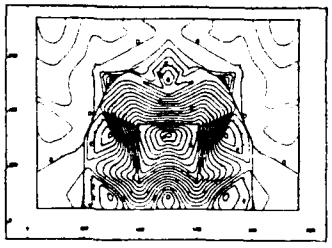
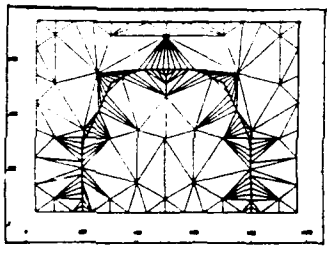
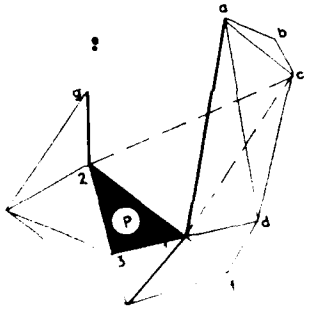












(2)

**COMPUTER APPLICATIONS IN GEOTECHNICAL ENGINEERING (CAGE)**

MISCELLANEOUS PAPER GL-92-31

# McCON—A GENERAL CONTOURING PROGRAM FOR PERSONAL COMPUTERS

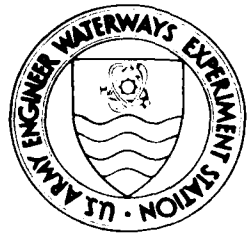
by

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



Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | September 1992 | Final report |

**4. TITLE AND SUBTITLE**

McCON--A General Contouring Program for Personal Computers

**5. FUNDING NUMBERS**

Work Unit No. 31700

**6. AUTHOR(S)**

John B. Palmerton

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

USAEWES, Geotechnical Laboratory,
3909 Halls Ferry Road, Vicksburg, MS 39180-6199

**8. PERFORMING ORGANIZATION REPORT NUMBER**

Miscellaneous Paper GL-92-31

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

US Army Corps of Engineers
Washington, DC 20314-1000

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

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

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

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.

| 14. SUBJECT TERMS | | 15. NUMBER OF PAGES |
|---|---|---|
| Contouring<br>Personal computers | Topography | 107 |
| | | 16. PRICE CODE |

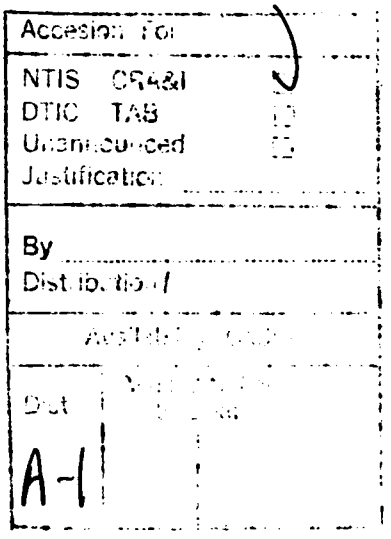
| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | | |

## 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

# 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 co-processor 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

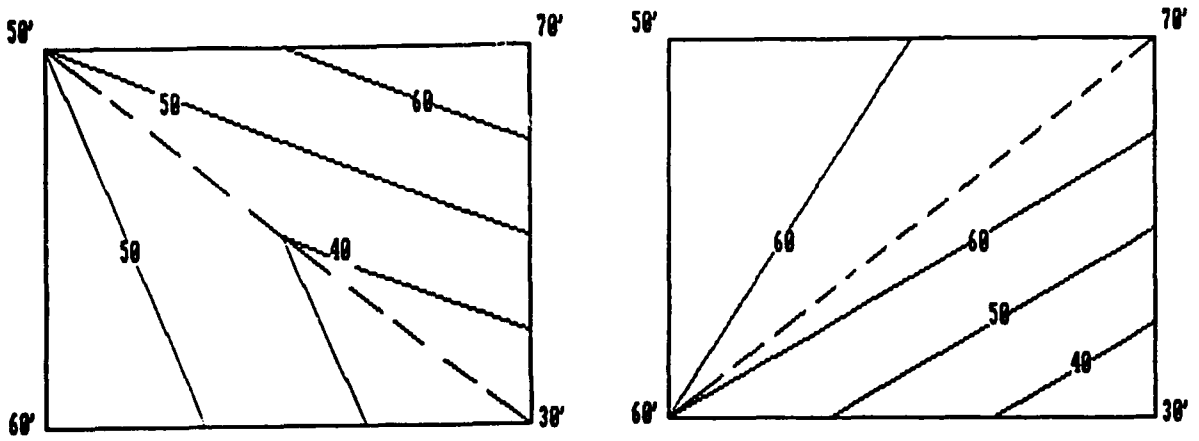
complexity of the contours nor to the spatial distribution of the nodes. Considerably more time for drawing is required if the output is sent to a plotter.

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, (1)

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

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

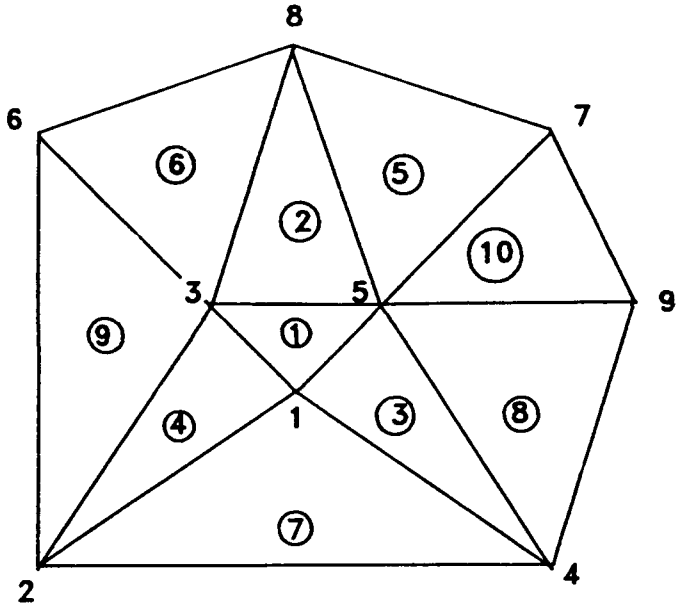



Figure 2. Spawning triangles

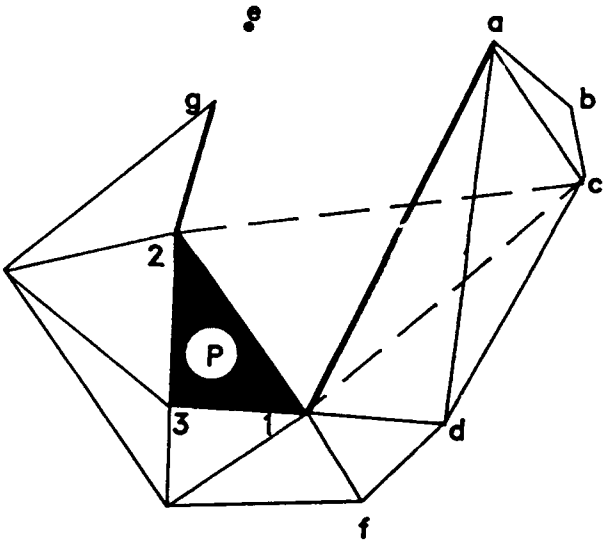



Figure 3. Criterion C illustration

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

clockwise. Nodes g and e are not rejected since Area (1,a,g) and Area (1,a,e) are ordered counterclockwise. Area (1,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:

$$Z_i = \sum_{j}^{j} [Z_j/(d_{i\,j})^Q] \;/\; \sum_{j}^{j} [1/(d_{i\,j})^Q] \qquad \textit{for all } d_{i\,j} \neq 0 \tag{2}$$

$$Z_i = Z_j \qquad \text{for any } d_{i\,j} = 0$$

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

$Z_J$ = the elevations at the given original points, j,

$d_{i\,j}$ = 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.

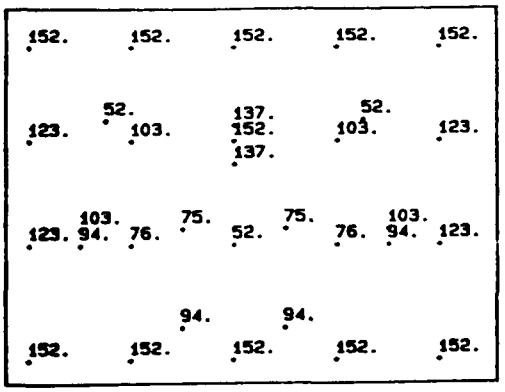
17. The results from various data input manipulations are shown in Figure 4. The location and contourable value (elevation) of 32 nodes is shown in Figure 4a. The "as is" contour plot is shown in Figure 4b. The results of "gridding" the data and not including the "original" data points are shown in Figure 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

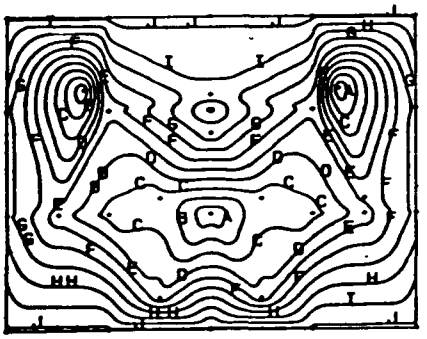
18. Since the spawning process outlined above is a single-pass process, the contour lines can be drawn as the triangles are spawned. Figure 5 shows a triangle being spawned. The numbers at the nodes represent contourable value (or elevation). Presume that it is required to draw the 10 and 20 contours. 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




a. Location and elevation of nodes

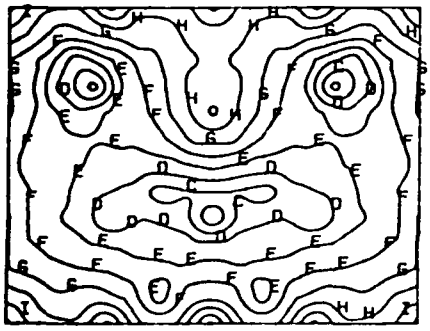


MCDEMO.DAT

CONTOURS
X 1.

| | |
|---|---|
| A | 60. |
| B | 70. |
| C | 80. |
| D | 90. |
| E | 100 |
| F | 110 |
| G | 120 |
| H | 130 |
| I | 140 |
| J | 150 |

b. "As is" contours

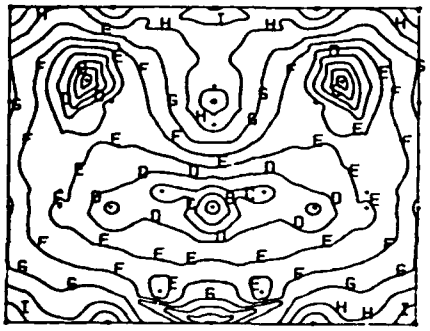


MCDEMO.DAT

CONTOURS
X 1.

| | |
|---|---|
| A | 60. |
| B | 70. |
| C | 80. |
| D | 90. |
| E | 100 |
| F | 110 |
| G | 120 |
| H | 130 |
| I | 140 |
| J | 150 |

c. "Gridded," no original points

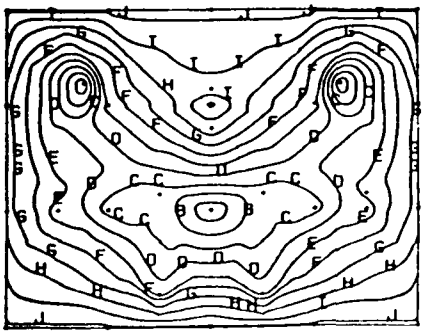


MCDEMO.DAT

CONTOURS
X 1.

| | |
|---|---|
| A | 60. |
| B | 70. |
| C | 80. |
| D | 90. |
| E | 100 |
| F | 110 |
| G | 120 |
| H | 130 |
| I | 140 |
| J | 150 |

d. "Gridded," honor original points

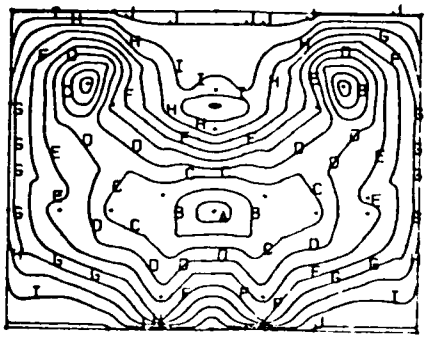


MCDEMO.DAT

CONTOURS
X 1.

| | |
|---|---|
| A | 60. |
| B | 70. |
| C | 80. |
| D | 90. |
| E | 100 |
| F | 110 |
| G | 120 |
| H | 130 |
| I | 140 |
| J | 150 |

e. Triple triangles, linear averaging



MCDEMO.DAT

CONTOURS
X 1.

| | |
|---|---|
| A | 60. |
| B | 70. |
| C | 80. |
| D | 90. |
| E | 100 |
| F | 110 |
| G | 120 |
| H | 130 |
| I | 140 |
| J | 150 |

f. Triple triangles, inverse power averaging

Figure 4. Contours resulting from various input data enhancements

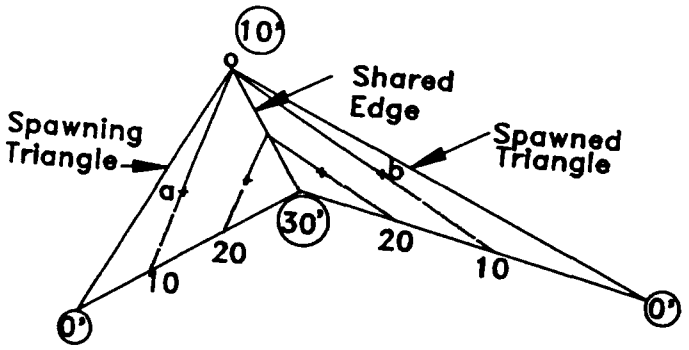



Figure 5. Drawing sharp-cornered contours

triangles. 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

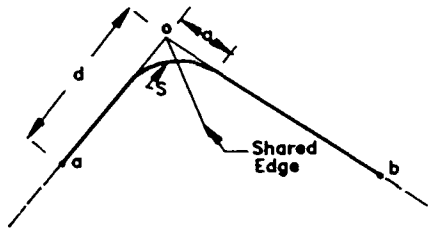



Figure 6. Circularly segmented contours.

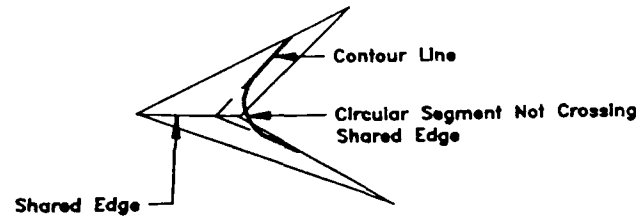



Figure 7. Nonpermitted circular segment.

q, not cross the shared line. It is necessary to ensure that all contour lines are drawn entirely within the two triangles under consideration; otherwise, problems with crossing contour lines could arise.

## Noncontoured Region

20. At times, it is desirable NOT to draw contours within some finite area on the contour plot. For example, the region beneath a building is generally not contoured in conjunction with the surrounding area. McCON allows the user to exclude from contouring a single internal convex area. As previously stated, the program generates triangles that always completely encompass the range of data points without "holes." The triangle generation ceases only when the convex set of triangles connecting the exterior nodes is accomplished. The ability to create a noncontoured region requires a two-step spawning process. The user must specify which nodes form the outline of the noncontoured region. (Those nodes should specify an internal convex region. If the specified nodes do not form a convex set, the program will create a convex region defined by those points that may not conform to the user's wishes.) The first stage of the triangle generation process uses only those specified nodes to create a mesh. The result of this stage will be a convex mesh that has an external boundary coincident with the convex set formed by the specified nodes. A list of 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.

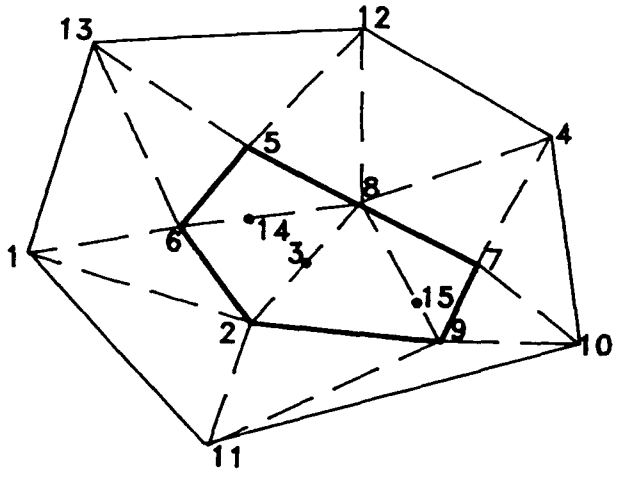



Figure 8. Describing a non-contoured region

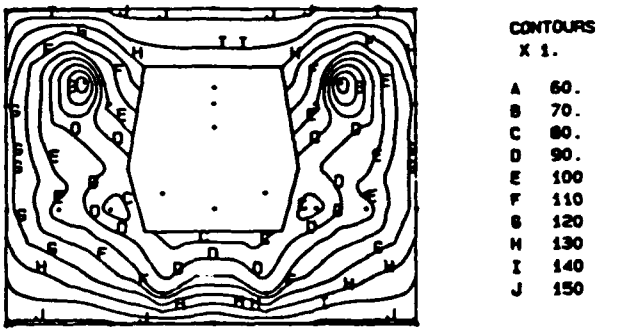



Figure 9. Example of a non-contoured region

## 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.

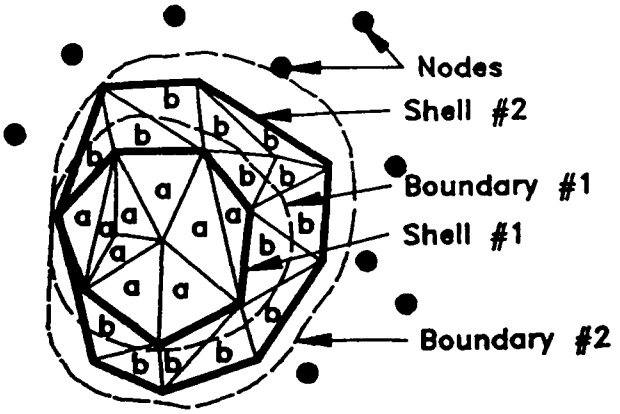



Figure 10. Spawning with successive shells

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.

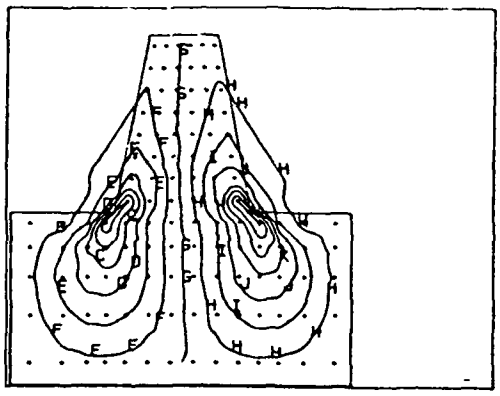
28. The successive shell procedure does have the disadvantage that "peculiar" triangles may be generated at the shell boundaries. This may result in some unwarranted "waviness" in contour line drawing. Therefore, in the event that the contour plots are being directed to a plotter, the number

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.

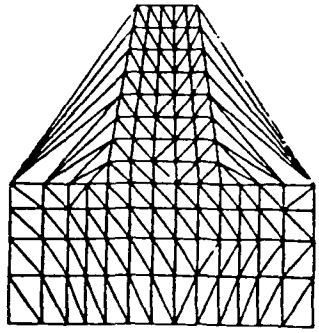
## Contouring Nonconvex Areas By Composite Fragments

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.

