)URSOR (DEF=C) \?'
READ \( 'A)!' \), FORFILE
OPEN FORFILE=FORFILE, STATUS=\( 'OLD' \), ERR=9346
GO TO 8347
9367 CALL CLEAR
PRINT \( 'A)!' \), \' ENTER NAME OF FILE WHICH CONTAINS COORDINATES OF'
PRINT \( 'A)!' \), \' LOWER LEFT CORNER AND WIDTH OF WINDOW \?'
READ \( 'A)!' \), FORFILE
OPEN FORFILE=FORFILE, STATUS=\( 'OLD' \), ERR=9346
GO TO 9346
9346 CALL CLEAR
PRINT \( 'A)!' \), \' FILE', FORFILE, \' NOT FOUND ...PRESS RETURN \)?'
READ \( 'A)!' \), FORFILE
GO TO 9347
9347 CONTINUE
READ \( (\* \), XPL, YPL, WDT
WDT=(XPL-SUBX-XMINX)/SM+100.
YPL=(YPL-SUBY-YMINY)/SH+100.
XPL=XPL+XDT
YPL=YPL+YDT
GO TO 8345
8346 CONTINUE
PRINT \( 'A)!' \), \' USE CURSOR TO LOCATE LOWER LEFT CORNER OF AREA\,'
CALL UMOVE(XPL, YPL)
CALL KURSL(XPL, YPL)
CALL UPPM(XPL+10, YPL)
CALL UMOVE(XPL, YPL-10)
CALL UPPM(XPL, YPL)
CALL CLEAR
PRINT \( 'A)!' \), \' USE CURSOR TO INDICATE WIDTH (OR HEIGHT) OF AREA\,'
CALL UMOVE(XPL, YPL)
CALL UMKIN(XPR, YPR)
8345 CONTINUE
IFABS(XPR-XPL).GT.0.001 XPR=XPL+[780./1039.]*\{XPR-XPL\}
IFABS(YPR-YPL).GT.0.001 YPR=YPL+[780./1039.]*\{YPR-YPL\}
XWMIN=XPL
XWMAX=XPR
YWMIN=YPL
YWMAX=YPR
? \( 'C)' \) THEN
CALL UDIFF(XPR, XPR, YPR)
XDMIN=(XPR-1.)/10.0
XMAX=(XPR+1.)/10.0
YDMIN=(YPR-1.)/10.0
YMAX=(YPR+1.)/10.0
CALL UDAREA(XDMIN, XMAX, YDMIN, YMAX)
GO TO 8867
ENDIF
IFZOOM.EQ. \( 'Z' \) THEN
CALL UWINDOW(XPL, YPL, YPR)
FSOFT=\{XPR-XPL\}20.1039.
GO TO 1938
ENDIF
ENDIF
CALL UUPSET('\{WGT\}', \{SOFT\}, FSFT)
IF JsAKC.EQ. \( 'Y' \) THEN
IF (\*).GT.0 THEN
XMINMM=X2, X1, Y1, Y2, \text{RDASK, O, XMINX, XMAXX,}
YMINMM=\{SH, O, PIPTE\}
GO TO 1938
ENDIF
ENDIF
8867 CALL CLEAR
X1=O
Y1=0
LEN=LEN
XH=XH
ANG=ANG
PRINT \( 'A)!' \), \' ENTER PROFILE GRID BOX ORIGIN OF COORDINATES, ANGLE OF'