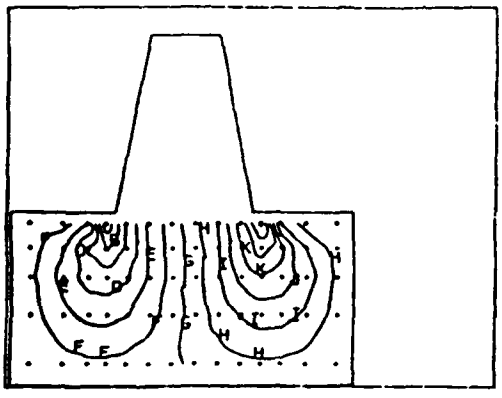
30. Figure 11a shows the contours of the shearing stresses within an embankment and its foundation. (The small crosses or tics on the drawing represent the location of the data points. The outline of the embankment and foundation was drawn using the "template" option). Contour lines are drawn outside 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



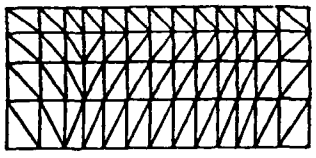

a. Noncomposite contours



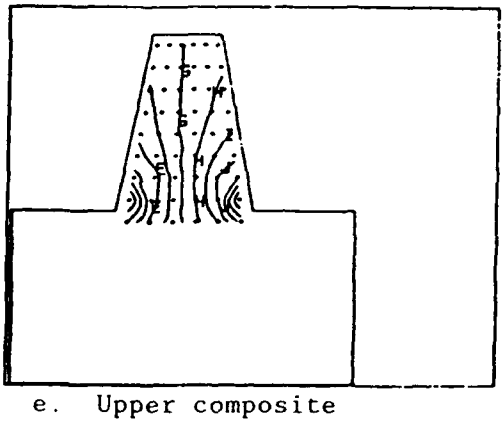
b. Generated triangle mesh



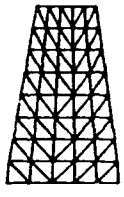

c. Lower composite



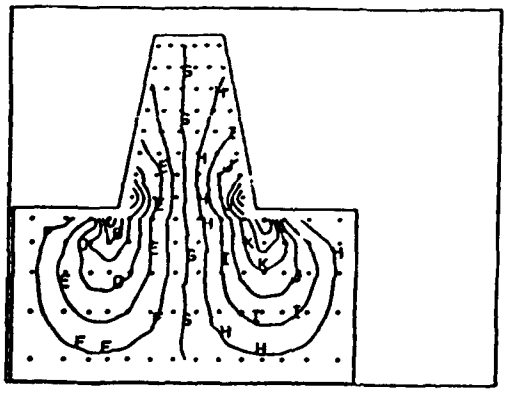
d. Triangle mesh for foundation portion



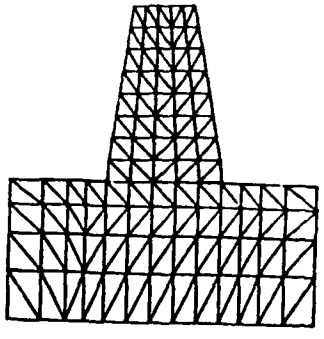

e. Upper composite



f. Triangle mesh for embankment portion




g. Combined composites



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 $Z_j$ 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 $Z_j$ restricted to points outside the non-contoured area.

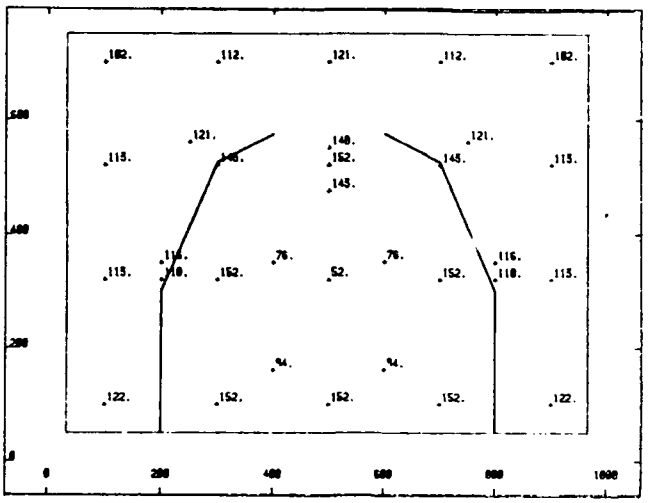
33. Figure 12 contains a series of drawings which demonstrate contouring in the vicinity of a "discontinuity" boundary. The data represent water table elevations on either side of a vertical impermeable boundary. Since water cannot flow across an impermeable boundary, interpolation of elevations across 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

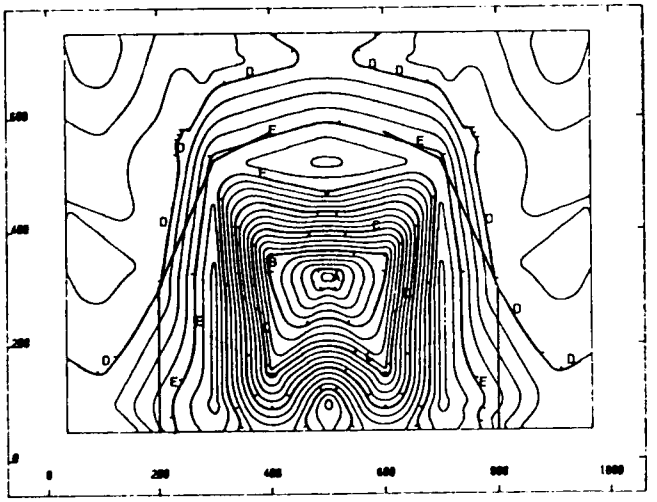
34. After a triangular mesh has been formed, it is a relatively simple process to create an "elevation" profile along any section of the mesh. The x,y location coordinates and the z value (elevation) for the vertices of each triangle is then known. After the user specifies the beginning and ending locations 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:

$$z = ax + by + c \tag{3}$$

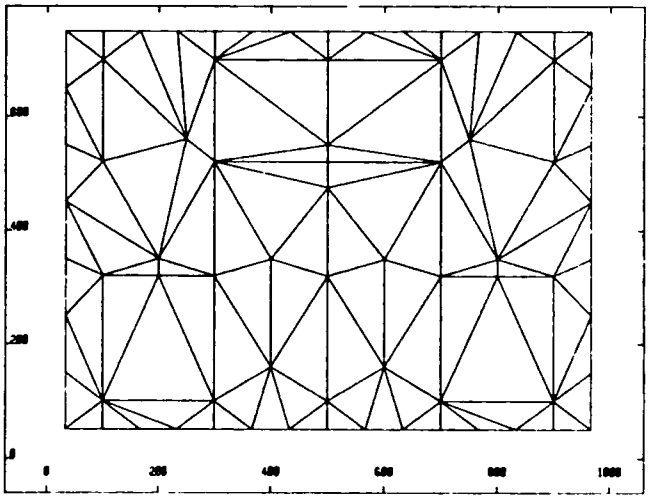
where the constants a,b, and c are found by substitution of the known values of x,y, and z at the three vertices of the triangle in which the profile point lies. The profile drawings may either be superimposed directly on the contour plots as shown in Figure 13a, or the (same) profile may be produced 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.



a. Location of "barriers" and elevations of nodes

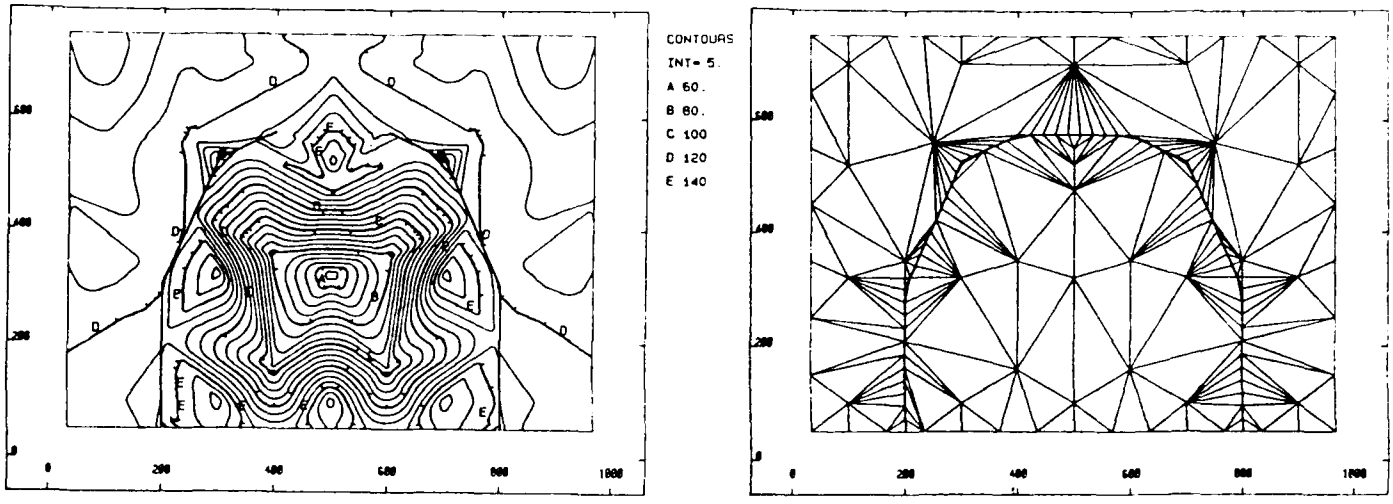



b. Contours without "barriers"



c. Mesh for no "barrier"

Figure 12. Contours across a "fault" boundary

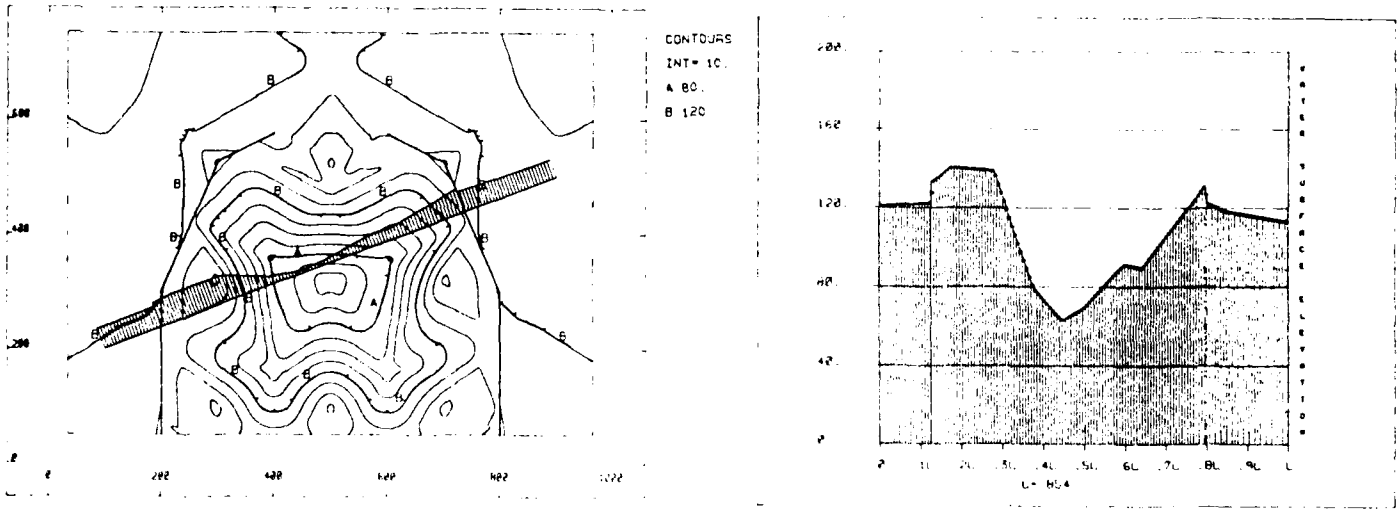



d. Contours with "barriers"

e. Mesh (composite) with "barriers"

Figure 12. Continued

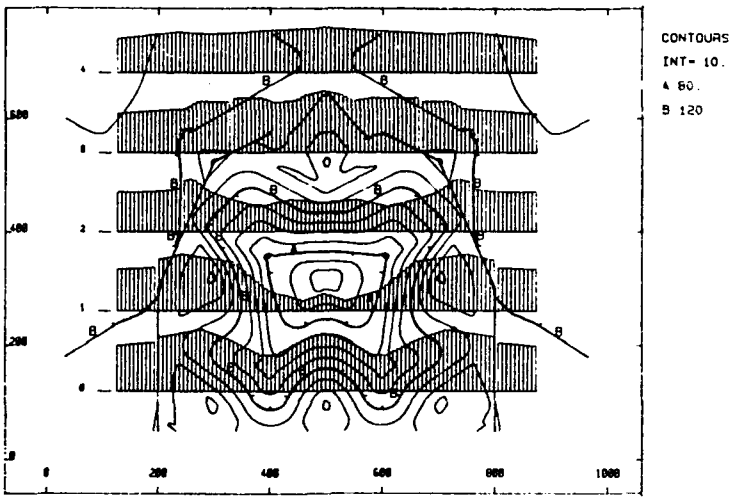
35. The user may also create a "mesh" file following completion of a triangle mesh. The "mesh" file may be used to create additional contour plots and profiles without having to again go through the process of mesh generation. The user's guide (Appendix A) gives complete instructions for the use and application of McCON.



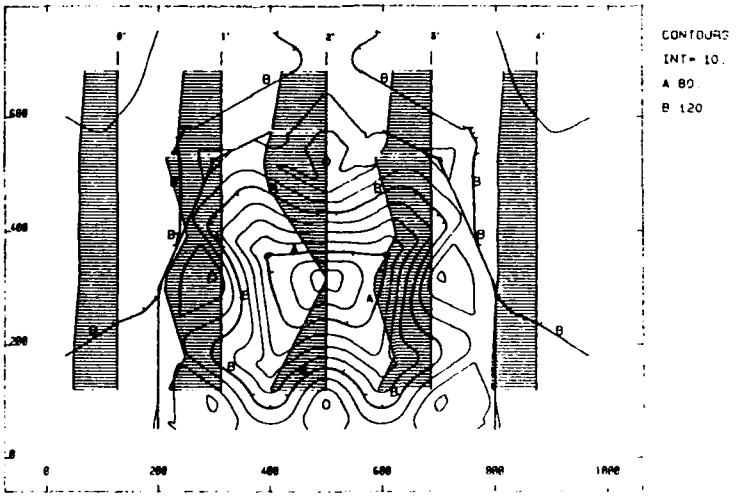

a. Profile over contour plot

b. Single page profile

Figure 13. Skewed profile plots




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 (MICRO1.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.*

**o 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 examine 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)
XXXXXX 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)
 etc.  ..   ...   ...    ...  ..   (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 contouring 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 rectangular 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:

$$Z_i = \sum_{1}^{j} [Z_j/(d_{ij})^Q] \; / \; \sum_{1}^{j} [1/(d_{ij})^Q] \quad \text{for all } d_{ij} \neq 0$$

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

where $Z_i$ 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).

(1.07) THRU (1.14) RELATE TO THE IDENTIFICATION OF THE DATA ELEMENTS.

The following discussion refers to the sample"columnar" data file below.

```
1     2    3    4       5 6     7     8     9
XXXXX XXXX XXXX XXXXXXX X XXXXX XXXXX XXXXX XXXXX
24-A1 6226 3756 5150.23 D 14.54 13.50 23.22
24-A2 6444 3890 5160.34 D 15.43 23.45 16.45 15.34
24-A3 6489 3967 5170.33 A 17.34 24.44 13.23
24-A1 6226 3756 5150.23 D             23.22 12.80
25-A1 6422 4201 5180.33 A 14.54 12.32 15.44 13.34
... 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. Therefor, the information associated with the second entry will be used for any operations involving Columns 8 or 9 and the first entry will 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.

**o 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 applicable 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 boundaries 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. Specifying 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 ·,6,‹, AND › KEYS. ---THE ·,6,‹. 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.

***(2.20) SKIP LEGEND ? (Y/N)***: The contours lines are 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., $x_1,y_1$ - $x_2,y_2$) 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 is 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 .0P, .1P, etc., (for profiles drawn parallel to the base of the grid and .0N, .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 EXAGERRATION 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

```
C   ..................................................................
C   .                                                                .
C   .                      *** ((( McCON ))) ***                     .
C   .                                                                .
C   ..................................................................
C   .                                                                .
C   .                  .....A CONTOURING PROGRAM.....                .
C   .                            WRITTEN BY                          .
C   .                         JOHN B. PALMERTON                      .
C   .                    GEOTECHNICAL LABORATORY                     .
C   .                  WATERWAYS EXPERIMENT STATION                  .
C   .                  U.S. ARMY CORPS OF ENGINEERS                  .
C   .                     VICKSBURG, MISSISSIPPI                     .
C   .                          AUG  1991                             .
C   ..................................................................
C
C..THIS PROGRAM REQUIRES A SET OF (N) X-Y-ELEV DATA POINTS
C..A GRID IS CREATED WHICH CONNECT THE DATA POINTS INTO TRIANGLES.
C. THESE TRIANGLES WILL COMPLETELY OCCUPY THE CONVEX REGION DEFINED
C. BY THE EXTERIOR POINTS.  THE GRID IS CREATED BY:
C. 1. CREATING A SEED TRIANGLE WITH EDGES 1-2-3 LABELED COUNTERCLKWSE.
C. 2. IN ORDER OF EDGES, FORMING NEW TRIANGLES BY SELECTING THE POINT
C.    WHICH LIES OUTSIDE OF THE EDGE OF THE GENERATING TRIANGLE
C.    AND FORMS THE LARGEST ANGLE WITH THAT EDGE.
C. 3. REPEATING THIS PROCESS FOR ALL EDGES OF ALL TRIANGLES UNTIL
C.    EDGES ARE SHARED BY TWO TRIANGLES OR A CONVEX BOUNDARY IS
C.    FORMED.
C..CONTOURING IS DONE BY ASSUMING A LINEAR INTERPOLATION OF ELEV
C. BETWEEN TRIANGLE VERTICES.  AS EACH NEW TRIANGLE IS FORMED THE
C. CONTOUR LINES WHICH CROSS THE SHARED EDGE ARE ARE DRAWN AS EITHER
C. STRAIGHT LINES OR CIRCULAR SEGMENTS.  (IF THE USER WISHES TO
C. INSURE THAT ALL CONTOURS ARE STRAIGHT LINES BETWEEN INTERPOLATION
C. POINTS, SET THE VARIABLE QDEL=1. THIS IS USEFUL FOR REPRESENTING
C. SITUATIONS INVOLVING GRADED LANDFORMS.)
C.
C.......DEFINITION OF SOME PARAMETERS
C.
C. X(I),Y(I),Z(I) -----COORDINATES AND ELEVATION VALUES OF DATA POINTS
C. SIDE(M,I) ---------POINTS ON ENDS OF EDGES(I) OF TRIANGLE(M)
C. VERTEX(M,I) --------POINTS(I) AT VERTICES OF TRIANGLE(M) NUMBERED
C.                     COUNTERCLOCKWISE
C. NODE(I) ------------LINKED LIST OF THE NUMBERS OF THE ENDPOINTS
C.                     WHICH HAVE THE OTHER END AT POINT I.
C.
      COMMON/CONTOUR/NC,CL(500),IANNS,FACT,PX,PY,PZT,CLS
      COMMON/CRAP/QDEL,MM
      COMMON/JPOW/IPOW
      COMMON/BOUND/NCON,LCON(999),MED,MCON
      COMMON/BOX/IBN(72,54)
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      DIMENSION X(999),Y(999),Z(999)
      CHARACTER IANS*1,DEV*4,CLS*4,JNC*1
      CHARACTER FNME*12,BNME*12,FMME*14
      CHARACTER COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER PERC*10
      CHARACTER CONVAL*10
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER COL(8)*4
      DATA COL/'YELL','BLAC','RED ','CYAN','GREE','WHIT','MAGE','BLUE'/
      DATA IFIRST/0/
      QDEL=25.
      FNME=
      BNME='MAP.DAT'
      NCOLER=0
 5933 PRINT,'(A\)',' ENTER A ''Y''  ) '
      READ '(A1)',IANS
      IF(IANS.EQ.'Y') GO TO 5938
      IF(IANS.EQ.'y') PRINT*,'SET CAPSLOCK !!!'
      GO TO 5933
 5938 CONTINUE
    5 CONTINUE
      DO 88 I=1,72
      DO 88 J=1,54
   88 IBN(I,J)=0
      XWMIN=1.
      XWMAX=1040.
      YWMIN=0.
      YWMAX=780.
```

```
      INQ=1
      NCON=0
      PRINT*,'  .....PLEASE NOTE !.....'
      PRINT*,' FOR QUESTIONS THAT ARE TO BE ANSWERED ''Y'' OR ''N''.'
      PRINT*,' ANY RESPONSE,(INCLUDING A CARRIAGE RETURN) OTHER THAN'
      PRINT*,'  ''Y'' MEANS ''NO''.A BEEP MEANS HIT RETURN TO CONTINUE.'
      PRINT*,' (2.01)'
      PRINT*,' ENTER DATA FILE NAME. FIRST DEFAULTS TO ''MAP.DAT'''
      PRINT*,'                       THEN TO LAST FILE NAMED.'
      PRINT*,' OR ENTER ''TRIPLE'', TO USE TRIPLE TRIANGLE MESH,'
      PRINT*,' OR ENTER ''*D(raw)'' TO DRAW A PREVOIUSLY SAVED '
     +'PROFILE'
      PRINT*,' OR ENTER ''*M(ake)'' TO CREATE A PROFILE FROM A'
     +' PREVIOUSLY'
      PRINT*,'                      GENERATED AND SAVED MESH'
      PRINT*,' OR ENTER ''*G(rid)'' TO DRAW GRID BOX PROFILES (A MESH'
     +' MUST'
      PRINT*,'                      HAVE BEEN PREVIOUSLY SAVED)'
      PRINT*,' OR ENTER ''*R(econ)'' TO REDRAW THE CONTOURS FROM A'
      PRINT*,'                       PREVIOUSLY SAVED MESH.'
      PRINT,'(A)',' OR ENTER ''S'' TO STOP  )'
      READ '(A12)',FNME
      IF(FNME.EQ.'S') THEN
      REWIND 09
      CLOSE(09)
      OPEN(09,FILE='TAP.DAT',STATUS='OLD',ERR=1881)
      CLOSE(09,STATUS='DELETE')
1881  OPEN(09,FILE='TEM',STATUS='OLD',ERR=1882)
      CLOSE(09,STATUS='DELETE')
1882  OPEN(09,FILE='SKUN',STATUS='OLD',ERR=1883)
      CLOSE(09,STATUS='DELETE')
1883  OPEN(09,FILE='SKUNK',STATUS='OLD',ERR=1884)
      CLOSE(09,STATUS='DELETE')
1884  OPEN(09,FILE='SAVCON',STATUS='OLD',ERR=1885)
      CLOSE(09,STATUS='DELETE')
1885  OPEN(09,FILE='OUTEDGE',STATUS='OLD',ERR=1886)
      CLOSE(09,STATUS='DELETE')
1886  OPEN(09,FILE='EDGECON',STATUS='OLD',ERR=1887)
      CLOSE(09,STATUS='DELETE')
1887  CONTINUE
      STOP
      ENDIF
      IF(NCOLER.EQ.0) THEN
      NCOLER=1
      PRINT*,,'COLOR CHOICES ARE:  RED,YELLOW,BLUE,MAGENTA,CYAN'
      PRINT*,'                     BLACK,GREEN, AND WHITE'
      PRINT '(A)',
     +' (2.02) ENTER COLOR FOR LINE WORK (DEF=YELLOW) ) '
      READ '(A4)',COLOR
      IF(COLOR.EQ.'RED') COLOR='RED '
      IF(COLOR.EQ.'GREN') COLOR='GREE'
      IF(COLOR.NE.'RED '.AND.COLOR.NE.'BLUE'.AND.COLOR.NE.'WHIT'.AND.
     +COLOR.NE.'MAGE'.AND.COLOR.NE.'YELL'.AND.COLOR.NE.'CYAN'.AND.
     +COLOR.NE.'GREE'.AND.COLOR.NE.'BLAC') COLOR='YELL'
      PRINT '(A)',' (2.02) ENTER BACKGROUND COLOR (DEF=BLACK) ) '
      READ '(A4)',CLRBCK
      IF(CLRBCK.EQ.'RED') CLRBCK='RED '
      IF(CLRBCK.EQ.'GREN') CLRBCK='GREE'
      IF(CLRBCK.NE.'RED '.AND.CLRBCK.NE.'BLUE'.AND.CLRBCK.NE.'WHIT'.
     +AND.CLRBCK.NE.'MAGE'.AND.CLRBCK.NE.'YELL'.AND.CLRBCK.NE.'CYAN'.
     +AND.CLRBCK.NE.'GREE'.AND.CLRBCK.NE.'BLAC') CLRBCK='BLAC'
      DO 8904 I=1,8
      IF(COLOR.EQ.COL(I)) JJ=I
      IF(CLRBCK.EQ.COL(I)) KK=I
8904  CONTINUE
      DO 7700 I=JJ,JJ+8
      L=I
      IF(L.GT.8) L=L-8
      COL1=COL(L)
      IF(COL1.NE.COLOR.AND.COL1.NE.CLRBCK) GO TO 7701
7700  CONTINUE
7701  DO 7702 I=KK,KK+8
      L=I
      IF(L.GT.8) L=L-8
      COL2=COL(L)
      IF(COL2.NE.COLOR.AND.COL2.NE.CLRBCK.AND.COL2.NE.COL1) GO TO 7703
7702  CONTINUE
7703  DO 7704 I=1,8
      COL3=COL(I)
      IF(COL3.NE.COLOR.AND.COL3.NE.CLRBCK.AND.COL3.NE.COL1.
     +   AND.COL3.NE.COL2) GO TO 7705
7704  CONTINUE
7705  COL4=COLOR
```

```
      ENDIF
      IF(FNME.EQ.' ') FNME=BNME
      IF(FNME.EQ.'*D') GO TO 7777
      IF(FNME.EQ.'*M') GO TO 7777
      IF(FNME.EQ.'*G') GO TO 7777
      IF(FNME.EQ.'*R') GO TO 7777
      BNME=FNME
      OPEN(9,FILE=FNME,STATUS='OLD',ERR=567)
      GO TO 568
  567 IND=0
  568 CONTINUE
      READ(9,'(A14)') FMME
      READ(9,*) S1,S2
      READ(9,*) S3,S4
      REWIND 9
      OPEN(89,FILE='MCINFO.DAT')
      WRITE(89,'(A14)') FMME
      DO 189 I=1,5
  189 WRITE(89,'(A4)') '1111'
      WRITE(89,*) S1,S2
      WRITE(89,*) S3,S4
      CLOSE(89)
      IF(IFIRST.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,Z VALUES--1 SET'
      PRINT*,'  PER LINE.  THE POINTS WILL BE SCALED TO FIT SCREEN.'
      PRINT*,' TEMPLATES MAY BE DRAWN ON THE CONTOUR MAP. THE TEMPLATE'
      PRINT*,' DATA MUST BE IN A SEPARATE FILE;-- A FREE FIELD FILE'
      PRINT*,' OF -X,Y,PEN- VALUES. A NEGATIVE "PEN" VALUE INDICATES'
      PRINT*,' THE END OF A TEMPLATE (I.E. PEN IS UP TO NEXT POINT).'
      PRINT*,' (USE SAME COORDIDNATE SCALE AS SCALE FOR MAP.DAT)'
      PRINT*,'
      END IF
      IFIRST=1
      IF(IND.EQ.0) THEN
      PRINT*,
      PRINT*,' ---DATA FILE ',FNME,' NOT FOUND---'
      GO TO 5
      END IF
      IF(IND.NE.0) THEN
      PRINT*,' ..DATA FILE IS ',FNME,' ..
      ENDIF
      IPOW=0
      IF(FNME.EQ.'TRIPLE') IPOW=1
  876 CONTINUE
      JNC='1'
      NC=0
      IF(JNC.EQ.' ') NC=1
      IF(JNC.EQ.'1') NC=1
      IF(JNC.EQ.'2') NC=2
      IF(NC.EQ.1.OR.NC.EQ.2) GO TO 877
      GO TO 876
  877 CONTINUE
      REWIND 09
      QDEL=25
      IF(IABS(NC).GT.1) QDEL=1.
      REWIND 09
 7777 CONTINUE
      PRINT*,'FOR THE OUTPUT.....'
      PRINT '(A\)',
     +'(2.04) ENTER 1 FOR SCREEN, 2 FOR PLOTTER (DEF=1) )'
      READ '(A1)',JNC
      PRINT*,'.....WAIT'
      MM=1
      IF(JNC.EQ.' ') MM=1
      IF(JNC.EQ.'1') MM=1
      IF(JNC.EQ.'2') MM=2
      IF(JNC.EQ.'3') MM=3
      IF(MM.EQ.1) DEV='IBMH'
      IF(MM.EQ.2) DEV='HP4 '
      IF(MM.EQ.3) DEV='EPS '
      REWIND 09
      PCY=1.0
      PCX=PCY
      IF(MM.EQ.2) THEN
      PRINT*,
 5008 PRINT*,'(2.05) ENTER SIZE REDUCTION FACTOR (DEF=100)'
      PRINT '(A\)',  (NUM. BETWN 1-100) FOR PLOTTER )'
      READ '(A10)',PERC
      IF(PERC.EQ.' ') PCY=100.
      IF(PERC.NE.' ') THEN
      OPEN(43,FILE='M43234')
      WRITE(43,'(A10)') PERC
```

```
      REWIND 43
      READ(43,*,ERR=5008) PCY
      CLOSE(43,STATUS='DELETE')
      ENDIF
      IF(PCY.EQ.0.) PCY=100.
      PCY=PCY/100.
      PCX=PCY
      PRINT '(A\)',
     + ' (2.06) PAPER SIZE ? (''A''=SMALL.''B''=LARGE) (DEF=A) ) '
      READ '(A1)',PSIZE
      IF(PSIZE.NE.'B') PSIZE='A'
      IF(PSIZE.EQ.'B') PCX=.825*PCY
      PRINT '(A\)', ' (2.07) WANT PLOT IN SINGLE COLOR ONLY ? (Y/N) ) '
      READ(*,222) IANS
      IF(IANS.NE.'Y') THEN
      DO 8004 I=1,8
 8004 LPEN(I)=I
      PRINT*,'PEN1 IS USED TO DRAW CONTOURS AND LEGEND.'
      PRINT*,'PEN2 IS  .   .   .   DATA TIC POINTS & ELEVATIONS.'
      PRINT*,'PEN3 IS  .   .   .   TEMPLATES.'
      PRINT*,'PEN4 IS  .   .   .   PROFILES.'
      PRINT*,'PEN4 IS  .   .   .   HORIZONTAL (PARALLEL) GRID PROFILES'
      PRINT*,'PEN2 IS              VERTICAL (NORMAL) GRID PROFILES'
      PRINT*
      PRINT*,'SO,MAKE SURE YOU KNOW WHAT IS IN THE PEN HOLDER!'
      PRINT*
      CALL UBELL
      CALL UPAUSE
      ENDIF
      IF(IANS.EQ.'Y') THEN
      DO 8002 I=1,8
 8002 LPEN(I)=1
      ENDIF
      ENDIF
      XUDMIN=10.*PCX
      XUDMAX=75.*PCX
      YUDMIN=10.*PCY
      YUDMAX=75.*PCY
      IF(FNME.EQ.'*D'.OR.FNME.EQ.'*M'
     + .OR.FNME.EQ.'*G'.OR.FNME.EQ.'*R') THEN
      IF(MM.EQ.1) DEV='IBMH'
      IF(MM.EQ.2) DEV='HP4'
      CALL PSTART(MM)
      IF(FNME.EQ.'*D') CALL OUTPRO(MM)
      IF(FNME.EQ.'*M') CALL INPRO(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,
     + YMAXY,SM,0)
      IF(FNME.EQ.'*G') CALL FENCE(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,
     +  YMAXY,SM,0,0)
      IF(FNME.EQ.'*R') CALL FENCE(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,
     +  YMAXY,SM,0,1)
      CALL PEND
      GO TO 5
      ENDIF
      IPCELL=60
  222 FORMAT (A1)
      IF(MM.EQ.2) IPCELL=999
      CALL DRAWCON(IPCELL)
      CALL PEND
      IF(MM.EQ.3) PRINT*,'ENTER EPRINT TO SEND TO PLOTTER'
      IF(MM.EQ.3) STOP
      CLOSE(09)
      GO TO 5
      STOP
      END
C
      SUBROUTINE CONTER(ZINT)
      COMMON/TRI/MBL,MB
      COMMON/STUFF/ZI(4)
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +      VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/CONTOUR/NC,CL(500),IANS,FACT,PX,PY,PZT,CLS
      COMMON/BOUND/NCON,LCON(999),MED,MCON
      COMMON/BOUND/NUMCON,KCON(100)
      COMMON/CELLS/NUM(1000),IRN(100),IRD(100),KRD,AVX,AVY
      COMMON/MINS/XMIN,XMAX,YMIN,YMAX
      COMMON/DRAWS/MBLNUM(100),NUMBR
      COMMON/NODEL/NODE(17000),NAV,NPMAX
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      DIMENSION LED(1000)
      DIMENSION IS(3),P(3)
      DIMENSION XPT(2),YPT(2)
```

```
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      INTEGER SIDE,VERTEX,P1,P2,P3,P
      OPEN(51,FILE='SAVCON')
      DO 12 I=1,2000
      DO 12 J=1,3
      VERTEX(I,J)=0
   12 SIDE(I,J)=0
      DO 820 I=1,NP
  820 NODE(I)=0
      NAV=NP+1
      MBL=1
      NPT=NP
      M=1
      MB=1
      MSTART=1
      IBL=1
      MBLNUM(1)=0
      DO 950 JJ=2,KRD
      NP=IRN(JJ)
      IL=1
      DO 952 I=1,NPI
      D=(X(I)-AVX)**2+(Y(I)-AVY)**2
      ID=SQRT(D)+1.0001
      IF(ID.GT.IRD(JJ-1).AND.ID.LE.IRD(JJ)) THEN
      NUM(IL)=I
      IL=IL+1
      END IF
  952 CONTINUE
   20 IF(JJ.EQ.2) CALL SEED
   22 CONTINUE
  150 DO 200 L=1,3
      IF(L.EQ.1) THEN
      IS(1)=1
      IS(2)=2
      IS(3)=3
      GO TO 6
      ENDIF
      IF(L.EQ.2) THEN
      IS(1)=2
      IS(2)=3
      IS(3)=1
      GO TO 6
      ENDIF
      IS(1)=3
      IS(2)=1
      IS(3)=2
    6 CONTINUE
      IF(SIDE(M,IS(1)).LT.0) GO TO 200
      MPAREN=M
      IPAREN=IS(1)
      IDASM=0
 9916 NCRIC=0
 8810 CONTINUE
      NW=0
      DS=2000000.
C  SET MIN ANGLE EQUAL TO 15 DEGREES
      ASM=30.E30
      CX1=X(VERTEX(M,IS(1)))
      CX2=X(VERTEX(M,IS(2)))
      CY1=Y(VERTEX(M,IS(1)))
      CY2=Y(VERTEX(M,IS(2)))
      SD1=(CX1-CX2)**2+(CY1-CY2)**2
      DO 260 IP=1,NP
      I=NUM(IP)
      IF(NCON.GT.0) THEN
      DO 707 LC=1,NCON
      IF(I.EQ.LCON(LC)) GO TO 708
  707 CONTINUE
      GO TO 250
      END IF
  708 CONTINUE
      LLIN=0
C.....CRITERIA A.....
      A=CX1*(CY2-Y(I))+CX2*(Y(I)-CY1)+X(I)*(CY1-CY2)
C.....
      IF(A.GE.-1.0.AND.A.LE.1.0) GO TO 250
      IF(A.LT.0.0001) GO TO 250
C.....CRITERIA B.....
      LLIN=1
      SD2=(CX1-X(I))**2+(CY1-Y(I))**2
      SD3=(CX2-X(I))**2+(CY2-Y(I))**2
      IF(IDASM.EQ.1) AD=SD2+SD3
```

```
C   USE STATEMENT ABOVE TO MINIMIZE PERIMETER OF TRIANGLE
C   USE STATEMENT BELOW TO MAXIMIZE VERTEX ANGLE OF TRIANGLE
      IF(IDASM.EQ.0) AD=(SD2+SD3-SD1)/SQRT(SD2*SD3)
      IF(AD.GE.ASM) GO TO 250
C.....CRITERIA C.....
      IF(NCRIC.EQ.0) GO TO 8815
 8816 CONTINUE
      IG=VERTEX(M,IS(2))
      KN=NODE(IG)/100000
      IF(KN.EQ.0) GO TO 42
      LL=NODE(IG)
      LP=LL/100000
      LCC=LL-LP*100000
      DO 40 K=1,KN
      LIND=NODE(LCC)/100000
      LL=NODE(LCC)
      LP=LL/100000
      LCC=LL-LP*100000
      X1=X(VERTEX(M,IS(1)))
      X2=X(VERTEX(M,IS(2)))
      X3=X(LIND)
      X4=X(I)
      Y1=Y(VERTEX(M,IS(1)))
      Y2=Y(VERTEX(M,IS(2)))
      Y3=Y(LIND)
      Y4=Y(I)
      CALL AREA(X1,X2,X3,Y1,Y2,Y3,A)
      IF(A.LE.0.001) GO TO 40
      CALL AREA(X2,X3,X4,Y2,Y3,Y4,A)
      IF(A.GE.-0.001) GO TO 40
      IF(NCRIC.EQ.1) THEN
      NCRIC=3
      GO TO 8810
      ENDIF
      GO TO 260
 40   CONTINUE
 42   CONTINUE
      IG=VERTEX(M,IS(1))
      KN=NODE(IG)/100000
      IF(KN.EQ.0) GO TO 47
      LL=NODE(IG)
      LP=LL/100000
      LCC=LL-LP*100000
      DO 45 K=1,KN
      LIND=NODE(LCC)/100000
      LL=NODE(LCC)
      LP=LL/100000
      LCC=LL-LP*100000
      X1=X(LIND)
      X2=X(VERTEX(M,IS(1)))
      X3=X(VERTEX(M,IS(2)))
      X4=X(I)
      Y1=Y(LIND)
      Y2=Y(VERTEX(M,IS(1)))
      Y3=Y(VERTEX(M,IS(2)))
      Y4=Y(I)
      CALL AREA(X1,X2,X3,Y1,Y2,Y3,A)
      IF(A.LE.0.001) GO TO 45
      CALL AREA(X1,X2,X4,Y1,Y2,Y4,A)
      IF(A.GE.-0.001) GO TO 45
      IF(NCRIC.EQ.1) THEN
      NCRIC=3
      GO TO 8810
      ENDIF
      GO TO 260
 45   CONTINUE
 47   CONTINUE
C.....
 8815 CONTINUE
      IF(AD.GE.ASM) GO TO 250
C IF THE ANGLE IS GREATER THAN 150 DEGREES FIND MINIMUM PERIMETER
      IF(IDASM.EQ.0.AND.AD.LT.-1.73) THEN
      IDASM=1
      GO TO 9916
      ENDIF
      ASM=AD
      NW=I
  250 CONTINUE
      IF(IP.EQ.NP.AND.NCRIC.EQ.0) THEN
      IF(NW.EQ.0) GO TO 260
      NCRIC=1
      I=NW
      GO TO 8816
```