PRINT \( 'A)!' \), \' BOX TO HOH, LENGTH OF BOI & HIGHT \( (X, Y, \text{ANG, \_I) \text{ FOR SAME AS LAST TIME (DEF=AUTO) }\)'
READ \( 'A)!' \), FORFILE
CALL CLEAR
GO TO 8825
EOF: IF EXAG.EQ."" THEN
OPEN (34, FILE='PIF', STATUS='OLD', ERR=5569)
READ (34) XV, YV, ANG, XLEN, XHT
CLOSE (34)
GO TO 8868
END
OPEN (35, FILE='MJOB4')
WRITE (35, 12X, AGOT) PIF
REWRITE 35
READ (35) ERR=8863) XV, YV, ANG, XLEN, XHT
OPEN (35, 15F15.4) XV, YV, ANG, XLEN, XHT
CLOSE (35, STATUS='DELETE')
GO TO 8868
OPEN (36, FILE='PIF', STATUS='OLD', ERR=8869)
READ (36) ERR=8869) XV, YV, ANG, XLEN, XHT
CLOSE (36, STATUS='DELETE')
Call CLEAR
GO TO 81
OPEN (36, FILE='PIF', STATUS='OLD', ERR=8869)
READ (36) ERR=8869) XV, YV, ANG, XLEN, XHT
CLOSE (36)
CONTINUE
AC=(XV-SUBY-XMINX)/SM)*100,
P1=YA
P2=YB
ANG=ANG/57.2958
P3=(XV-SUBY-XMINX+XLEN)/SM)*100.
PE=P5
P6=(YV-SUBY-YMINY+XHT)/SM)*100.
PE=PE
CALL ROTATE(P1, P2, P3, P4, P5, P6, ANG)
CALL SUBP (P1, P2, P3, P4, P5, P6, XLEN, XHT)
8855 FORMAT (I2,2x,F12.2,SH T1=,F12.2,SH ANG=,F6.2,3H L=,F10.2,
+SH H=,F12.2)
READ (A1), FTYPE
CALL CLEAR
IF (EXAG.EQ."" THEN
CALL USEF (1,A1)
CALL USEP (P1, P2, P3, P4, P5, P6)
CALL USEP (P1, P2)
CALL USEF (A1)
GO TO 8869
19 PRINT", '(2,42) WANT PARALLEL (TO BASE) OR NORMAL GRIDS ?'
PRINT (A1), 'INPUT P OR N (DEF=F)'
READ (A1), FTYPE
CALL CLEAR
IF (FTYPE.EQ."" THEN
FTYPE='P'
5578 PRINT (A1), 'C' AND FTYPE.NE.'W') GO TO 19
READ (A1), EXAG
IF (EXAG.EQ."" THEN
MANY=11
OPEN (43, FILE='M3443')
WRITE (43, A1) EXAG
REWIND 43
READ (43) ERR=5578) MANY
CLOSE (43, STATUS= 'DELETE')
EXIT IF (MANY.NE.0) THEN
PX=MIN (XLEN/5, XHT/5)/MANY
CALL CLEAR
PRINT (A1)', (2,44) ENTER SCALE EXAG. FACTOR (DEF=1.0)
READ '('A0')', EXAG
READ('EXAG', '('F10.0')', EX
IF(EX.GE.0.) EX=EX+1.
ENDF
IF(EX.GE.0.1) THEN