```
      ENDIF
  200 CONTINUE
      IF(NW.EQ.0) GO TO 200
      MB=MB+1
C     MNOW IS THE PARENT TRIANGLE
      MNOW=M
      VERTEX(MB,1)=NW
      VERTEX(MB,2)=VERTEX(M,IS(1))
      VERTEX(MB,3)=VERTEX(M,IS(2))
      I1=VERTEX(MB,1)
      I2=VERTEX(MB,2)
      I3=VERTEX(MB,3)
      IF(2.EQ.3) THEN
      CALL UMOVE(X(VERTEX(MB,1)),Y(VERTEX(MB,1)))
      CALL UPEN(X(VERTEX(MB,2)),Y(VERTEX(MB,2)))
      CALL UPEN(X(VERTEX(MB,3)),Y(VERTEX(MB,3)))
      CALL UPEN(X(VERTEX(MB,1)),Y(VERTEX(MB,1)))
      CALL UMOVE((X(I1)+X(I2)+X(I3))/3.,(Y(I1)+Y(I2)+Y(I3))/3.)
      XXMB=MB
      CALL UPRNT1(XXMB,'INTE')
      END IF
      CALL AREA(X(I1),X(I2),X(I3),
     +          Y(I1),Y(I2),Y(I3),A)
      IF(A.LT.0.) GO TO 290
      IQ=VERTEX(MB,2)
      VERTEX(MB,2)=VERTEX(MB,3)
      VERTEX(MB,3)=IQ
  290 DO 295 II=1,3
  295 P(II)=VERTEX(MB,II)
      CALL ORDER(NG,P(1),P(2))
      SIDE(MB,1)=NG
      CALL ORDER(NG,P(2),P(3))
      SIDE(MB,2)=-NG
      CALL ORDER(NG,P(3),P(1))
      SIDE(MB,3)=NG
      SIDE(MPAREN,IPAREN)=-IABS(SIDE(MPAREN,IPAREN))
      IFON=0
      KN=NODE(NW)/100000
      IF(KN.EQ.0) GO TO 311
      LL=NODE(NW)
      LP=LL/100000
C  A LOCATION
      LCC=LL-LP*100000
      DO 310 K=1,KN
      LIND=NODE(LCC+1)
      ILP=LIND/100000
C THE TRIANGLE(S) WHICH SHARES NODE NW IS M2
      M2=LIND-ILP*100000
      LL=NODE(LCC)
      LP=LL/100000
      LCC=LL-LP*100000
      IF(M2.EQ.MB) GO TO 310
      DO 300 M1=1,3
      DO 320 M3=1,3
      IF(IABS(SIDE(MB,M1)).EQ.IABS(SIDE(M2,M3))) GO TO 350
      GO TO 320
  350 CONTINUE
      IF(SIDE(M2,M3).LT.0.AND.SIDE(MB,M1).LT.0.) GO TO 320
      SIDE(MB,M1)=-IABS(SIDE(MB,M1))
      SIDE(M2,M3)=-IABS(SIDE(M2,M3))
      CALL UALPHA
      IF(M2.NE.M) THEN
      CALL USET('INTE')
      IFON=IFON+1
      IF(IFON.EQ.3) PRINT*,'IFON=3'
      IF(IFON.EQ.3) STOP
      IF(IFON.EQ.2) THEN
      MZ3=M2
      K1=-SIDE(M2,M3)
      K2=K1/1000
      K3=K1-K2*1000
      GO TO 320
      END IF
      MZ=M2
      J1=-SIDE(M2,M3)
      J2=J1/1000
      J3=J1-J2*1000
      END IF
      GO TO 320
  320 CONTINUE
  300 CONTINUE
  310 CONTINUE
  311 CONTINUE
```

```
      CALL NODES(I1,I2,MB)
      CALL NODES(I1,I3,MB)
      CALL NODES(I2,I1,MB)
      CALL NODES(I3,I1,MB)
305   CONTINUE
      IF(NCON.GT.0) GO TO 200
      IF(MNOW.LT.MBL) GO TO 544
      ZT(1)=Z(I2)
      ZT(2)=Z(I3)
      ZT(3)=Z(I1)
      DO 541 IC=1,3
      IF(VERTEX(MNOW,IC).EQ.I2) GO TO 541
      IF(VERTEX(MNOW,IC).EQ.I3) GO TO 541
      ZT(4)=Z(VERTEX(MNOW,IC))
      NOLD=VERTEX(MNOW,IC)
      GO TO 542
541   CONTINUE
542   CONTINUE
      WRITE(51,*) MNOW,MB,I2,I3,NW,NOLD
     +,ZT(1),ZT(2),ZT(3),ZT(4)
      DO 543 IC=1,NC
      KJ=IC
      ZZ=CL(IC)
      IF(ZZ.EQ.ZT(1)) ZT(1)=ZT(1)+0.01*ZINT
      IF(ZZ.EQ.ZT(2)) ZT(2)=ZT(2)-0.01*ZINT
      IF(ZZ.EQ.ZT(3)) ZT(3)=ZT(3)+0.01*ZINT
      IF(ZZ.EQ.ZT(4)) ZT(4)=ZT(4)-0.01*ZINT
      ZN=AMIN1(ZT(1),ZT(2))
      ZM=AMAX1(ZT(1),ZT(2))
      IF(ZZ.GE.ZN.AND.ZZ.LE.ZM)
     + CALL DRAWIT(MNOW,MB,ZZ,I2,I3,NW,NOLD,KJ)
543   CONTINUE
544   CONTINUE
      IF(IFON.EQ.0) GO TO 200
      IF(IFON.EQ.2) THEN
      IFON=IFON-1
      MNOW=MZ3
      I2=K2
      I3=K3
      GO TO 548
      END IF
      IF(IFON.EQ.1) THEN
      IFON=IFON-1
      MNOW=MZ
      I2=J2
      I3=J3
      END IF
548   CONTINUE
      DO 546 IC=1,3
      IF(VERTEX(MB,IC).EQ.I2) GO TO 546
      IF(VERTEX(MB,IC).EQ.I3) GO TO 546
      I1=VERTEX(MB,IC)
      NW=I1
      GO TO 547
546   CONTINUE
547   GO TO 305
200   CONTINUE
      M=M+1
      IF(M.GT.MB) GO TO 400
      GO TO 150
400   CONTINUE
      IF(NCON.GT.0) THEN
      MCON=NCON
      NCON=0
      M=1
      MBL=MB+1
      LEDGES=0
      DO 808 I=1,MB
      DO 808 J=1,3
      IF(SIDE(I,J).GT.0) THEN
      LEDGES=LEDGES+1
      LED(LEDGES)=SIDE(I,J)
      END IF
808   CONTINUE
      IBL=IBL+1
      MBLNUM(IBL)=MB
      GO TO 22
      END IF
C  MAX&MIN ELEV
      IBL=IBL+1
      MBLNUM(IBL)=MB
      ZMAX=-2000000.
      ZMIN=2000000.
```

```
      DO 450 IP=1,NP
      I=NUM(IP)
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
  450 CONTINUE
C ... DRAW EDGES
      KCON(NUMCON+1)=KCON(1)
      CALL USET(COL1)
      OPEN(61,FILE='EDGECON')
      WRITE(61,*) NUMCON
      DO 1111 I=1,NUMCON
      CALL UMOVE(X(KCON(I)),Y(KCON(I)))
      CALL UPEN(X(KCON(I+1)),Y(KCON(I+1)))
      WRITE(61,*) X(KCON(I)),Y(KCON(I)),X(KCON(I+1)),Y(KCON(I+1))
 1111 CONTINUE
      CLOSE(61)
      CALL USET(COLOR)
      IF(JJ.EQ.KRD) THEN
      IF(MCON.GT.0) THEN
      IF(LEDGES.GT.0) THEN
      DO 1400 I=1,LEDGES
      DO 1400 J=MBL,MB
      DO 1400 K=1,3
      IF(LED(I).EQ.IABS(SIDE(J,K))) SIDE(J,K)=IABS(SIDE(J,K))
 1400 CONTINUE
      END IF
      END IF
      OPEN(65,FILE='OUTEDGE')
      IF(MED.NE.1) WRITE(65,*) 'ABC'
      DO 600 I=MBL,MB
      DO 605 J=1,3
      IF(SIDE(I,J).GT.0) GO TO 660
      GO TO 605
  660 K=SIDE(I,J)
      I1=K/1000
      I2=K-I1*1000
      IF(MED.EQ.1) THEN
      CALL USET(COL1)
      CALL UMOVE(X(I1),Y(I1))
      CALL UPEN(X(I2),Y(I2))
      WRITE(65,*) X(I1),Y(I1),X(I2),Y(I2)
      CALL USET(COLOR)
      END IF
      DO 662 IC=1,3
      IF(VERTEX(I,IC).EQ.I1) GO TO 662
      IF(VERTEX(I,IC).EQ.I2) GO TO 662
      I3=VERTEX(I,IC)
      I4=I3
      ZT(1)=Z(I1)
      ZT(2)=Z(I2)
      ZT(3)=Z(I3)
      ZT(4)=Z(I4)
      GO TO 668
  662 CONTINUE
  668 CONTINUE
      WRITE(51,*) I,J,I1,I2,I3,I4
     +,ZT(1),ZT(2),ZT(3),ZT(4)
      DO 680 II=1,NC
      ZZ=CL(II)
      IF(ZZ.EQ.ZT(1)) ZT(1)=ZT(1)+0.01*ZINT
      IF(ZZ.EQ.ZT(2)) ZT(2)=ZT(2)-0.01*ZINT
      IF(ZZ.EQ.ZT(3)) ZT(3)=ZT(3)+0.01*ZINT
      IF(ZZ.EQ.ZT(4)) ZT(4)=ZT(4)-0.01*ZINT
  680 CALL DRAWIT(I,J,ZZ,I1,I2,I3,I4,II)
  605 CONTINUE
  600 CONTINUE
      CLOSE(65)
      END IF
      DO 383 I=MSTART,MB
      DO 383 J=1,3
      IF(SIDE(I,J).GT.0) THEN
      K=SIDE(I,J)
      I1=K/1000
      I2=K-I1*1000
      IF(Y(I1).EQ.YMIN.AND.Y(I2).EQ.YMIN) GO TO 383
      IF(Y(I1).EQ.YMAX.AND.Y(I2).EQ.YMAX) GO TO 383
      IF(X(I1).EQ.XMIN.AND.X(I2).EQ.XMIN) GO TO 383
      IF(X(I1).EQ.XMAX.AND.X(I2).EQ.XMAX) GO TO 383
      GO TO 384
      END IF
  383 CONTINUE
      MSTART=MB
      GO TO 385
```

```
  384 MSTART=I
  385 CONTINUE
      M=MSTART
      CALL PUAIN
  950 CONTINUE
      NUMBR=IBL
      NP=NPT
      CLOSE(51)
      RETURN
      END
C
      SUBROUTINE NODES(I,J,NTRI)
      COMMON/NODEL/NODE(17000),NAV,NPMAX
      IF(NTRI.GT.2000) THEN
      PRINT*,' MORE THAN 2000 TRIANGLES -- PROBLEM TOO BIG !'
      STOP
      END IF
      M=I
      LL=NODE(I)
      LP=LL/100000
      LCC=LL-LP*100000
      I1=I
C     I2=LP*100000+LCC
      NODE(I)=(LP+1)*100000+LCC
   40 KCC=LCC
      IF(KCC.EQ.0) THEN
      NODE(M)=NODE(M)+NAV
      NODE(NAV)=J*100000
      NODE(NAV+1)=I*100000+NTRI
      NAV=NAV+2
      RETURN
      END IF
      M=LCC
      LL=NODE(KCC)
      LP=LL/100000
      LCC=LL-LP*100000
      GO TO 40
      END
C
      SUBROUTINE SEED
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/CONTOUR/NC,CL(500),IANS,FACT,PX,PY,PZT,CLS
      COMMON/BOUND/NCON,LCON(999),MED,MCON
      COMMON/CELLS/NUM(1000),IRN(100),IRD(100),KRD,AVX,AVY
      COMMON/NODEL/NODE(17000),NAV,NPMAX
      INTEGER SIDE,VERTEX,P1,P2,P3
      P1=NUM(NP/3)
      IF(NCON.GT.0) P1=LCON(1)
      DS=2000000.
      DO 100 IP=1,NP
      I=NUM(IP)
      IF(NCON.GT.0) THEN
      DO 707 LC=1,NCON
      IF(I.EQ.LCON(LC)) GO TO 708
  707 CONTINUE
      GO TO 100
      END IF
  708 CONTINUE
      IF(I.EQ.P1) GO TO 100
      D=(X(I)-X(P1))**2+(Y(I)-Y(P1))**2
      IF(D.LT.0.1) GO TO 100
      IF(D.GT.DS) GO TO 100
      DS=D
      P2=I
  100 CONTINUE
      DS=2000000.
      DO 200 IP=1,NP
      I=NUM(IP)
      IF(NCON.GT.0) THEN
      DO 807 LC=1,NCON
      IF(I.EQ.LCON(LC)) GO TO 808
  807 CONTINUE
      GO TO 200
      END IF
  808 CONTINUE
      IF(I.EQ.P1) GO TO 200
      IF(I.EQ.P2) GO TO 200
      D=(X(I)-X(P1))**2+(Y(I)-Y(P1))**2
      CALL AREA(X(P1),X(P2),X(I),Y(P1),Y(P2),Y(I),A)
      IF(ABS(A).LT.0.1) GO TO 200
      IF(D.LT.0.1) GO TO 200
      IF(D.GT.DS) GO TO 200
```

```
      DS=0
      P3=Y
  200 CONTINUE
      CALL AREA(X(P1),X(P2),X(P3),Y(P1),Y(P2),Y(P3),A)
      IF(A.LT.0.) GO TO 300
      Q=P2
      P2=P3
      P3=Q
  300 VERTEX(I,1)=P1
      VERTEX(I,2)=P2
      VERTEX(I,3)=P3
      NG=P1*1000+P2
      IF(P2.LT.P1) NG=P2*1000+P1
      SIDE(I,1)=NG
      NG=P2*1000+P3
      IF(P3.LT.P2) NG=P3*1000+P2
      SIDE(I,2)=NG
      NG=P3*1000+P1
      IF(P1.LT.P3) NG=P1*1000+P3
      SIDE(I,3)=NG
      CALL NODES(P1,P2,I)
      CALL NODES(P1,P3,I)
      CALL NODES(P2,P1,I)
      CALL NODES(P2,P3,I)
      CALL NODES(P3,P1,I)
      CALL NODES(P3,P2,I)
      RETURN
      END
C
      SUBROUTINE DLAREA(A1,A2,A3,B1,B2,B3,A)
      D=SQRT(MAX((A1-A2)**2+(B1-B2)**2,(A2-A3)**2+(B2-B3)**2,
     +       (A1-A3)**2+(B1-B3)**2))
      A=A1*B2+A2*B3+A3*B1-A1*B3-A2*B1-A3*B2
      IF(ABS(A/D).LT.0.04*D) A=0.
      IF(A.GT.-0.0000001.AND.A.LT.0.0000001) A=0.
      RETURN
      END
C
      SUBROUTINE AREA(A1,A2,A3,B1,B2,B3,A)
      A=A1*(B2-B3)+A2*(B3-B1)+A3*(B1-B2)
      IF(A.GT.-0.0000001.AND.A.LT.0.0000001) A=0.
      RETURN
      END
C
      SUBROUTINE CORCOR(I)
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      INTEGER SIDE,VERTEX,P1,P2,P3
      DO 100 J=1,3
      CX(J)=X(VERTEX(I,J))
      CY(J)=Y(VERTEX(I,J))
      CZ(J)=Z(VERTEX(I,J))
  100 CONTINUE
      RETURN
      END
C
      SUBROUTINE ORDER(N,M,I)
      N=M*1000+I
      IF(I.LT.M) N=I*1000+M
      RETURN
      END
C
      SUBROUTINE DRAWIT(M,MB,ZZ,I1,I2,I3,I4,KJ)
      INTEGER SIDE,VERTEX
      COMMON/STUFF/ZT(4)
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/CRAP/QDEL,MM
      COMMON/BOX/IBN(72,54)
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      COMMON/HVY/HEAVY,NUMH(500)
      COMMON/HHVY/HLET(52)
      CHARACTER HLET*2
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER*2 LET(52)
      CHARACTER BLET*2
      DATA QQ/.5/,TOL/2./
      DATA LET/'a\',' b\',' c\',' d\',' e\',' f\',
     +   'g\',' h\',' i\',' j\',' k\',' l\',' m\',' n\',
```

```
     +  ,O\',D\',Q\',R\',S\',T\',U\',V\',
     +  ,W\',X\',Y\',Z\',A\',B\',C\',D\',
     +  ,E\',F\',G\',H\',I\',J\',K\',L\',
     +  ,M\',N\',O\',P\',Q\',R\',S\',T\',
     +  ,U\',V\',W\',X\',Y\',Z\'/
      DATA IFLOP/1/
      IBOLD=1
      IF(HEAVY.NE.0.0) THEN
      III=(ABS(ZZ)/HEAVY)+0.01
      HH=ABS(ZZ)-III*HEAVY
      ENDIF
      IF(ABS(HH).LT.0.1*HEAVY) IBOLD=2
      IF(HEAVY.EQ.0.) IBOLD=1
      CALL LOC(ZZ,ZT(1),ZT(2),X12,Y12,I1,I2,ISCC)
      IF(ISCC.EQ.0) RETURN
      CALL LOC(ZZ,ZT(1),ZT(3),XMB,YMB,I1,I3,ISCC)
      IF(ISCC.EQ.1) GO TO 100
      CALL LOC(ZZ,ZT(2),ZT(3),XMB,YMB,I2,I3,ISCC)
  100 CONTINUE
      CALL LOC(ZZ,ZT(1),ZT(4),XM,YM,I1,I4,ISCC)
      IF(ISCC.EQ.1) GO TO 200
      CALL LOC(ZZ,ZT(2),ZT(4),XM,YM,I2,I4,ISCC)
  200 CONTINUE
      IF(ISCC.EQ.0) RETURN
      IF(M.EQ.MB) GO TO 450
      KBOLD=IBOLD
      IF(MM.EQ.1) KBOLD=1
      DO 763 JBOLD=1,KBOLD
      FBL=0.
      IF(JBOLD.EQ.1.AND.IBOLD.EQ.2) FBL=0.75*FSOFT
      IF(JBOLD.EQ.2) FBL=-FSOFT*0.75
      CALL USET(COL1)
      IF(IBOLD.GT.1) CALL USET(COL2)
      X2=X12
      Y2=Y12
      IF((XM-X2)**2+(YM-Y2)**2.LT.
     + (XMB-X2)**2+(YMB-Y2)**2) GO TO 300
      X1=(XMB+X2)/2.
      Y1=(YMB+Y2)/2.
      X3=(XM+X2)/2.
      Y3=(YM+Y2)/2.
      GO TO 350
  300 X1=(XM+X2)/2.
      Y1=(YM+Y2)/2.
      X3=(XMB+X2)/2.
      Y3=(YMB+Y2)/2.
  350 CONTINUE
      XN=X1
      YN=Y1
C FIND CIRCLE WHICH INSCRIBES (1-2-3)....A1 IS ANGLE 1-2-3
      D1=SQRT((X1-X2)**2+(Y1-Y2)**2)
      D2=SQRT((X1-X3)**2+(Y1-Y3)**2)
      D3=SQRT((X2-X3)**2+(Y2-Y3)**2)
      SL=0.5*(D1+D2+D3)+0.00001
      RL=ABS((SL-D1)*(SL-D2)*(SL-D3))/SL)
      RL=SQRT(RL)
      IF(SL-D2.LT.0.01) THEN
      XNOW=X2
      YNOW=Y2
      IF(YNOW-Y1.EQ.0.0.AND.XNOW-X2.EQ.0.) ANG=0.
      IF(YNOW-Y1.NE.0.0.OR.XNOW-X1.NE.0.) ANG=ATAN2(YNOW-Y1,XNOW-X1)
      CALL UMOVE(X1+FBL*SIN(ANG),Y1-FBL*COS(ANG))
      CALL UPEN(XNOW+FBL*SIN(ANG),YNOW-FBL*COS(ANG))
      GO TO 400
      END IF
      TANA=RL/(SL-D2)
      A1=ATAN(TANA)
      SIG=1.
C DEL IS THE DELTA OF THE CURVE 1-2-3
      CALL AREA(X1,X2,X3,Y1,Y2,Y3,A)
      IF(A.LT.0.0) SIG=-1.
      DEL=(1.5707963-A1)*SIG*57.29578
      DEL=DEL*2.
      PDEL=QDEL
  378 CONTINUE
C MOVE POINT 1 SUCH THAT THE DISTANCE BETWEEN THE PI & THE CURVE
C IS LESS THAN OR EQUAL TO QDEL (SET QDEL=1 FOR SHARP CONTOURS).
      D2=(X2-X1)**2+(Y2-Y1)**2
      CHORD=D2*COS(1.5707963-A1)**2
      DCH=QDEL*SIN(1.5707963-A1)/(1.-COS(1.5707963-A1))
      DD2=SQRT(D2)
      IF(DCH.LT.DD2-0.05) THEN
C NEW COORDINATES FOR POINT 1 (IF NECESSARY)
```

```
      X1=X1+(DD2-DCH)*(X2-X1)/DD2
      Y1=Y1+(DD2-DCH)*(Y2-Y1)/DD2
      GO TO 378
      END IF
C     ..SIMILAR TRIANGLES..
      R=SQRT(CHORD*D2/(D2-CHORD))
      AA=ATAN2(Y2-Y1,X2-X1)
C AND YC COORDINATES OF CENTER OF CURVE OF RADIUS RAD
      XC=X1-R*SIN(AA)*SIG
      YC=Y1+R*COS(AA)*SIG
      RAD=((X1-XC)**2+(Y1-YC)**2)
C MAKE SURE THAT THE CURVE CROSSES THE COMMON EDGE OF THE TWO
C TRIANGLES.  IF NOT, MOVE THE PT (POINT 1) CLOSER TO POINT 2.
      IF(X(I2)-X(I1).NE.0) GO TO 837
      SX1=X(I1)
      SX2=X(I1)
      FB=-2.*YC
      FC=SX1*SX1+XC*XC+YC*YC-RAD-2.*SX1*XC
      F2=FB*FB-4.*FC
      IF(F2.LT.0.) GO TO 834
      SY1=(-FB+SQRT(F2))/2.
      SY2=(-FB-SQRT(F2))/2.
      GO TO 838
 837  GM=(Y(I2)-Y(I1))/(X(I2)-X(I1))
      GB=Y(I1)-GM*X(I1)
      GA=SQRT(RAD)
      GH=XC
      GK=YC
      FA=1.+GM*GM
      FB=2.*(GM*GB-GM*GK-GH)
      FC=GH*GH+GB*GB+GK*GK-GA*GA-2.*GK*GB
      F2=(FB*FB-4.*FA*FC)
      IF(F2.LT.0.) GO TO 834
C  SX1,SX2,SY1,SY2, ARE COORDINATES OF INTERSECTIONS OF THE CIRCLE
C  WITH THE COMMON EDGE.  SEE IF INTERSECTION IS BETWEEN ENDPOINTS
C  OF THE EDGE
      SX1=(-FB+SQRT(F2))/(2.*FA)
      SX2=(-FB-SQRT(F2))/(2.*FA)
      SY1=GM*SX1+GB
      SY2=GM*SX2+GB
 838  D12=(X(I1)-X(I2))**2+(Y(I1)-Y(I2))**2
      DRA=(SX1-X(I1))**2+(SY1-Y(I1))**2
      DRB=(SX1-X(I2))**2+(SY1-Y(I2))**2
      IF(DRA.LE.D12.AND.DRB.LE.D12) GO TO 935
      DRA=(SX2-X(I1))**2+(SY2-Y(I1))**2
      DRB=(SX2-X(I2))**2+(SY2-Y(I2))**2
      IF(DRA.LE.D12.AND.DRB.LE.D12) GO TO 935
 834  CONTINUE
C  REDUCE QDEL IF NECESSARY
      QDEL=0.90*QDEL
      IF(QDEL.LT.1.) GO TO 935
      GO TO 378
 935  QDEL=PDEL
      ANG=ATAN2(Y2-Y1,X2-X1)
      CALL UMOVE(XN+FBL*SIN(ANG),YN-FBL*COS(ANG))
      CALL UPEN(X1+FBL*SIN(ANG),Y1-FBL*COS(ANG))
      IF(IBOLD.EQ.2.AND.MM.EQ.2) THEN
      CALL CORCOR(M)
      CALL AREA(CX(1),CX(2),X1,CY(1),CY(2),Y1,A1)
      CALL AREA(CX(2),CX(3),X1,CY(2),CY(3),Y1,A2)
      CALL AREA(CX(3),CX(1),X1,CY(3),CY(1),Y1,A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0) GO TO 1703
      CALL CORCOR(MB)
      CALL AREA(CX(1),CX(2),X1,CY(1),CY(2),Y1,A1)
      CALL AREA(CX(2),CX(3),X1,CY(2),CY(3),Y2,A2)
      CALL AREA(CX(3),CX(1),X1,CY(3),CY(1),Y1,A3)
      IF(A1.GT.0.OR.A2.GT.0.OR.A3.GT.0) GO TO 1705
 1703 CALL CONSIS(X1+FBL*SIN(ANG),Y1-FBL*COS(ANG),ZNEW)
      IF(ZNEW.LT.ZZ) CALL UPEN(X1+8.*FBL*SIN(ANG),Y1-6.*FBL*COS(ANG))
      IF(ZNEW.GT.ZZ) CALL UPEN(X1-8.*FBL*SIN(ANG),Y1+6.*FBL*COS(ANG))
 1705 CALL UMOVE(X1+FBL*SIN(ANG),YN-FBL*COS(ANG))
      ENDIF
      CALL UARC(XC+FBL*SIN(ANG),YC-FBL*COS(ANG),DEL)
      CALL UWHERE(XNOW,YNOW)
      ANG=ATAN2(Y3-Y2,X3-X2)
 400  CONTINUE
      CALL UPEN(X3+FBL*SIN(ANG),Y3-FBL*COS(ANG))
      IF(HEAVY.EQ.0.OR.IBOLD.EQ.2) THEN
      IF(XNOW.LT.XWMIN.OR.XNOW.GT.XWMAX) GO TO 890
      IF(YNOW.LT.YWMIN.OR.YNOW.GT.YWMAX) GO TO 890
      IF(X3-X2.EQ.0.) BANG=0.
      IF(X3-X2.NE.0.) BANG=(Y3-Y2)/(X3-X2)
      VSIGN=5.
```

```
      IF(BANG.GT.0.) VSIGN=-30.
      CALL UMOVE((XNOW+X3+VSIGN*FSOFT)/2.,(YNOW+Y3)/2.)
      IBX=(((XNOW-XWMIN)/FSOFT)/60.1+2
      IBY=(((YNOW-YWMIN)/FSOFT)/60.1+2.
      IF(IBX.LT.2) IBX=2
      IF(IBX.GT.71) IBX=71
      IF(IBY.LT.2) IBY=2
      IF(IBY.GT.53) IBY=53
      IF(IBN(IBX,IBY).NE.0) GO TO 890
      DO 893 IB1=IBX-1,IBX+1
      DO 893 IB2=IBY-1,IBY+1
  893 IBN(IB1,IB2)=1
      XKJ=KJ
      CALL USET('INTE')
      CALL USET(COL2)
      KF=-26
      IF(KJ.LE.26) KF=26
      IF(IBOLD.EQ.2) CALL UPRNT1(HLET(NUMH(KJ)),'TEXT')
      IF(HEAVY.EQ.0.) CALL UPRNT1(LET(KJ+KF),'TEXT')
      CALL USET(COLOR)
      CALL USET('INTE')
  890 CONTINUE
      ENDIF
  763 CONTINUE
      CALL USET(COL1)
      RETURN
  450 CONTINUE
      ANG=ATAN2(Y12-YM,X12-XM)
      KBOLD=IBOLD
      IF(MM.EQ.1) KBOLD=1
      DO 764 JBOLD=1,KBOLD
      CALL USET(COL1)
      IF(IBOLD.GT.1) CALL USET(COL2)
      FBL=0.
      IF(JBOLD.EQ.1.AND.IBOLD.EQ.2) FBL=FSOFT*0.75
      IF(JBOLD.EQ.2) FBL=-FSOFT*0.75
      CALL UMOVE(X12+FBL*SIN(ANG),Y12-FBL*COS(ANG))
      CALL UPEN((X12+XM+FBL*SIN(ANG))/2.,(Y12+YM-FBL*COS(ANG))/2.)
  764 CONTINUE
      CALL USET(COL1)
      RETURN
      END
C
      SUBROUTINE LOC(ZZ,Z1,Z2,XP,YP,I,J,ISCC)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CT(3),CZ(3),
     +     VERTEX(2000,3),SIDE(2000,3),NP
      ISCC=0
      C1=AMIN1(Z1,Z2)
      C2=AMAX1(Z1,Z2)
      IF(ZZ.LE.C1.OR.ZZ.GE.C2) RETURN
      IF((Z1-Z2).EQ.0.0) RETURN
      CM=(Z1-ZZ)/(Z1-Z2)
      XP=X(I)+(X(J)-X(I))*CM
      YP=Y(I)+(Y(J)-Y(I))*CM
      ISCC=1
      RETURN
      END
C
      SUBROUTINE PSTART(M)
      COMMON/DVICE/DEVC,PCX,PCY,PSFI,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      CHARACTER DEVC*4
      CHARACTER DEV*4
      CHARACTER COLOR*4,CLRBCK*4,PSIZE*1
      IF(M.EQ.1.OR.M.EQ.2) THEN
C   ..OUTPUT TO SCREEN OR PLOTTER...
      DEV=DEVC
      CALL UDEVIC(DEV)
      IF(DEV.EQ.'IBMH') CALL UDEVICE ('EGA ')
      IF(DEV.EQ.'HP4 ') CALL UHPSIZ(PSIZE)
      IF(DEV.EQ.'HP4 ') CALL UCOMPT(1)
      CALL USTART
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL USET('MEDIUM')
      IF(DEV.EQ.'IBMH') CALL USET(COLOR)
      CALL USET('PERCENT')
      IF(M.EQ.1) CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      IF(M.EQ.2) CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL USET('SOFTWARE')
```

```
      PSFT=1.
      IF(M.EQ.2) PSFT=.67
      CALL UPSET('VERT',40.*PSFT)
      CALL UPSET('HORI',20.*PSFT)
      IF(M.EQ.2) CALL USET('HARD')
      RETURN
      END IF
      IF(M.EQ.3) THEN
C     ..OUTPUT TO PRINTER
      DEV=DEVC
      CALL UDEVIC(DEV)
      CALL UEPSON('FX80',1.)
      CALL USTART
      CALL USET('PERCENT')
      CALL UDAREA(10.,75.,10.,50.)
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL USET('SOFT')
      CALL UPSET('VERT',30.)
      CALL UPSET('HORI',24.)
      RETURN
      END IF
      RETURN
      END
C
      SUBROUTINE PEND
      CALL UFLUSH
      CALL UEND
      RETURN
      END
C
      SUBROUTINE PUAIN
      CHARACTER IP*1
      IF(2.EQ.2) RETURN
      CALL UBELL
      CALL UMOVE(500.,500.)
      CALL UFLUSH
      CALL UGRIN(X,Y,IP)
      IF(IP.EQ.'E') THEN
      CALL UMOVE(400.,400.)
      CALL UPRNTI('\','TEXT')
      CALL UFLUSH
      CALL UERASE
      END IF
      RETURN
      END
C
      SUBROUTINE INF(NC)
      CHARACTER FLNE*14,BORD*2,HONOR*2,TYP*4
      CALL UMOVE(810.,560.)
      X=775.
      Y=750.
      IF(NC.GT.26) X=810.
      IF(NC.LT.26) Y=NC*25.+100.
      IF(Y.LT.550.) Y=550.
      CALL UMOVE(X,Y)
      CALL UPRNTI('CONTOURS\','TEXT')
      RETURN
      END
C
      SUBROUTINE PROFILE(ZMX,ZMN,MM,XMINX,XMAXX,YMINY,YMAXY,SM,RDASK)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/TRI/MBL,MB
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      DIMENSION XT(260),YT(260),ZT(260)
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER IP*1,JASK*1,FILEN*12,DESC*80,SLASH*1,IPP*2,YLABEL*40
      CHARACTER MINMAX*40,PFILE*20
      CHARACTER RDASK*1,ILX*1
      DATA SLASH/'\'/
      IPP=' '
      IP=
      CALL CLEAR
      PRINT*,'(2,25a) ARE PROFILE LINE ENDPOINTS TO BE LOCATED WITH'
      PRINT,'(A)','     THE CURSOR(C) OR FROM A NAMED FILE(F) ?(DEF=C) '
      READ '(A1)',JASK
      IF(JASK.EQ.' ') JASK='C'
```

```
      IF(IASK.EQ.'C') GO TO 7744
3334  PRINT*,' ENTER NAME OF FILE WITH X1,Y1,X2,Y2 '
      PRINT '(A)',' OR ENTER ''CON'' IF FROM KEYBOARD ) '
      READ '(A20)',PFILE
      IF(PFILE.EQ.'CON') THEN
      CALL CLEAR
      PRINT '(A)',' TYPE IN X1,Y1,X2,Y2 ) '
      READ*,X1,X2,Y1,Y2
      GO TO 7744
      ENDIF
      OPEN(33,FILE=PFILE,ERR=3334,STATUS='OLD')
      READ(33,*) X1,Y1,X2,Y2
      X1=(X1-SUBX)*FACX
      X1=((X1-XMINX)/SM)+100.
      X2=(X2-SUBX)*FACX
      X2=((X2-XMINX)/SM)+100.
      Y1=(Y1-SUBY)*FACY
      Y1=((Y1-YMINY)/SM)+100.
      Y2=(Y2-SUBY)*FACY
      Y2=((Y2-YMINY)/SM)+100.
      CALL UCRCLE(X1,Y1,5.)
      CALL UCRCLE(X2,Y2,5.)
      GO TO 7755
7744  CALL CLEAR
      PRINT*,'(2,25b) WANT CURSOR TO INDICATE NEAREST (TO UPPER '
      PRINT,'(A)',' RIGHT) DATA NODE POINT ? (Y/N) '
      READ '(A1)',ILX
      CALL CLEAR
      PRINT*,'(2,25c)'
      PRINT*,' POSITION CURSOR AT 1ST END OF PROF. LINE & HIT RETURN'
      PRINT*,'  USE (.)  AND 6 KEYS TO ''FINE'' POSITION CURSOR.'
      CALL UMOVE(100.,100.)
      CALL UBELL
      CALL KURSIN(X1,Y1,IP)
      IF(ILX.EQ.'Y') CALL CLOSEST(X1,Y1)
      CALL CLEAR
      PRINT*,'(2,25d)'
      PRINT*,' POSITION CURSOR AT 2ND END OF PROF. LINE & HIT RETURN'
      CALL USET(COL1)
      CALL UCRCLE(X1,Y1,5.)
      CALL UFLUSH
      CALL UBELL
      CALL KURSIN(X2,Y2,IP)
      IF(ILX.EQ.'Y') CALL CLOSEST(X2,Y2)
      CALL USET(COL1)
      CALL UCRCLE(X2,Y2,5.)
      CALL UFLUSH
7755  CONTINUE
      CALL CLEAR
      IF(X1.EQ.X2.AND.Y1.EQ.Y2) GO TO 7902
      PRINT*,'...WAIT..'
      EX=X1
      EY=Y1
      TX=X2
      TY=Y2
      IF(ABS(X2-X1).GE.0.1) THEN
      IF(X2.LT.X1) THEN
      X2=EX
      Y2=EY
      X1=TX
      Y1=TY
      GO TO 123
      ENDIF
      ENDIF
      IF(ABS(X2-X1).LT.0.1.AND.Y2.LT.Y1) THEN
      X2=TX
      Y2=EY
      X1=TX
      Y1=TY
      ENDIF
 123  CONTINUE
      XB1=X1
      YB1=Y1
      XE1=X2
      YE1=Y2
      DO 333 I=1,1041,2
      IF(XB1.LE.0.0.AND.XE1.LE.0.0) GO TO 335
      IF((X2-X1).NE.0.0) F=(I-1)/(X2-X1)
      IF((X2-X1).EQ.0.0) F=(I-1)/(Y2-Y1)
      IF(XB1.GT.0.) THEN
      XB1=X1+F*(X2-X1)
      YB1=Y1+F*(Y2-Y1)
      ENDIF
```

```
      IF(XE1.GT.0.0) THEN
      XE1=X2-F*(X2-X1)
      YE1=Y2-F*(Y2-Y1)
      ENDIF
C  1.MBL WILL INCLUDE POINTS IN NON-CONTOURED AREA
      DO 334 J=1,MB
      CALL CORCOR(J)
      IF(XB1.GT.0.) THEN
      CALL AREA(CX(1),CX(2),XB1,CY(1),CY(2),YB1,A1)
      CALL AREA(CX(2),CX(3),XB1,CY(2),CY(3),YB1,A2)
      CALL AREA(CX(3),CX(1),XB1,CY(3),CY(1),YB1,A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      XB1=-XB1
      YB1=-YB1
      ENDIF
      ENDIF
      IF(XE1.GT.0.) THEN
      CALL AREA(CX(1),CX(2),XE1,CY(1),CY(2),YE1,A1)
      CALL AREA(CX(2),CX(3),XE1,CY(2),CY(3),YE1,A2)
      CALL AREA(CX(3),CX(1),XE1,CY(3),CY(1),YE1,A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      XE1=-XE1
      YE1=-YE1
      ENDIF
      ENDIF
  334 CONTINUE
  333 CONTINUE
  335 CONTINUE
      X1=-XB1
      Y1=-YB1
      X2=-XE1
      Y2=-YE1
      CALL USET(COL1)
      CALL UCRCLE(X1,Y1,5.)
      CALL UCRCLE(X2,Y2,5.)
      CALL UFLUSH
      NJ=MAX(ABS((X2-X1)/5.),ABS((Y2-Y1)/5.))
      IF(NJ.LE.1) RETURN
      DO 100 I=1,NJ
      OI=I
      ONJ=NJ
      F=(OI-1)/(ONJ-1)
      XT(I)=X1+F*(X2-X1)
      YT(I)=Y1+F*(Y2-Y1)
  100 CONTINUE
      ZMIN=99.E30
      ZMAX=-ZMIN
      II=1
      DO 210 J=1,NJ
      ZT(J)=-999
C     FIND TRIANGLE POINT IS IN.....
      CALL UALPHA
      DO 200 IG=II,MB+II-1
      KK=0
      IF(IG.GT.MB) KK=MB
      I=IG-KK
C     IF(I.LT.MBL) GO TO 200
      CALL CORCOR(I)
      CALL AREA(CX(1),CX(2),XT(J),CY(1),CY(2),YT(J),A1)
      CALL AREA(CX(2),CX(3),XT(J),CY(2),CY(3),YT(J),A2)
      CALL AREA(CX(3),CX(1),XT(J),CY(3),CY(1),YT(J),A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      CALL UALPHA
C     FOUND IT...
      CALL CONSIS(XT(J),YT(J),ZT(J))
      IF(ZT(J).GT.ZMAX) ZMAX=ZT(J)
      IF(ZT(J).LT.ZMIN) ZMIN=ZT(J)
      II=MAX(1,I-(MB/10))
      GO TO 210
      ENDIF
  200 CONTINUE
  210 CONTINUE
   88 CALL CLEAR
      PRINT*,'(2,26) Want profile (D)rawn on this plot, sent to ',
      PRINT,'(A1)','   (F)ile. or (B)oth ? (Enter D,F or B) DEF=D ) '
      READ,'(A1)',JASK
      IF(JASK.EQ.' ') JASK='D'
      IF(JASK.NE.'D'.AND.JASK.NE.'F'.AND.JASK.NE.'B') GO TO 88
   91 CONTINUE
      IF(JASK.EQ.'F'.OR.JASK.EQ.'B') THEN
      CALL CLEAR
      PRINT,'(A)',
     +' (2,27) ENTER A FILE NAME (DEF=D.PRO) ) '
```