PRINT*, 'MIN. EL= ', ZMIN., ' MAX EL= ', ZMAX, ' FOR PROFILE SCALES'
PRINT*, 'A0', ', ENTER NEW MIN & MAX EL'S. (DEF=NO CHANGE)':'
ENDIF
READ (A0), NEWZ
IF(NEWZ NE.0.) THEN
OPEN(43,FILE='M345SR')
WRITE (43,'(A40)') NEWZ
ENDIF
ENDF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDER
SUBROUTINE XYMBOL(X,Y,SYM)
CHARACTER*1 SYM
IF SYM.EQ.' ' THEN
END IF
CALL CLEAR
CALL UDELETE
CALL UBACKG(CLRBCK)
RETURN
END

C
CALL LINE(SYM.X,Y,1,6)
CALL LINE(SYM.X,Y,1,9)
CALL LINE(SYM.X,Y,2,9)
ENDIF

IF (SYM.EQ. 'S') THEN
CALL LINE(SYM.X,Y,1,6)
CALL LINE(SYM.X,Y,1,9)
CALL LINE(SYM.X,Y,2,1)
ENDIF

IF (SYM.EQ. 'T') THEN
CALL LINE(SYM.X,Y,1,6)
CALL LINE(SYM.X,Y,1,9)
ENDIF

IF (SYM.EQ. 'C') THEN
CALL CIRCLE(X,Y,2,5)
ENDIF

IF (SYM.EQ. 'D') THEN
CALL LINE(SYM.X,Y,2,6)
CALL LINE(SYM.X,Y,2,8)
CALL LINE(SYM.X,Y,5,8)
ENDIF
RETURN
END

SUBROUTINE LINE(SYM.X,Y,I,J)
CHARACTER SYM(8)
DATA X(-2.5,-2,2.5,2.5,2.5,-2,2.5,2.5)/
W=1.0

IF (SYM.EQ. 'D') W=1.25
CALL UMOVE(X,Y)
DO 100 K=1,3
CALL UMOVE(X*W'U(K),Y*W'V(K))
100 CALL UPEN(F*W'U(K),F*W'V(K))
RETURN
END

SUBROUTINE ROTATE(X,XA,YA,ANG)
XN=(X-XA)*COS(ANG)+(-Y-YA)*SIN(ANG)
YN=(X-XA)*SIN(ANG)+(-Y-YA)*COS(ANG)
X=XN+XA
Y=YN+YA
RETURN
END
APPENDIX C. FORTRAN LISTING FOR McGRID

DIMENSION(12000), (1,2000), (1,200), (1,200), (1,200)

CHARACTER(100), (300), (300), (300), (300)

SUBSEQUENT ROUTINE:

PRINT 'This is McCon--a general contouring code.'
PRINT 'This section of the contour package provides several options for manipulating your data: Essentially, this routine allows you to name an input data file and then create a file named "map.dat" that is used by the subsequent contouring routines. If the "map.dat" file you want was just previously created and you want to go directly to 'contouring using that file then press "C". Now to go to contouring...

PRESS ANY KEY TO CONTINUE

READ (A), IASK

A word about the input file: This routine will ask you for the name of a data file which you have prepared for input. This data must be either freeform or columnar. The data must all be numeric (no letters anywhere). Each line of the file must contain the x & y coordinates of a data point. The line may also contain multiple data points. The routine prints the first few lines.

READ (A)

If the file is "freeform"...

There must be the same number of entries on each line. Each entry must be separated by a comma or by one or two blanks.

READ (A)

If the file is "columnar"...

All of the data columns must be "right-justified". Suppose that the data looks like...

1303 1.2 34.5 22 156 0 123.4

If the file contains fields that are blank (5th column of 2nd line), then the information in that line is ignored. If you are doing an operation that involves...
PRINT: 'THAT FIELD (LIKE SPECIFYING THAT THE 5TH COLUMN IS
PRINT: THE ELEVATION TO BE CONToured).
PRINT: (A))', 'PRESS ANY KEY TO CONTINUE :
READ (A), IASK
PRINT: 'ONE MORE THING: IF THE IDENTIFIER (WHEN SPECIFIED) IS
PRINT: REPEATED, ONLY THE LAST LINE WITH THE SAME VALUE IS
PRINT: RETAINED.'
PRINT: (A), 'PRESS ANY KEY TO CONTINUE :
READ (A), IASK

17 PRINT: :
PRINT: ' ENTER (ORIGINAL) DATA FILE NAME (DEF=MCDMO.DAT) :
READ (24), FILLE
OPEN (0,FILLE;STATUS='OLD';ERR=6)
GO TO 7
PRINT: 'FILE ',FILLE,' NOT FOUND....TRY AGAIN.'
GO TO 12

7 CONTINUE:
CLOSE(10;STATUS='DELETE')
OPEN(10;FILE='TEMP';STATUS='NEW')
READ (10) ISG
IF (ISG.EQ.,') Is FOR=1
PRINT: 'WANT INVERSE POWER GRIDING ?(Y,N OR ?) :
READ (A), ISG
IF (ISG.EQ.,)' THEN
PRINT: 'THIS OPTION WILL CREATE A RECTANGULAR GRID OF DATA:
PRINT: POINTS WHICH WILL ENCOMPASS THE ORIGINAL POINTS WITH:
PRINT: A SMALL BORDER. THE CONTORABLE VALUE (ELEVATION) OF
PRINT: THE GRID POINTS IS DETERMINED BY THE INVERSE POWER
PRINT: AVERAGE OF ALL ORIGINAL DATA POINTS THE ORIGINAL DATA
PRINT: ARE NOT RETAINED (UNLESS YOU REQUEST SUCH). THEREFORE.
PRINT: THE EFFECT OF THE AVERAGING WILL SMOOTH THE DATA
PRINT: THIS OPTION IS USEFUL FOR SPARSE, CLUSTERED DATA POINTS:
PRINT: IT IS NOT RECOMMENDED FOR CASES WHERE YOU HAVE GOOD
PRINT: SPATIAL COVERAGE.
PRINT: 'THE FORMULA FOR INVERSE POWER AVERAGING IS:
PRINT: Z(0)=SUM(Z(i)/(1-D(i)**P)) / SUM(1/(1+D(i)**P))
PRINT: WHERE Z(0) IS THE SOUGHT ELEVATION AT GRID POINT 0;
PRINT: Z(i) IS THE ELEVATION AT AN ORIGINAL POINT i;
PRINT: D(i) IS THE DISTANCE FROM 0 TO i.
PRINT: P IS THE POWER (USUALLY 2).
GO TO 17
END IF
IF (ISG.EQ.'N' OR ISG.EQ.'?') GO TO 814
PRINT: ' WANT ORIGINAL DATA POINTS AS IS' ? (NO FILLING
PRINT: 'OUT TO EDGES OF PLOT) ?(Y/N) :
READ (A), IASK

814 CONTINUE
IF (ISG.EQ.'Y') OR (ISG.EQ.'?') THEN
END IF
IF (ISG.EQ.'N') THEN
PRINT: ' ENTER VALUE FOR Q (POWER FOR INVERSE FITTING) DEF=2 :
READ (A5), DEFAULT
CALL DEFAULT.NMBR.2)
END IF
IF (ISG.EQ.'Y') OR (ISG.EQ.'?') THEN
I A S K
I F I A S K.E Q. ' Y ' O R I S G . E Q. ' ? ' I C B O R = ' N '
C2
IBOR=0
IF(ICHOR.EQ.'Y'.OR.IBOR.EQ.'Y') IBOR=2
HONOR='Y'
IF(ISG.EQ.'Y'.OR.ISG.EQ.'Y') THEN
18 PRINT 'A'
  + (1,N) 'DO NOT HONOR ORIGINAL DATA POINTS (Y, N OR ?)'
  READ (44,IASK)
  HONOR='Y'
  IF(IASK.EQ.'Y'.OR.IASK.EQ.'Y') THEN
  PRINT
  'ORIGIINAL DATA POINTS, IF HONORED, WILL BE ADDED TO THE'
  PRINT
  'GRIDED POINTS, THIS MAY BE DISIRABLE TO AVOID'
  PRINT
  'SMOOTHING OF PEAKS AND VALLEYS WHICH OFTEN RESULTS FROM'
  PRINT
  'INVERSE POWER AVERAGING.'
  GO TO 18
  ENDIF
ENDIF
VORMAT='['
FORMAT='['
IF(FORMAT.EQ.0) GO TO 8447
8879 READ (99,'(400X,END=8877) DUM')
  READ(1000,A14) IFDUM(I),I=1,200
  GD(I)=1.200
  IFDUM(I).NE. ' ' THEN
8878 CONTINUE
GO TO 8879
8877 READ (99,'(400X,END=8877) DUM')
  READ(1000,A14) IFDUM(I),I=1,200
  GD(I)=1.200
  GO TO 8877
8876 PRINT('1X,'IFDUM(I),'(GX(I),I=1,80)
  IFDUM(I).NE. ' ' THEN
8450 CONTINUE
8451 CONTINUE
8440 LAST=I+1
   IF(I<0) THEN
   CONTINUE
8467 CONTINUE
8466 CONTINUE
   IF(I<2) THEN
   CONTINUE
8460 CONTINUE
   I=I+1
   IF(I>LAST) THEN
   CONTINUE
8447 CONTINUE