```
      READ '(A12)',FILEN
      IF(FILEN.EQ.' ') FILEN='O.PRO'
      OPEN(14,FILE=FILEN,ERR=87)
      GO TO 89
   87 PRINT*,FILEN,' ALREADY USED-- HIT RETURN, THEN TRY A NEW NAME'
      CALL UBELL
      CALL UPAUSE
      GO TO 91
   89 CALL CLEAR
      PRINT*,' (2,28) ENTER A ONE LINE DESCRIPTION (OR RETURN) '
      READ '(A70)',DESC
      CALL CLEAR
      PRINT,'(A)',' (2,29) ENTER SECTION LABEL (LIKE A FOR A-A) ) '
      READ '(A1)',IP
      WRITE(IPP,'(2A1)') IP,SLASH
      PRINT,'(A)'
     +' (2,30) ENTER Y-AXIS LABEL (DEFAULT=ELEVATION) ) '
      READ '(A40)',YLABEL
      IF(YLABEL.EQ.'                                        ')
     + YLABEL='ELEVATION'
      VX=SM*(X1-100.)+XMINX
      VY=SM*(Y1-100.)+YMINY
      WRITE(14,*) VX,VY
      VX=SM*(X2-100.)+XMINX
      VY=SM*(Y2-100.)+YMINY
      WRITE(14,*) VX,VY
      CALL UMOVE(X1-20.,Y1)
      CALL UPRNTI(IPP,'TEXT')
      CALL UMOVE(X2+20.,Y2)
      CALL UPRNTI(IPP,'TEXT')
      CALL UMOVE(X1,Y1)
      CALL UPEN(X2,Y2)
      WRITE(14,'(A70)') DESC
      WRITE(14,'(A1)') IP
      WRITE(14,'(A40)') YLABEL
      DO 603 I=1,NJ
  603 WRITE(14,*) XT(I),YT(I),ZT(I)
      CLOSE(14)
      IF(JASK.EQ.'F') GO TO 7902
      ENDIF
      CALL CLEAR
      PRINT 12,ZMN,ZMX,ZMIN,ZMAX
   12 FORMAT(8H MIN EL ,F8.2,8H MAX EL ,F8.2,20H -ALL POINTS ON PLOT,
     +/,8H         ,F8.2,8H         ,F8.2,24H -POINTS ON PROFILE LINE)
 6634 PRINT*,'(2,31) ENTER YOUR MIN AND MAX PROFILE ELEVATIONS.'
      PRINT,'(A)'
     +' (DEF=MIN AND MAX ELEV OF ALL POINTS ON PLOTS) ) '
      READ '(A40)',MINMAX
      IF(MINMAX.EQ.' ') ZZMIN=ZMN
      IF(MINMAX.EQ.' ') ZZMAX=ZMX
      IF(MINMAX.NE.' ') THEN
      OPEN(43,FILE='M43243')
      WRITE(43,'(A40)') MINMAX
      REWIND 43
      READ(43,*,ERR=6634) ZZMIN,ZZMAX
      CLOSE(43,STATUS='DELETE')
      ENDIF
      CALL CLEAR
 7902 CONTINUE
      IF(MM.EQ.2) THEN
      CALL PEND
      DEV='HP4'
      CALL PSTART(2)
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL UCHPEN(LPEN(4))
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      ENDIF
      IF(RDASK.EQ.'Y') THEN
      CALL USET(COLOR)
      CALL UCHPEN(LPEN(1))
      IF(MM.EQ.2) CALL ROCON(ZMX,ZMN,0,XMINX,XMAXX,YMINY,YMAXY,SM)
      CALL UCHPEN(LPEN(4))
      RDASK='N'
      CALL USET(COL1)
      ENDIF
      CLOSE(81)
      IF(X1.EQ.X2.AND.Y1.EQ.Y2) RETURN
      CALL UMOVE(X1-20.,Y1)
      CALL UPRNTI(IPP,'TEXT')
      CALL UMOVE(X2+20.,Y2)
      CALL UPRNTI(IPP,'TEXT')
      IF(IP.NE.' ') CALL UPRNTI('*\'.'TEXT')
```

```
      CALL UMOVE(X1,Y1)
      CALL UPEN(X2,Y2)
      CALL UFLUSH
      IF(JASK.EQ.'F') RETURN
C     DRAW THE PROFILE
      ZDIF=(ZZMAX-ZZMIN)/50.
      XP=0.
      THE=ATAN2(Y2-Y1,X2-X1)
      CALL USET(COL3)
      DO 300 I=2,NJ
      IF(ZT(I).EQ.-999.) GO TO 300
      IF(ZT(I-1).EQ.-999.) GO TO 300
      IF(ZT(I).GT.ZZMAX) ZT(I)=ZZMAX
      IF(ZT(I).LT.ZZMIN) ZT(I)=ZZMIN
      ZZ=(ZT(I)-ZZMIN)/ZDIF
      CALL UMOVE(XT(I),YT(I))
      CALL UPEN(XT(I)-ZZ*SIN(THE),YT(I)+ZZ*COS(THE))
      CALL UPEN(XT(I-1)-ZZ*SIN(THE),YT(I-1)+ZZ*COS(THE))
      CALL UPEN(XT(I-1),YT(I-1))
      CALL UPEN(XT(I),YT(I))
  300 CONTINUE
      CALL USET(COLOR)
      RETURN
      END
C
      SUBROUTINE CLEAR
      CHARACTER BLN1*40,BLN2*38
      DATA BLN1/'                                        '/
      DATA BLN2/'                                      '/
      CALL UHOME
      CALL UALPHA
      PRINT*,BLN1,BLN2
      PRINT*,BLN1,BLN2
      PRINT*,BLN1,BLN2
      PRINT*,BLN1,BLN2
      PRINT*,BLN1,BLN2
      CALL UHOME
      CALL UALPHA
      RETURN
      END
C
      SUBROUTINE KURSIN(XP,YP,IP)
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER IP*1,CLR*4,LP*1
      CLR=COLOR
      COLOR=COL1
      CALL USET(COLOR)
    5 CALL UGRIN(XP,YP,IP)
      XNOW=XP
      YNOW=YP
      IF(IP.EQ.'6'.OR.IP.EQ.'-'.OR.IP.EQ.','.OR.IP.EQ.'('.OR.
     +IP.EQ.'.'.OR.IP.EQ.')') THEN
   83 DX=0.
      DY=0.
      IF(IP.EQ.'6') DY=-3.
      IF(IP.EQ.'-') DY=3.
      IF(IP.EQ.','.OR.IP.EQ.'(') DX=-3.
      IF(IP.EQ.'.'.OR.IP.EQ.')') DX=3.
   73 CALL UMOVE(XNOW,YNOW)
      CALL UPEN(XNOW+DX,YNOW+DY)
      GX=XNOW
      GY=YNOW
      GX1=XNOW+DX
      GY1=YNOW+DY
      CALL UWHERE(XNOW,YNOW)
      IP=
      CALL UMOVE(5.,700.)
      CALL UGRIN(AA,BB,IP)
      IF(IP.EQ.'6'.OR.IP.EQ.'-'.OR.
     +IP.EQ.','.OR.IP.EQ.'('.OR.IP.EQ.'.'.OR.IP.EQ.')') THEN
      CALL UMOVE(GX,GY)
      CALL USET(CLRBCK)
      CALL UPEN(GX1,GY1)
      CALL USET(COLOR)
      GO TO 83
      ENDIF
      CALL UMOVE(GX,GY)
      CALL USET(CLRBCK)
      CALL UPEN(GX1,GY1)
      CALL USET(COLOR)
```

```
   93 CONTINUE
      XP=XNOW
      YP=YNOW
      COLOR=CLR
      CALL USET(COLOR)
      RETURN
      ENDIF
      COLOR=CLR
      CALL USET(COLOR)
      RETURN
      END
C
      SUBROUTINE CONSTS(XX,YY,ZZ)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            SIDE(2000,3),VERTEX(2000,3),NP
      C=((CX(1)-CX(2))*(CZ(1)-CZ(3))-(CX(1)-CX(3))*(CZ(1)-CZ(2)))/
     +  ((CX(1)-CX(2))*(CY(1)-CY(3))-(CX(1)-CX(3))*(CY(1)-CY(2)))
      B=((CY(1)-CY(2))*(CZ(1)-CZ(3))-(CY(1)-CY(3))*(CZ(1)-CZ(2)))/
     +  ((CY(1)-CY(2))*(CX(1)-CX(3))-(CY(1)-CY(3))*(CX(1)-CX(2)))
      A=CZ(1)-B*CX(1)-C*CY(1)
      ZZ=A+B*XX+C*YY
      RETURN
      END
C
      SUBROUTINE OUTPRO(MM)
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      CHARACTER COLOR*4,CLRBCK*4,PSIZE
      CHARACTER FILEN*12,JASK*1,OLDPRO*12
      DATA OLDPRO/'D.PRO       '/
      CALL CLEAR
      PRINT '(A\)',
     +  ' (2.35) WANT TO DRAW A PREVIOUSLY SAVED PROFILE ? (Y/N) ) '
      READ '(A1)',JASK
      CALL CLEAR
 5332 CONTINUE
      IF(JASK.EQ.'Y') THEN
      PRINT*,' (2.36) ENTER SAVED FILE''S NAME (DEF= ',OLDPRO,')'
      PRINT '(A)','   OR  Q  TO QUIT )'
      READ '(A12)',FILEN
      IF(FILEN.EQ.' ') FILEN=OLDPRO
      IF(FILEN.EQ.'Q') RETURN
      OPEN(14,FILE=FILEN,STATUS='OLD',ERR=5333)
      OLDPRO=FILEN
      GO TO 5334
 5333 CALL CLEAR
      PRINT*,FILEN,' NOT FOUND..  TRY AGAIN'
      GO TO 5332
 5334 CONTINUE
      CALL UCHPEN(LPEN(4))
      CALL DRWPRF(MM)
      CLOSE(14)
      CALL CLEAR
      PRINT '(A\)',' (2.35) WANT ANOTHER PROFILE PLOT ? (Y/N) ) '
      READ '(A1)',JASK
      GO TO 5332
      ENDIF
      RETURN
      END
C
      SUBROUTINE INPRO(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,YMAXY,SM,IW)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/TRI/MBL,MB
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER COLOR*4,CLRBCK*4,PSIZE*1,DEV*4
      CHARACTER JASK*1,FIMESH*12,PASK*1,REFEL*10,OLDMSH*12
      CHARACTER RDASK*1,OUM*1
      DATA OLDMSH/'M.MSH       '/
      RDASK='N'
      CALL CLEAR
      PRINT '(A\)', '(2.23) WANT TO DRAW A PROFILE ? (Y/N) ) '
      READ '(A1)',JASK
 1456 CONTINUE
      IF(JASK.EQ.'Y') THEN
      IF(IW.EQ.0) THEN
  102 PRINT*,'(2.40) ENTER NAME OF "MESH" FILE (DEF= ',OLDMSH,')'
```

```
      PRINT '(A\)',' OR ''Q'' TO QUIT ) '
      READ '(A12)',FIMESH
      IF(FIMESH.EQ.'Q') RETURN
      IF(FIMESH.EQ.'            ') FIMESH=OLDMSH
      OPEN(81,FILE=FIMESH,STATUS='OLD',ERR=100)
      OLDMSH=FIMESH
      GO TO 103
  100 PRINT*,'FILE NOT FOUND.  TRY AGAIN'
      GO TO 102
  103 READ(81,*) SUBX,SUBY
      READ(81,*) FACX,FACY
      READ(81,*) XMIN,XMAX,YMIN,YMAX
      READ(81,*) ZMAX,ZMIN,XMINX,XMAXX
      READ(81,*) YMINY,YMAXY,SM
      READ(81,*) NP,MB,MBL
      DO 105 I=1,NP
  105 READ(81,*) X(I),Y(I),Z(I),NCC(I)
      DO 106 I=1,MB
  106 READ(81,*) (VERTEX(I,J),J=1,3)
      CALL CLEAR
      PRINT '(A\)',
     +' (2.24a) WANT TO REDRAW THE CONTOURS ? (Y/N) ) '
      READ '(A1)',RDASK
      ENDIF
      CALL CLEAR
      PRINT '(A\)',
     +' (2.24b) WANT THE VOLUME UNDER THE MESH ? (Y/N) ) '
      READ '(A1)',PASK
      IF(PASK.EQ.'Y') THEN
      CALL CLEAR
      PRINT*,'MINIMUM ELEVATION= ',ZMIN
 4455 PRINT '(A\)'
     +' (2.24c) ENTER REFERENCE ''BASE'' ELEVATION (DEF=0.0)) '
      READ '(A10)',REFEL
      IF(REFEL.EQ.'') REFELE=0.0
      IF(REFEL.NE.'') THEN
      OPEN(43,FILE='M33343')
      WRITE(43,'(A10)') REFEL
      REWIND 43
      READ(43,*,ERR=4455) REFELE
      CLOSE(43,STATUS='DELETE')
      ENDIF
      ATOT=0.0
      VOL=0.0
      VOL1=0.
      VOL2=0.
      DO 540 I=MBL,MB
      CALL CORCOR(I)
      CALL AREA(CX(1),CX(2),CX(3),CY(1),CY(2),CY(3),A)
      A=SM*SM*ABS(A)/2.
      ATOT=ATOT+A
      ZAV=(CZ(1)+CZ(2)+CZ(3))/3.
      VOL=VOL+A*(ZAV-REFELE)
      VOL1=VOL1+A*(ZAV-ZMIN)
      VOL2=VOL2+A*(ZAV-ZMAX)
  540 CONTINUE
      REF=(-VOL1*(ZMAX-ZMIN)/(VOL2-VOL1))+ZMIN
      ENDIF
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      CALL PEND
      DEV='IBMH'
      CALL PSTART(1)
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
C     DRAW POINTS
      CALL USET(COL2)
      CALL UOUTLN
      DO 1931 I=1,NP
      IF(NCC(I).GE.1000) CALL USET(COL4)
      IF(NCC(I).LT.1000) CALL USET(COL2)
      CALL UMOVE(X(I)-3.,Y(I))
      CALL UPEN(X(I)+3.,Y(I))
      CALL UMOVE(X(I),Y(I)-3.)
      CALL UPEN(X(I),Y(I)+3.)
 1931 CONTINUE
      CALL USET(COL2)
      CALL UPSET('VERT',30.)
      CALL UPSET('HORI',15.)
      CALL XYAXIS(XMINX,XMAXX,YMINY,YMAXY,SM,1)
      CALL USET(COLOR)
      IF(RDASK.EQ.'Y') THEN
      CALL RDCON(ZMAX,ZMIN,1,XMINX,XMAXX,YMINY,YMAXY,SM)
```

```
      OPEN(81,FILE=FIMESH)
      REWIND 81
      READ(81,*) FFF
      READ(81,*) FFF
      READ(81,*) FFF
      READ(81,*) FFF
      READ(81,*) FFF
      READ(81,*) FFF
      DO 1928 I=1,NP
1928  READ(81,*) FFF
      DO 1929 I=1,MB
1929  READ(81,*) III
      ENDIF
      IF (JASK.EQ.'Y') THEN
      CALL PROFILE(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,YMAXY,SM,RDASK)
      IF(PASK.EQ.'Y') THEN
      CALL USETT('SOFT')
      CALL UPSET('VERT',.30.*PSFT)
      CALL UPSET('HORI',.15.*PSFT)
      CALL UMOVE(100.,75.)
      CALL UPRNTI('VOL ABV Z= ','TEXT')
      CALL UPRNTI(REFELE,'REAL')
      CALL UPRNTI('=','TEXT')
      CALL UPRNTI(VOL,'REAL')
      CALL UPRNTI(' AV Z= ','TEXT')
      CALL UPRNTI(REF,'REAL')
      CALL UPRNTI(' AREA=','TEXT')
      CALL UPRNTI(ATOT,'REAL')
      IF(MM.EQ.2) CALL USETT('HARD')
      CALL CLEAR
      PRINT*,'VOL ABOVE ',REFELE,'=',VOL,'  AREA= ',ATOT
      PRINT*,'AV Z = ',REF
      CALL UBELL
      CALL UPAUSE
      ENDIF
      CALL CLEAR
      PRINT'(A)',' (2.23) WANT TO DRAW ANOTHER PROFILE ? (Y/N) > '
      READ '(A1)',JASK
      IF(JASK.EQ.'Y') GO TO 1456
      ENDIF
      ENDIF
      CALL UERASE
      CALL UBACKG(CLRBCK)
      RETURN
      END
C
      SUBROUTINE RDCON(EZMAX,EZMIN,IW,XMINX,XMAXX,YMINY,YMAXY,SM)
      INTEGER SIDE,VERTEX
      COMMON/TRI/MBL,MB
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +       VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/CONTOUR/NC,CL(500),IANS,FACT,PX,PY,PZT,CLS
      COMMON/STUFF/ZT(4)
      COMMON/JPOW/IPOW
      COMMON/BOUND/NCON,LCON(999),MED,MCON
      COMMON/HOUND/NUMCON,KCON(100)
      COMMON/CRAP/QDEL,MM
      COMMON/CELLS/NUM(1000),IRN(100),IRD(100),KRD,AVX,AVY
      COMMON/MINS/XMIN,XMAX,YMIN,YMAX
      COMMON/DRAWS/MBLNUM(100),NUMBR
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/BOX/IBN(72,54)
      COMMON/HVY/HEAVY,NUMH(500)
      COMMON/HHVY/HLET(52)
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      CHARACTER HLET*2
      CHARACTER DUM*72
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4
      CHARACTER COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER*2 LET(52),BLET,JASK*1,TITLE*48,TITTLE(48)*1,PASK*1
      CHARACTER COLR(7)*4,FILEN*12,FIMESH*12
      CHARACTER FINE*14,TYP*4,HONOR*2,BORD*2,IP*1,TMPLT*20
      CHARACTER CONVAL*20,CONVA(20)*1
      CHARACTER ASPACE*1,ALABEL(999)*10,BSLSH*1,BLABEL*11,BLAB(11)*1
      CHARACTER HELV*1,AASK*1,BASK*1,IDRAW*1
      DATA BSLSH/'\'/
      DATA XLAST/0./,YLAST/0./,ULAST/0./,VLAST/0./
      DATA LET/'a','b','c','d','e','f',
     +   'g','h','i','j','k','l','m','n',
```