PRINT: 'FIRST FEW LINES OF DATA FILE 'FILLE.' LOOKS LIKE...

IF PREP EQ 1 THEN PRINT FORMAT 11, I=1, J=1
PRINT [10, 79A1], [10X(I), J=1:70]

REWIND 09
DO 666 J=1,6
READ(09, F(200)), END=667) DUM
       CONTINUE
666 CONTINUE

8899 KEEP(J) = 1.10
       IF(KEEP(J) .NE. 0) THEN
       PRINT ('A', '0.076) WHICH COL IS 'IDENTIFIER' ? (0 IF NONE. DEF=0) '
       READ (AS), DEFAULT
       CALL DEFINE(DEFAULT, NUMBER, 0)
       IF (DEF .EQ. 0) IDENT = 1
       IF (DEF .EQ. 0) IDENT = 1
       PRINT ('A', '0.076) DO NOT REMOVE DUPLICATE LABEL ENTRIES ? (Y/N) '
       IF (AS = 'Y') OR (ASK = 'Y') IELIM = 1
       PRINT ('A', '0.076) WHICH COL IS 'SELECTOR' ? (0 IF NONE. DEF=0) '
       READ (AS), DEFAULT
       CALL DEFINE(DEFAULT, NUMBER, 0)
       IF (DEF .EQ. 0) THEN
       PRINT ('A', '0.09) ENTER 1ST SELECTOR 'KEEP VALUE'(DEF=CR) '
       READ (A20), KEEP(J)
       6693 IF(KEEP(J) .NE. 0) THEN
              PRINT ('A', '0.09) ENTER NEXT SELECTOR 'KEEP VALUE'(DEF=CR) '
              READ (A20), KEEP(J)
       GO TO 6693
              END IF
              IF (KEEP(J) .NE. 0) THEN
              PRINT ('A', '0.10) WHICH COL IS 'X' ? (DEF=1) '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IX, 1)
              PRINT ('A', '0.11) WHICH COL IS 'Y' ? (DEF=2) '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IY, 2)
              PRINT ('A', '0.12) WHICH COL IS 'Z1' ? (-)SIGN IF DESIRED (DEF=3) '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IZ1, 3)
              PRINT ('A', '0.13) WHICH COL IS 'Z2' ?(FOR Z1+Z2)OR 0(DEF)WHEN DONE '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IZ2, 0)
              IF (ZE .NE. 0) THEN
              PRINT ('A', '0.13) WHICH COL IS 'Z3' ?(Z1-Z2+Z3)OR 0(DEF)WHEN DONE '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IZ3, 0)
              IF (Z3 .NE. 0) THEN
              PRINT ('A', '0.13) WHICH COL IS 'Z4' ?(DEF=0) (ETC) '
              READ (AS), DEFAULT
              CALL DEFINE(DEFAULT, IZ4, 0)
              END IF
              IF (IZMAX = MAX(ABS(I1), ABS(IY), ABS(IZ1), ABS(IZ2), ABS(IZ3), ABS(IZ4)))
IF (FREFOR.EQ.0.0) THEN
    IF (FREFOR.EQ.0.0) THEN
        IF (FREFOR.EQ.0.0) THEN
            IF (FREFOR.EQ.0.0) THEN
                IF (FREFOR.EQ.0.0) THEN
                    IF (FREFOR.EQ.0.0) THEN
                        IF (FREFOR.EQ.0.0) THEN
                            IF (FREFOR.EQ.0.0) THEN
                                IF (FREFOR.EQ.0.0) THEN
                                    IF (FREFOR.EQ.0.0) THEN
                                        IF (FREFOR.EQ.0.0) THEN
                                            IF (FREFOR.EQ.0.0) THEN
                                                IF (FREFOR.EQ.0.0) THEN
                                                    IF (FREFOR.EQ.0.0) THEN
                                                        IF (FREFOR.EQ.0.0) THEN
                                                            IF (FREFOR.EQ.0.0) THEN
                                                                IF (FREFOR.EQ.0.0) THEN
                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                                IF (FREFOR.EQ.0.0) THEN
                                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                                                IF (FREFOR.EQ.0.0) THEN
                                                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                                                                IF (FREFOR.EQ.0.0) THEN
                                                                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                                                                                IF (FREFOR.EQ.0.0) THEN
                                                                                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                                                                                                IF (FREFOR.EQ.0.0) THEN
                                                                                                                                                    IF (FREFOR.EQ.0.0) THEN
                                                                                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                                                                                        IF (FREFOR.EQ.0.0) THEN
                                                                                                                                                            IF (FREFOR.EQ.0.0) THEN
                                                                rende...
G0?
TO
.5".
F
-ND IF
58 CONTINUE
.PRINT,
.PRINT, 'OUTPUT DATA FILE IS ''MAP. DAT''
.PRINT, 'DATA POINTS WHICH ARE CLOSE TOGETHER MAY BE ASSIGNED:
.PRINT, 'TO A SINGLE POSITION (THE ELEVATIONS OF THOSE POINTS
.PRINT, 'ARE AVERAGED AND ASSIGNED TO THE SINGLE POSITION).
.PRINT, 'THE DEFAULT CHECKING DISTANCE BETWEEN DATA POINTS (TO BE; 
.PRINT, 'COMBINED) IS .7 SCREEN UNITS, THEREFORE, ALL POINTS
.PRINT, 'WITHIN .7 SCREEN UNITS OF EACH OTHER WILL BE COMBINED.

(2) ENTER NEW CHECKING DISTANCE IF DESIRED (DEF=7).

110 READ [.AS] DEF =_ default
CALT DEFAUL=DEF,.MSET.7) 
.PRINT, '..............WAIT'
110 ID=12
YMIN=.99,E30
YMAX=.99,E30
YMIN=XMIN
YMAX=XMAX
137 READ 10,2000 END=130, Y(I),Y(I)
.FIX(Y),F(X),YMAX=Y(I)
FIX(Y),F(X),YMIN=Y(I)
f=4
GO TO 137

139 CONTINUE
ND01=1
SSX= invariably)
SSY=(YMAX-.YMIN)/200.
FSSY=6T, SSS) SM == SSY
LDUP=0
LDUP=0
23 READ 10,2000 END=44, X(J),Y(J),Z(J),CD(J) 
DU 99 111
I=1
.45 DIST= |X(I)-X(J)|**2+(Y(I)-Y(J)**2)/SM
DIST= DIST DSET ,IDUP =IDUP+1
IF(DIST.EQ.0) LDUP=LDUP+1
IF(DIST.EQ.0) EHCD(J),AND.EDIT=ED.0.OR.DIST.E.DSET) THEN
Z(J) X(J)
NE CD(J) Z(J)Z(J)Z(J)
IF(DIST,EQ.0) EHCD(J),AND.EDIT=EB.0.OR.DIST.E.DSET) THEN
CD(J) CD(J)
GO TO 23
29 CONTINUE
J=I
GO TO 23
44 GY=.15 Y(I)
445 IF(DUP,GY) THEN 
PRINT 455, DUP+1+.MSET
4455 FORMAT (I2, 15H POINTS WITHIN, I4, 27H SCREEN UNITS OF EACH OTHER) 
PRINT, 'POSITIONS OF FIRST INVOLVED POINT(S) FOUND IN FILE. 
PRINT, '[A]' TRASH 
PRINT, 'ALLDup, GYO) THEN 
PRINT.
PRINT '445, DUP L' DUP I DUPLICATE LABELS.
PRINT '446, INFO FOR LAST DUPLICATE LABEL IN FILE WAS USED.'
PRINT '(A1),' '
READ '(A1), TRASH'
PRINT '.
END
XMAX=.99.E30
XMIN=XMAX
YMIN=XMIN
DO 33 I=1,J
IF (Y(J:I+1).GT.YMAX) XMAX=X(J)
IF (Y(J:I).LT.YMIN) XMIN=X(J)
IF (X(J:.I+1).LT.XMIN) XMAX=X(J)
IF (X(J:.J).GT.XMAX) XMIN=X(J)
33 CONTINUE
SUBX=0
SUBY=0
IF (ABS(YMIN).GT.1.0D0.A(YMAX-YMIN)) OR
ABS(YMIN).GT.1.0D0.(YMAX-YMIN) THEN
LOG(LOG10(YMIN))=LOG=LOG-1