```
     +  ,'G','D','O','R','S','L','U','V',
     +  ,'W','X','Y','Z','A','B','C','D',
     +  ,'E','F','G','H','I','J','K','L',
     +  ,'M','N','O','P','Q','R','S','T',
     +  ,'U','V','W','X','Y','Z'/
      DATA COLR/'YELL','RED ','GREE','BLUE','CYAN','MAGE','WHIT'/
C  SET IFLAG EQUAL TO 0 IF ZOOMED. 1 IF CLIPPED
      IFLAG=0
      IF(ABS(XUDMIN-10.*PCX).GT.0.001.AND.ABS(XUDMAX-75.*PCX)
     +      .GT.0.001) IFLAG=1
      IF(IFLAG.EQ.1) CALL UOUTIN
      CALL UPSET('VERT',30.*PSFT*FSOFT)
      CALL UPSET('HORI',20.*PSFT*FSOFT)
      DO 88 I=1,72
      DO 88 J=1,54
   88 IBN(I,J)=0
 2301 CALL CLEAR
      ZMIN=9.E30
      ZMAX=-ZMIN
      DO 2307 I=1,NP
      IF(X(I).GT.XWMIN.AND.X(I).LT.XWMAX.AND.Y(I).GT.YWMIN.AND.
     +   Y(I).LT.YWMAX) THEN
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
      ENDIF
 2307 CONTINUE
      CALL CLEAR
      HEAVY=0.
      CONVAL='                    '
      PRINT*,'(2,13) MIN EL.= ',ZMIN,' MAX EL.= ',ZMAX
      PRINT*,' ENTER BEGINNING CONTOUR, CONTOUR INTERVAL, HEAVY LINE'
      PRINT*,' MULTIPLE, AND NO. OF CONTOURS(0 MEANS SPAN RANGE).'
      PRINT,'(A)'
     +,' OR ''N'' FOR NO CONTOURING (DEF=AUTO) (H FOR HELP) ) '
      READ '(A20)',CONVAL
      IF(CONVAL.EQ.'NONE'.OR.CONVAL.EQ.'N') CONVAL='999999,1,0,1'
      IF(CONVAL.EQ.'H') THEN
      CALL CLEAR
      PRINT*,'FOUR NUMBERS ARE TO BE ENTERED(OR PRESS RETURN FOR AUTO)'
      PRINT*
     +,'THE 1ST IS THE BEGINNING CONTOUR VALUE, THE 2ND IS A + OR -'
      PRINT,'(A)','  CONTOUR INTERVAL, (PRESS RETURN FOR MORE INFO) ) '
      READ '(A1)',AASK
      CALL CLEAR
      PRINT*,' THE 3RD CAUSES ALL CONTOUR LINES WHICH ARE MULTIPLES OF'
      PRINT*,' OF THIS VALUE TO BE DRAWN HEAVY (PLOTTER) OR ANOTHER'
      PRINT*,' COLOR (SCREEN).  THE 4TH IS THE NO. OF LINES TO BE'
      PRINT '(A)'
     +,'  DRAWN (0 MEANS SPAN RANGE OF DATA). PRESS RETURN FOR MORE) '
      READ '(A1)',AASK
      CALL CLEAR
      PRINT*,'I.E. 5020,-2,10,6 MEANS START AT ELEV 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,'(A)','      PRESS RETURN TO CONTINUE )'
      READ '(A1)',AASK
      GO TO 2307
      ENDIF
      IF(CONVAL.EQ.'                    ') THEN
      ZD=(ZMAX-ZMIN)/26.
      BLOG=LOG10(ZD)
      LOG=BLOG
      IF(BLOG.LT.0.0) LOG=LOG-1.
      IDEL=(ZD/(10.0**LOG))+1.
      JDEL=IDEL
      IF(IDEL.GT.5) JDEL=10
      IF(IDEL.EQ.3) JDEL=4
      ZINT=JDEL*(10.0**LOG)
      HEAVY=ZINT*5
      IZ=(ZMIN/ZINT)+0.0001*ZMIN/ZINT
      ZZMIN=IZ*ZINT+ZINT
      NCM=0
      GO TO 7503
      ENDIF
      OPEN(63,FILE='M345ER')
      WRITE(63,'(A20)') CONVAL
      REWIND 63
      READ(63,*,ERR=2302) ZZMIN,ZINT,HEAVY,NCM
      IF(HEAVY.EQ.0.) GO TO 2303
      IF(HEAVY.LT.ZINT) THEN
      CALL CLEAR
      PRINT*,'HEAVY,LINE MULTIPLE MUST BE GREATER THAN INTERVAL'
      PRINT '(A)',' PRESS RETURN TO CONTINUE )'
```

```
      READ '(A1)',AASK
      GO TO 2302
      ENDIF
      GO TO 2303
 2302 CLOSE(43,STATUS='DELETE')
      GO TO 2301
 2303 CONTINUE
      CLOSE(43,STATUS='DELETE')
 7503 CONTINUE
      CALL CLEAR
      PRINT '(A\)',
     +' (2,17a) WANT TO SEE ELEVATIONS PLOTTED ? (Y/N) ) '
      READ '(A1)',HELV
      CALL CLEAR
      PRINT '(A\)',
     +' (2,17c),LEAVE X-Y AXES ANNOTATIONS OFF (Y/N) ? ) '
      READ '(A1)',AASK
      CALL CLEAR
      PRINT*
     +' (2,18) LEAVE,LOCATION,TIC''S AT DATA PTS. OFF TOO ? (Y/N)' ,
      PRINT '(A\)' ,' (ENTER ''A'' TO INCLUDE ''GRID'' POINTS ALSO) ) '
      READ '(A1)',PASK
      CALL CLEAR
      PRINT*,' (2,19) ENTER TITLE OF PLOT--(IF NONE, HIT RETURN)'
      PRINT '(A\)' ,'  -->'
      READ '(A48)',TITLE
      IF(PASK.NE.'Y') THEN
      CALL UCHPEN(LPEN(2))
      DO 1923 I=1,NP
      IF(PASK.NE.'A') THEN
      IF(NCC(I).EQ.0) GO TO 1923
      IF(NCC(I).LT.1000) GO TO 1923
      ENDIF
      IF(NCC(I).GE.1000) CALL USET(COL4)
      IF(NCC(I).LT.1000) CALL USET(COL2)
      CALL UMOVE(X(I),Y(I)-3.*FSOFT)
      CALL UPEN(X(I),Y(I)+3.*FSOFT)
      CALL UMOVE(X(I)-3.*FSOFT,Y(I))
      CALL UPEN(X(I)+3.*FSOFT,Y(I))
      IF(HELV.EQ.'Y') THEN
      CALL UMOVE(X(I)+6.*FSOFT,Y(I)+6.*FSOFT)
      CALL UPRNT1(Z(I),'REAL')
      ENDIF
 1923 CONTINUE
      ENDIF
      IF(IFLAG.EQ.1) THEN
      CALL UWINDO(1.,1040.,0.,780.)
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      ENDIF
      CALL USET(COL1)
      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.EQ.0.AND.HEAVY.NE.0.) NCM=52*HEAVY/ZINT
      IF(NCM.GT.500) NCM=500
      CL(1)=ZZMIN
      NC=2
  777 CL(NC)=CL(1)+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
      IF(NC-1.EQ.NCM) GO TO 778
      NC=NC+1
      GO TO 777
  778 NC=NC-1
      IF(NC.EQ.1.AND.CL(1).LT.ZMIN) NC=0
      IF(NC.EQ.1.AND.CL(1).GT.ZMAX) NC=0
      IF(HEAVY.NE.0.) THEN
      JJ=1
      DO 7608 I=1,NC
      NUMH(I)=0
      ZZ=CL(I)
      III=(ABS(ZZ)/HEAVY)+0.01
      HH=ABS(ZZ)-III*HEAVY
      IF(ABS(HH).LT.0.1*HEAVY) THEN
      KF=26
      IF(JJ.GT.26) KF=-26
      HLET(JJ)=LET(JJ+KF)
      NUMH(I)=JJ
      JJ=JJ+1
      ENDIF
 7608 CONTINUE
      DO 7607 I=1,NC
      IF(NUMH(I).NE.0) GO TO 159
```

```
7607 CONTINUE
     HEAVY=0.
     ENDIF
 159 CALL CLEAR
     PRINT '(A\)',' (2,15) ENTER TEMPLATE FILE NAME (DEF=NONE) ) '
     READ '(A20)',TMPLT
     IF(TMPLT.NE.' ') THEN
     CALL CLEAR
     CALL UCHPEN(LPEN(3))
     OPEN(12,FILE=TMPLT,STATUS='OLD',ERR=159)
     JPN=-1
  49 READ(12,*,END=59) X7,Y7,IPN
     X7=(X7-SU8X)*FACX
     Y7=(Y7-SUBY)*FACY
     X7=((X7-XMINX)/SM)+100.
     Y7=((Y7-YMINY)/SM)+100.
     CALL UWHERE(X77,Y77)
     ANG=ATAN2(Y7-Y77,X7-X77)
     IF(JPN.LT.0) CALL UMOVE(X7,Y7)
     IF(JPN.GE.0) CALL UPEN(X7,Y7)
     CALL UMOVE(X77+1.*FSOFT*SIN(ANG),Y77-1.*FSOFT*COS(ANG))
     IF(JPN.LT.0)CALL UMOVE(X7+1.*FSOFT*SIN(ANG),Y7-1.*FSOFT*COS(ANG))
     IF(JPN.GE.0)CALL UPEN(X7+1.*FSOFT*SIN(ANG),Y7-1.*FSOFT*COS(ANG))
     CALL UMOVE(X77-1.*FSOFT*SIN(ANG),Y77+1.*FSOFT*COS(ANG))
     IF(JPN.LT.0)CALL UMOVE(X7-1.*FSOFT*SIN(ANG),Y7+1.*FSOFT*COS(ANG))
     IF(JPN.GE.0)CALL UPEN(X7-1.*FSOFT*SIN(ANG),Y7+1.*FSOFT*COS(ANG))
     CALL UMOVE(X7,Y7)
     JPN=IPN
     GO TO 49
  59 CONTINUE
     IF(IFLAG.EQ.1) THEN
     CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
     CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
     ENDIF
     CALL UBELL
     CALL UPAUSE
     CLOSE(12)
     ENDIF
     CALL UCHPEN(LPEN(1))
     XSFT=PSFT
     IF(IW.EQ.1) XSFT=1.
     CALL USET('SOFT')
     CALL UPSET('VERT',30.*XSFT*FSOFT)
     CALL UPSET('HORI',15.*XSFT*FSOFT)
     IF(AASK.NE.'Y') CALL XYAXIS(XMINX,XMAXX,YMINY,YMAXY,SM,1)
     IF(MM.EQ.2.AND.IW.EQ.0) CALL USET('HARD')
     CALL USET(COL1)
     READ(81,*) NUMCON
     DO 7744 I=1,NUMCON
     READ(81,*) X1,Y1,X2,Y2
     CALL UMOVE(X1,Y1)
     CALL UPEN(X2,Y2)
7744 CONTINUE
     CALL CLEAR
     PRINT '(A\)',
    +' (2,14) LEAVE BOUNDARY OF CONTOURED REGION UNDRAWN ? (Y/N) ) '
     READ '(A1)',BASK
     CALL CLEAR
     READ(81,*) NUMCON
     DO 7745 I=1,NUMCON
     READ(81,*) X1,Y1,X2,Y2
     IF(BASK.NE.'Y') THEN
     CALL UMOVE(X1,Y1)
     CALL UPEN(X2,Y2)
     ENDIF
7745 CONTINUE
     CALL USET(COLOR)
1548 CONTINUE
     READ(81,*,END=1549) MN1,MN2,MN3,MN4,MN5,MN6
    +,ZT(1),ZT(2),ZT(3),ZT(4)
     DO 543 IC=1,NC
     KJ=IC
     ZZ=CL(IC)
     IF(ZZ.EQ.ZT(1)) ZT(1)=ZT(1)+0.01*ZINT
     IF(ZZ.EQ.ZT(2)) ZT(2)=ZT(2)-0.01*ZINT
     IF(ZZ.EQ.ZT(3)) ZT(3)=ZT(3)+0.01*ZINT
     IF(ZZ.EQ.ZT(4)) ZT(4)=ZT(4)-0.01*ZINT
     ZN=AMIN1(ZT(1),ZT(2))
     ZM=AMAX1(ZT(1),ZT(2))
     IF(ZZ.GE.ZN.AND.ZZ.LE.ZM)
    + CALL DRAWIT(MN1,MN2,ZZ,MN3,MN4,MN5,MN6,KJ)
 543 CONTINUE
     GO TO 1548
```

```
 1550 CONTINUE
 1549 CONTINUE
      CLOSE(81)
      CALL CLEAR
      PRINT,'(A)',' WANT TO DRAW THE TRIANGLE MESH ? (Y/N) ) '
      READ '(A1)',IDRAW
      IF(IDRAW.EQ.'Y') THEN
      CALL CLEAR
      PRINT,'(A)',' ERASE SCREEN ? (Y\N) ) '
      READ '(A1)',AASK
      IF(AASK.EQ.'Y') THEN
      CALL UERASE
      CALL UBACKG(CLRBCK)
      ENDIF
      DO 510 I=MBL,MB
      II=I
      CALL CORCOR(II)
      CALL UMOVE(CX(1),CY(1))
      CALL UPEN(CX(2),CY(2))
      CALL UPEN(CX(3),CY(3))
      CALL UPEN(CX(1),CY(1))
  510 CONTINUE
      ENDIF
C     .....DRAWING TITLE AND LEGEND.....
      IF(MM.EQ.1.OR.MM.EQ.2) THEN
      CALL UWINDO(1.,1040.,0.,780.)
      CALL UDAREA(10.*PCX,100.*PCX,0.*PCY,100.*PCY)
      CALL USET('SOFTWARE')
      CALL UPSET('VERT',30.*PSFT)
      CALL UPSET('HORI',15.*PSFT)
      IF(IW.EQ.1) CALL UDAREA(10.*PCX,100.*PCX,0.*PCY,100.*PCY)
      IF(IW.EQ.1) CALL UPSET('VERT',30.)
      IF(IW.EQ.1) CALL UPSET('HORI',15.)
      IF(MM.EQ.2.AND.IW.EQ.0) CALL USET('HARD')
      ENDIF
      IF(MM.EQ.3) THEN
      CALL UDAREA(1.,100.,0.,67.)
      CALL USET('SOFT')
      CALL UPSET('VERT',20.)
      CALL UPSET('HORI',15.)
      END IF
      CALL USET(COL4)
      CALL UMOVE(100.,40.)
      READ(TITLE,'(48A1)') (TITTLE(I),I=1,48)
      TITTLE(48)='
      WRITE(TITLE,'(48A1)') (TITTLE(I),I=1,48)
      CALL UPRNT1(TITLE,'TEXT')
      CALL CLEAR
      PRINT '(A)',
     +' (2.20) SKIP LEGEND ? (Y/N) ) '
      READ '(A1)',JASK
      CALL CLEAR
      IF(JASK.EQ.'Y') GO TO 162
      IPOW=NC
      KPOW=0
      DO 8900 I=1,NC
      IF(NUMH(I).GT.0.) KPOW=KPOW+1
 8900 CONTINUE
      IF(KPOW.GT.0) IPOW=KPOW
      CALL INF(IPOW)
      PX=775.
      YF=700.
      IF(IPOW.LT.26) YF=IPOW*25.+50.
      IF(YF.LT.500.) YF=500.
      CALL UMOVE(PX,YF+25.)
      CALL UPRNT1('INT=','TEXT')
      CALL UPRNT1(ZINT,'REAL')
      INC=1
      DO 500 IC=1,NC,INC
      IF(NUMH(IC).EQ.0.AND.HEAVY.NE.0.) GO TO 500
      IF(YF.LT.51.) THEN
      YF=YNG
      PX=XNG
      ENDIF
      ACL=CL(IC)
      CALL UMOVE(PX,YF)
      KF=-26
      IF(IC.LE.26) KF=26
      IF(HEAVY.EQ.0.) CALL UPRNT1(LET(IC+KF),'TEXT')
      IF(HEAVY.NE.0.) CALL UPRNT1(HLET(NUMH(IC)),'TEXT')
      CALL UPRNT1('=','TEXT')
      IF(ACL.GT.-0.0001.AND.ACL.LT.0.0001) ACL=0.
      IF(ABS(ACL).LT.100.) CALL UPRNT1(ACL,'REAL')
```

```
      IF(ABS(ACL).GE.100.) CALL UPRNTI(ACL,'INTE')
      IF(IC.EQ.1) THEN
      CALL UPRNTI('  ','TEXT')
      CALL UWHERE(XNG,YNG)
      ENDIF
      YF=YF-25.
  500 CONTINUE
  162 CONTINUE
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      CALL UPSET('VERT',.30.*PSFT*FSOFT)
      CALL UPSET('HORI',20.*PSFT*FSOFT)
      CALL UBELL
      CALL UPAUSE
      RETURN
      END
C
      SUBROUTINE CLOSEST(XX,YY)
      INTEGER VERTEX,SIDE
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      DMIN=99.E30
      J=0
      DO 100 I=1,NP
      IF(XX.GT.X(I)) GO TO 100
      IF(YY.GT.Y(I)) GO TO 100
      D=(XX-X(I))**2+(YY-Y(I))**2
      IF(D.LT.DMIN) THEN
      DMIN=D
      J=I
      ENDIF
  100 CONTINUE
      IF(J.NE.0) THEN
      XX=X(J)
      YY=Y(J)
      ENDIF
      RETURN
      END
C
      SUBROUTINE DRAWCON (IPCELL)
      DOUBLE PRECISION BX,BY,BZ
      INTEGER SIDE,VERTEX
      COMMON/TRI/MBL,MB
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/CONTOUR/NC,CL(500),IANS,FACT,PX,PY,PZT,CLS
      COMMON/JPOW/IPOW
      COMMON/BOUND/NCON,LCON(999),MED,MCON
      COMMON/HOUND/NUMCON,KCON(100)
      COMMON/CRAP/QDEL,MM
      COMMON/CELLS/NUM(1000),IRN(100),IRD(100),KRD,AVX,AVY
      COMMON/MINS/XMIN,XMAX,YMIN,YMAX
      COMMON/DRAWS/MBLNUM(100),NUMBR
      COMMON/NODEL/NODE(17000),NAV,NPMAX
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      COMMON/HVY/HEAVY,NUMH(500)
      COMMON/HHVY/HLET(52)
      DIMENSION NCCELL(1300),XTM(50),YTM(50),ZR(2),IAKSHN(50)
      DIMENSION MARK(999)
      CHARACTER HLET*2
      CHARACTER ABC*3,AASK*1
      CHARACTER DUM*72,ZOOM*1
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4
      CHARACTER COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER*2 LET(52),BLET,JASK*1,TITLE*48,TITTLE(48)*1,PASK*1
      CHARACTER COLR(7)*4,FILEN*12,FTMESH*12,SAVMSH*12
      CHARACTER FLNE*14,TYP*4,HONOR*2,BORD*2,IP*1,TMPLT*20,PQT*1
      CHARACTER CONVAL*40,CONVA(40)*1,FILEXY*12,FMME*14,FORCFILE*20
      CHARACTER ASPACE*1,ALABEL(999)*10,BSLSH*1,BLABEL*11,BLAB(11)*1
      DATA BSLSH/'\'/
      DATA XLAST/0./,YLAST/0./,ULAST/0./,VLAST/0./
      DATA LET/'a','b','c','d','e','f',
     +  'g','h','i','j','k','l','m','n',
     +  'o','p','q','r','s','t','u','v',
     +  'w','x','y','z','A','B','C','D',
     +  'E','F','G','H','I','J','K','L',
     +  'M','N','O','P','Q','R','S','T',
     +  'U','V','W','X','Y','Z'/
```

```
      DATA COLR/'YELL','RED ','GREE','BLUE','CYAN','MAGE','WHIT'/
    1 FORMAT(3F16.6,I12,A1,A10)
      OPEN(8,FILE='MCINFO.DAT')
      DO 434 I=1,50
  434 IAKSHN(I)=0
      DO 534 I=1,999
  534 MARK(I)=0
      DO 333 I=1,6
  333 READ(8,'(A1)') IP
      READ(8,*) FACX,FACY
      READ(8,*) SUBX,SUBY
      CLOSE(8)
      NPMAX=0
      JNPLT=1
  300 CONTINUE
      I=1
      ZMIN=99.E30
      ZMAX=-99.E30
      XMIN=99.E30
      XMAX=-99.E30
      YMIN=99.E30
      YMAX=-99.E30
      READ(9,'(A14)') FMME
      READ(9,*) FACX,FACY
      READ(9,*) SUBX,SUBY
  250 READ(9,1,END=200) X(I),Y(I),Z(I),NCC(I),ASPACE,ALABEL(I)
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      I=I+1
      IF(I.EQ.1000) THEN
      PRINT*,' --MORE THAN 999 NODES.. IGNORING EXTRA NODES..HIT RETURN'
      CALL UBELL
      CALL UPAUSE
      GO TO 200
      END IF
      GO TO 250
  200 NP=I-1
      IPCELL=NP/2+10
      IF(NP.LE.999) IPCELL=999
      IF(MM.EQ.2) IPCELL=999
      PRINT*
      PRINT*,'# NODES= ',NP,' .....PRESS RETURN'
      PRINT*
      CALL UBELL
      CALL UPAUSE
 7789 CONTINUE
      IF(2.EQ.2) GO TO 7779
      PRINT*,'XMIN= ',(XMIN/FACX)+SUBX,' XMAX= ',(XMAX/FACX)+SUBX
      PRINT*,'YMIN= ',(YMIN/FACY)+SUBY,' YMAX= ',(YMAX/FACY)+SUBY
      PRINT*,'(2.08) THE ABOVE ARE USED TO CHOOSE A SCALE TO FILL THE'
      PRINT*,'SCREEN. WANT TO CHANGE THESE VALUES ? (Y/N) '
      PRINT*,'OR ENTER NAME OF FILE GIVING XMIN,XMAX,YMIN,YMAX'
      PRINT '(A\)',' '
      READ '(A12)',FILEXY
      IF(FILEXY.EQ.'            ') GO TO 7779
      IF(FILEXY.EQ.'Y           ') THEN
      PRINT '(A\)',' ENTER NEW XMIN AND XMAX ) '
      READ*,XMIN,XMAX
      PRINT '(A\)',' ENTER NEW YMIN AND YMAX ) '
      READ*,YMIN,YMAX
      GO TO 7781
      ENDIF
      OPEN(44,FILE=FILEXY,STATUS='OLD',ERR=7789)
      READ(44,*) XMIN,XMAX,YMIN,YMAX
      CLOSE(44)
 7781 XMIN=(XMIN-SUBX)*FACX
      XMAX=(XMAX-SUBX)*FACX
      YMIN=(YMIN-SUBY)*FACY
      YMAX=(YMAX-SUBY)*FACY
 7779 CONTINUE
      SX=(XMAX-XMIN)/900.
      SY=(YMAX-YMIN)/640.
      SM=SX
      IF(SY.GT.SX) SM=SY
      DEV='IBMH'
      IF(JNPLT.EQ.1) CALL PSTART(1)
      IF(MM.EQ.3) DEV='EPS '
      IF(MM.EQ.2) DEV='HP4 '
      IF(MM.EQ.1) DEV='IBMH'
```