XMIN=XMIN/10.**LOG
YMIN=YMIN/10.**LOG
END IF
Y(J).LT.YMIN
Y(J)=YMIN
Y(J).GT.YMIN
Y(J)=YMIN
IF ABS(J-XMIN).GT.1000.
ABS(J-JMIN).GT.1000.
JMIN(JMIN)+J
JMIN=JMIN+J
CONTINUE
END
920 PRINT 8889,MIN(ZMIN,ZMAX),MAX(ZMIN,ZMAX),J
8889 FORMAT(6,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H)
PRINT 'DATA POINTS= J, J '
READ '(F20.0)',ZCHAR
ZMUL=ZCHAR,'(F20.0)' ZMUL
DO 923 I=1,J
923 ZMIN=FAC-ZMUL
ZMAX=FAC-ZMUL
PRINT
END
920 PRINT 8889,MIN(ZMIN,ZMAX),MAX(ZMIN,ZMAX),J
8889 FORMAT(6,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H)
PRINT 'DATA POINTS= J, J '
READ '(F20.0)',ZCHAR
ZMUL=ZCHAR,'(F20.0)' ZMUL
DO 821,822
820 PRINT 8889,MIN(XMIN,XMAX),MAX(XMIN,XMAX),MAX(XMIN,XMAX)
8889 FORMAT(6,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H,5.6H)
PRINT 'DATA POINTS= J, J '
READ '(F20.0)',ZCHAR
ZMUL=ZCHAR,'(F20.0)' ZMUL
DO 920,921
921 OPEN(43,FILE='M3445R')
WRITE(43, '(A20.)') ZCHAR
RENO(43)
READ(43,ERR=821) XMULT,YMUL
GO TO 820
821 PRINT 'ENTER TWO NUMBERS SEPARATED BY A COMMA.'
CLOSE(43)
GO TO 820
822 CONTINUE
CLOSE(43,STATUS='DELETE')
FAC=FAC*XMULT
FAC=FAC*YMUL
DO 824 I=1,J
C7
M

IN=M'YMI

-LT

YXT1IN =YMIN

riI.L

T

XM1AX

= MAVXULT

YMAX:YMAX'YMU~LT

GO TO

820

ENDIF

XMIN=99. E30

0054 5ý 3

I:1.J

IF (X II LT.XMINJ X?¶IN=X( I

IEYLIJLT.YIINJ YMIN=Y(I

xI

XIJGT.XMAX XMAX:X~

IF VIl.GT.YMAX

jYI¶AX=Y

L1

553 CONTINUE

PRINT

WRITE 6,'[2NL]' HONOR, SLASH

WRITE 6,'[2AI]' TC60R, SLASH

IF (TYF.EQ.'TR1\') PO=O.

WRITE 6,'[AI]' TYPE

WRITE 6,'[A1]' FACT, FACT

WRITE 6,'[A1]' SUBX, SUBY

CLOSE 80

WRITE 6,'(I4)', (F1(I), I=1,14)

WRITE 6,'I4', SUBX, SUBY

IF (ASIS.EQ.'Y' OR. ASIS.EQ.'Y') THEN

DO 87 I=1,3

LLBL=6(ED1)

87 WRITE 11,'(F1)', X(I), Y(I), Z(I), I+1000, ', ', , LLBL

GO TO 158

ENDIF

TASK='N' OR 'Y' OR. ISG.EQ.'Y') THEN

IF (HONOR NE.'Y') CC=998

GO TO 620

IF (IBOR.EQ.2) IG=IG+2.

PRINT ' ?, I6) WANT NORMAL(N) OR FINE(F) INVERSE POWER'

PRINT ' (AI)', 'IASK

PRINT ' (AI)', 'WAIT', ',

IF (ASK.EQ.'F') IGRID=166

ENDIF

GRID=IGRID

DMAX=(XMAX-XMIN)**2+(YMAX-YMIN)**2)**0. 5

MLAB=1

DO 30 I2=1, IGRID+1, IBOR

IF (L.0. I) X=O

Y=1YMIN+(2- IBOR/2)**((YMAX-YMIN))/XGRD

DO 30 I2=1, (XGRD+1, IBOR-1)

X-B-1)*((XMAX-XMIN)/(XGRD))+XMIN+(11-1- IBOR/2)**((XMAX-XMIN))

/XGRD

30 CONTINUE

DO 35 I=1,3

IF (ISG.NE.'Y' AND. ISG.NE.'Y') THEN

IF (L.0.0075*DMAX) GO TO 30

ENDIF

IF (HONOR.EQ.'Y' OR. HONOR.EQ.'Y') THEN

IF (L.0.0075*DMAX) GO TO 30

ENDIF

SB=SB+1

35 CONTINUE

IF (ISG.NE.'Y' AND. ISG.NE.'Y') THEN

WRITE 11,'X8.YB.X57/SB,MLAB.'

GO TO 28

ENDF

IF (L.0. AND. K.EQ.0) THEN

WRITE 11,'X8.YB.X57/SB,MLAB.'

GO TO 28

C8
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
END
SUBROUTINE巴菲L(C,D,E)
CHARACTER*5,C, D, E
IF (C'=') RETURN
READ(*,') (D(i),i=1,5)
DO 100 I=1,5
IF (D(i)'.EQ.' ') GO TO 100
IF (D(i)'.EQ.' ') GO TO 100
IF (D(i)'.EQ.' ') GO TO 100
IF (D(i)'.EQ.' ') GO TO 100
IF (D(i)'STD',C'=') RETURN
READ(*,') C
RETURN
END
Waterways Experiment Station Cataloging-In-Publication Data

Palmerton, John B.

McCON : a general contouring program for personal computers / by
John B. Palmerton ; prepared for Department of the Army, US Army
Corps of Engineers.

107 p. : ill. ; 28 cm. -- (Miscellaneous paper; GL-92-31)

Includes bibliographical references.


I. United States. Army. Corps of Engineers. II. U.S. Army Engineer
Waterways Experiment Station. III. Computer Applications in Geotechni-
cal Engineering (CAGE) IV. Title. V. Title: McCON -- a general contour-
ing program for personal computers. VI. McCON--a general contouring
program for personal computers. VII. Series: Miscellaneous paper (U.S.
Army Engineer Waterways Experiment Station) ; GL-92-31.

TA7 W34m no.GL-92-31