```
      IF(JNPLT.EQ.1) CALL UERASE
      CALL UBACKG(CLRBCK)
      XMINX=XMIN
      YMINY=YMIN
      XMAXX=XMAX
      YMAXY=YMAX
      DO 1778 I=1,NP
      X(I)=((X(I)-XMIN)/SM)+100.
      Y(I)=((Y(I)-YMIN)/SM)+100.
 1778 CONTINUE
      XQ1=100.
      YQ1=100.
      XQ2=((XMAX-XMIN)/SM)+100.
      YQ2=((YMAX-YMIN)/SM)+100.
      JTST=0
 1077 CONTINUE
      IF(JNPLT.EQ.1) THEN
      CALL CLEAR
      ENDIF
      IF(JTST.EQ.0) THEN
      JTST=1
      DO 1914 I=1,NP
      CALL UMOVE(X(I)-3.,Y(I))
      CALL UPEN(X(I)+3.,Y(I))
      CALL UMOVE(X(I),Y(I)-3.)
      CALL UPEN(X(I),Y(I)+3.)
 1914 CONTINUE
      CALL CLEAR
      NFALT=0
      PRINT*,'(2,09) DO YOU WANT TO DESCRIBE AN INTERNAL CONVEX AREA'
      PRINT,'(A)',' TO CONTAIN ALL CONTOURING ? (Y OR N) '
      READ(*,222) JASK
      IF(JASK.EQ.'Y') THEN
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL UOUTLN
      DO 1919 I=1,NP
      CALL UMOVE(X(I)-3.,Y(I))
      CALL UPEN(X(I)+3.,Y(I))
      CALL UMOVE(X(I),Y(I)-3.)
      CALL UPEN(X(I),Y(I)+3.)
 1919 CONTINUE
 8044 CALL CLEAR
      PRINT*
     +,'(2,10) ENTER # OF POINTS TO DESCRIBE AREA--(DEF=0) ) '
      PRINT,'(A)'
     +,' OR ENTER NAME OF A "FAULT" FILE ) '
      READ '(A40)',CONVAL
      IF(CONVAL.EQ.' ') JCON=0
      IF(CONVAL.NE.' ') THEN
      OPEN(43,FILE='M33343')
      WRITE(43,'(A40)') CONVAL
      REWIND 43
      READ(43,*,ERR=8046) JCON
      CLOSE(43,STATUS='DELETE')
      ENDIF
      GO TO 8047
 8046 JCON=-1
      CLOSE(43,STATUS='DELETE')
 8047 CONTINUE
      IF(JCON.EQ.1.OR.JCON.EQ.2) GO TO 8044
      IF(JCON.LT.0) THEN
      OPEN(44,FILE=CONVAL,STATUS='OLD',ERR=8044)
      OPEN(28,FILE='T4467')
      NFALT=1
      IL=1
   60 READ(44,*,END=61) XTM(IL),YTM(IL),IAKSHN(IL)
      IF(XTM(IL).EQ.0.0.AND.YTM(IL).EQ.0.0) GO TO 60
      XTM(IL)=(XTM(IL)-SUBX)*FACX
      YTM(IL)=(YTM(IL)-SUBY)*FACY
      XTM(IL)=((XTM(IL)-XMIN)/SM)+100.
      YTM(IL)=((YTM(IL)-YMIN)/SM)+100.
      IL=IL+1
      GO TO 60
   61 CLOSE(44)
      IL=IL-1
C.....SET 1ST AND LAST FAULT POINTS ELEV WITH ALL GIVEN POINTS
      DO 1102 I=1,2
      ST=0.
      SB=0.
      IF(I.EQ.1) THEN
      XR=XTM(1)
      YR=YTM(1)
```

```
      ENDIF
      IF(I.EQ.2) THEN
      IH=SQRT((XTM(IL-1)-XTM(IL))**2+(YTM(IL-1)-YTM(IL))**2)/20.
      IF(IH.EQ.0) IH=1
      XIH=IH
      XR=XTM(IL-1)+(IH-1)*(XTM(IL)-XTM(IL-1))/XIH
      YR=YTM(IL-1)+(IH-1)*(YTM(IL)-YTM(IL-1))/XIH
      ENDIF
      DO 1103 J=1,NP
      DD=(XR-X(J))**2+(YR-Y(J))**2
      IF(DD.EQ.0.) THEN
      ZR(I)=Z(J)
      GO TO 1102
      ENDIF
      ST=ST+Z(J)/DD
      SB=SB+1./DD
1103  CONTINUE
      ZR(I)=ST/SB
1102  CONTINUE
      OPEN(19,FILE='SKUNK')
      DO 520 I=1,NP
      DO 519 IK=1,IL-1
      CALL AREA(XTM(IK),XTM(IK+1),X(I),YTM(IK),YTM(IK+1),Y(I),A1)
      IF(A1.GE.0.0) GO TO 1521
 519  CONTINUE
      WRITE(19,*) X(I),Y(I),Z(I),NCC(I)
      GO TO 520
1521  WRITE(28,*) X(I),Y(I),Z(I)
 520  CONTINUE
      ENDFILE 19
      REWIND 19
      NP=1
1817  READ(19,*,END=1818) X(NP),Y(NP),Z(NP),NCC(NP)
      NP=NP+1
      GO TO 1817
1818  NP=NP-1
      NNPP=NP+1
      CALL USET(COL2)
      DO 1701 IK=1,IL-1
      IH=SQRT((XTM(IK)-XTM(IK+1))**2+(YTM(IK)-YTM(IK+1))**2)/20.
      IF(IH.EQ.0) IH=1
      XIH=IH
      DO 1702 I=1,IH
      NP=NP+1
      X(NP)=XTM(IK)+(I-1)*(XTM(IK+1)-XTM(IK))/XIH
      Y(NP)=YTM(IK)+(I-1)*(YTM(IK+1)-YTM(IK))/XIH
      CALL UMOVE(X(NP)-3.,Y(NP))
      CALL UPEN(X(NP)+3.,Y(NP))
      CALL UMOVE(X(NP),Y(NP)-3.)
      CALL UPEN(X(NP),Y(NP)+3.)
      SQ=(SQRT((XMAX-XMIN)**2+(YMAX-YMIN)**2)/20.)**2
      NCC(NP)=0
      ST=0.
      SB=0.
      DO 1706 J=1,NNPP-1
      DD=(X(NP)-X(J))**2+(Y(NP)-Y(J))**2
      IF(DD.EQ.0.) THEN
      Z(NP)=Z(J)
      GO TO 1704
      ENDIF
      ST=ST+Z(J)/DD
      SB=SB+1./DD
1706  CONTINUE
      IDO=0
      IF(IAKSHN(IK).EQ.0) IDO=1
      IF(IAKSHN(IK).EQ.1.AND.I.EQ.1) IDO=1
      IF(IAKSHN(IK).EQ.1.AND.IAKSHN(IK+1).EQ.0.AND.I.EQ.IH) IDO=1
      IF(IDO.EQ.1) THEN
      REWIND 28
6071  READ(28,*,END=6072) X8,Y8,Z8
      DD=(X(NP)-X8)**2+(Y(NP)-Y8)**2
      IF(DD.EQ.0.) THEN
      Z(NP)=Z8
      GO TO 1704
      ENDIF
      ST=ST+Z8/DD
      SB=SB+1./DD
      GO TO 6071
6072  CONTINUE
      ENDIF
      Z(NP)=ST/SB
1704  CONTINUE
1702  CONTINUE
```

```
1701 CONTINUE
     CALL USETICOLOR)
     X(NNPP1=ZR(1)
     Y(NP)=ZR(2)
     CALL UFLUSH
     CALL UBELL
     CALL UPAUSE
     CLOSE(28,STATUS='DELETE')
     GO TO 1523
     ENDIF
     CALL CLEAR
     PRINT*,'    (2,11)'
     PRINT*,'DESCRIBE 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  ,6,(,) KEYS GIVE FINE MOVEMENT).'
     CALL UMOVE(300.,300.)
     DO 986 I=1,JCON
     CALL UBELL
     CALL KURSIN(XP,YP,IP)
     DCL=999.E30
     DO 987 IC=1,NP
     IF(X(IC).GT.XP) THEN
     IF(Y(IC).GT.YP) THEN
     DCT=(XP-X(IC))**2+(YP-Y(IC))**2
     IF(DCT.LT.DCL) THEN
     ICL=IC
     DCL=DCT
     ENDIF
     ENDIF
     ENDIF
 987 CONTINUE
     LCON(I)=ICL
     CALL USETICOL1)
     CALL UMOVE(X(LCON(I))-9.,Y(LCON(I)))
     CALL UPEN(X(LCON(I))+9.,Y(LCON(I)))
     CALL UMOVE(X(LCON(I)),Y(LCON(I))-9.)
     CALL UPEN(X(LCON(I)),Y(LCON(I))+9.)
     CALL USETICOLOR)
     CALL UMOVE(XP,YP)
 986 CONTINUE
     LCON(JCON+1)=LCON(1)
     OPEN(19,FILE='SKUNK')
     DO 932 I=1,NP
     DO 930 J=1,JCON
     CALL DLAREA(X(LCON(J)),X(LCON(J+1)),X(I),Y(LCON(J)),Y(LCON(J+1)),
    +          Y(I),A)
     IF(A.LT.-0.0010) GO TO 931
 930 CONTINUE
     WRITE(19,*) X(I),Y(I),Z(I),NCC(I)
 931 CONTINUE
 932 CONTINUE
 523 CONTINUE
     REWIND 19
     NP=1
 252 READ(19,*,END=254) X(NP),Y(NP),Z(NP),NCC(NP)
     NP=NP+1
     GO TO 252
 254 CONTINUE
     NP=NP-1
1523 CONTINUE
     CLOSE(19,STATUS='DELETE')
     CALL UERASE
     CALL UBACKG(CLRBCK)
     GO TO 1077
     ENDIF
     ENDIF
     NCON=0
     NUMCON=0
     PRINT'(A\)',' (2,12) IS THERE A NON-CONTOURED AREA (Y/N)--? '
     READ(*,222) JASK
     CALL CLEAR
     IF(JASK.NE.'Y') PRINT*,'.....WAIT'
     IF(JASK.EQ.'Y') THEN
     CALL UERASE
     CALL UBACKG(CLRBCK)
     CALL UOUTLN
     DO 3344 I=1,NP
     CALL UMOVE(X(I)-3.,Y(I))
     CALL UPEN(X(I)+3.,Y(I))
     CALL UMOVE(X(I),Y(I)-3.)
     CALL UPEN(X(I),Y(I)+3.)
3344 CONTINUE
```

```
6601 CALL CLEAR
     PRINT*
    +'(2,10) ENTER # OF POINTS TO DESCRIBE AREA--(DEF=0) ) '
     PRINT '(A)'
    +' OR ENTER NAME OF A "FAULT" FILE ) '
     READ '(A40)',CONVAL
     IF(CONVAL.EQ.' ') NCON=0
     IF(CONVAL.NE.' ') THEN
     OPEN(43,FILE='M33344')
     WRITE(43,'(A40)') CONVAL
     REWIND 43
     READ(43,*,ERR=6603) NCON
     CLOSE(43,STATUS='DELETE')
     ENDIF
     GO TO 6605
6603 NCON=-1
     CLOSE(43,STATUS='DELETE')
6605 CONTINUE
     IF(NCON.EQ.1.OR.NCON.EQ.2) GO TO 6601
     IF(NCON.LT.0) THEN
     OPEN(44,FILE=CONVAL,STATUS='OLD',ERR=6601)
     NFALT=1
     IL=1
  70 READ(44,*,END=71) XTM(IL),YTM(IL),IAKSHN(IL)
     IF(XTM(IL).EQ.0.0.AND.YTM(IL).EQ.0.0) GO TO 70
     XTM(IL)=(XTM(IL)-SUBX)*FACX
     YTM(IL)=(YTM(IL)-SUBY)*FACY
     XTM(IL)=((XTM(IL)-XMIN)/SM)+100.
     YTM(IL)=((YTM(IL)-YMIN)/SM)+100.
     IL=IL+1
     GO TO 70
  71 CLOSE(44)
     IL=IL-1
     OPEN(19,FILE='SKUNK')
     NNPP=NP+1
     DO 1404 IK=1,IL-1
     IH=SQRT((XTM(IK)-XTM(IK+1))**2+(YTM(IK)-YTM(IK+1))**2)/20.
     IF(IH.EQ.0) IH=1
     XIH=IH
     CALL USET(COL2)
     DO 1405 I=1,IH
     NP=NP+1
     X(NP)=XTM(IK)+(I-1)*(XTM(IK+1)-XTM(IK))/XIH
     Y(NP)=YTM(IK)+(I-1)*(YTM(IK+1)-YTM(IK))/XIH
     NCC(NP)=0
     CALL UMOVE(X(NP)-3.,Y(NP))
     CALL UPEN(X(NP)+3.,Y(NP))
     CALL UMOVE(X(NP),Y(NP)-3.)
     CALL UPEN(X(NP),Y(NP)+3.)
     IF(IAKSHN(IK).EQ.1.AND.I.EQ.1) MARK(NP)=1
     IF(IAKSHN(IK).EQ.1.AND.IAKSHN(IK+1).EQ.0.AND.I.EQ.IH)MARK(NP)=1
1405 CONTINUE
     CALL USET(COLOR)
1404 CONTINUE
     CALL UFLUSH
     CALL UBELL
     CALL UPAUSE
     DO 521 I=1,NNPP-1
     DO 522 IK=1,IL-1
     CALL AREA(XTM(IK),XTM(IK+1),X(I),YTM(IK),YTM(IK+1),Y(I),A1)
     IF(A1.GT.0.00001) GO TO 521
 522 CONTINUE
     WRITE(19,*) I
 521 CONTINUE
     DO 734 I=NNPP,NP
 734 WRITE(19,*) I
     SQ=(SQRT((XMAX-XMIN)**2+(YMAX-YMIN)**2)/20.)**2
     DO 1407 I=NNPP,NP
     SB=0.0
     ST=0.0
     DO 1408 J=1,NNPP-1
     A2=0.0
C....USE ALL POINTS TO COMPUTE ELEV OF 1ST AND LAST FAULT POINTS..
     IF(I.EQ.NNPP.OR.I.EQ.NP) GO TO 1411
     IDO=0
     IF(IAKSHN(IK).EQ.0) IDO=1
     DO 6302 II=NNPP,NP
     IF(MARK(II).EQ.1) IDO=1
6302 CONTINUE
     IF(IDO.EQ.1) GO TO 1411
     DO 1409 IK=1,IL-1
     CALL DLAREA(XTM(IK),XTM(IK+1),X(J),YTM(IK),YTM(IK+1),Y(J),A1)
     IF(A1.GT.0.0) GO TO 1411
```

```
1409 CONTINUE
     GO TO 1405
1411 CONTINUE
     DD=(X(I)-X(J))**2+(Y(I)-Y(J))**2
     IF(DD.EQ.0.) THEN
     Z(I)=Z(J)
     GO TO 1407
     ENDIF
     ST=ST+Z(J)/DD
     SB=SB+1./DD
1408 CONTINUE
     Z(I)=ST/SB
1407 CONTINUE
     GO TO 958
     ENDIF
     NUMCON=NCON
     CALL CLEAR
     PRINT*,'(2,11)'
     PRINT*,'DESCRIBE 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 4,6,(,AND ) KEYS.
     PRINT*,'  ----THE  6,(,) KEYS GIVE FINE MOVEMENT.)'
     CALL UMOVE(300.,300.)
     DO 2086 I=1,NCON
     CALL UBELL
     CALL KURSIN(XP,YP,IP)
     DCL=999.E30
     DO 2087 IC=1,NP
     IF(X(IC).GT.XP) THEN
     IF(Y(IC).GT.YP) THEN
     DCT=(XP-X(IC))**2+(YP-Y(IC))**2
     IF(DCT.LT.DCL) THEN
     ICL=IC
     DCL=DCT
     ENDIF
     ENDIF
     ENDIF
2087 CONTINUE
     LCON(I)=ICL
     CALL USET(COL1)
     CALL UMOVE(X(LCON(I))-9.,Y(LCON(I)))
     CALL UPEN(X(LCON(I))+9.,Y(LCON(I)))
     CALL UMOVE(X(LCON(I)),Y(LCON(I))-9.)
     CALL UPEN(X(LCON(I)),Y(LCON(I))+9.)
     CALL USET(COLOR)
     CALL UMOVE(XP,YP)
2086 CONTINUE
     DO 8899 I=1,NUMCON
8899 KCON(I)=LCON(I)
     LCON(NCON+1)=LCON(1)
     OPEN(18,FILE='SKUN')
     REWIND 18
     DO 952 I=1,NP
     DO 954 J=1,NCON
     CALL DLAREA(X(LCON(J)),X(LCON(J+1)),X(I),Y(LCON(J)),Y(LCON(J+1)),
    +            Y(I),A)
     IF(A.LT.-0.001) GO TO 952
 954 CONTINUE
     WRITE(18,*) I
 952 CONTINUE
     ENDFILE 18
     REWIND 18
     OPEN(19,FILE='SKUNK')
     REWIND 19
 976 READ(18,*,END=958) I
     DO 957 J=1,NCON
     CALL DLAREA(X(LCON(J)),X(LCON(J+1)),X(I),Y(LCON(J)),Y(LCON(J+1)),
    +            Y(I),A)
     IF(ABS(A).LT.0.001) THEN
     WRITE(19,*) I
     GO TO 976
     ENDIF
 957 CONTINUE
     GO TO 976
 958 CONTINUE
     REWIND 19
     NCON=1
1952 READ(19,*,END=1954) I
     LCON(NCON)=I
     NCON=NCON+1
     GO TO 1952
1954 NCON=NCON-1
     REWIND 19
```

```
      CLOSE(11)
      END IF
      OPEN(11,FILE='TAP.DAT',STATUS='UNKNOWN')
      REWIND 11
      DO 779 I=1,NP
  779 WRITE(11,1) X(I),Y(I),Z(I),NCC(I),ASPACE,ALABEL(I)
      XPL=XWMIN
      XPR=XWMAX
      YPL=YWMIN
      YPR=YWMAX
      IF(JNPLT.EQ.1) THEN
      CALL CLEAR
      PRINT,'(A\)',' WANT TO (C)LIP OR (Z)OOM AREA (DEF=N) ?(C OR Z) )'
      READ '(A1)',ZOOM
      IF(ZOOM.EQ.'C'.OR.ZOOM.EQ.'Z') THEN
      CALL CLEAR
      PRINT,'(A\)',' WINDOW FROM (F)ILE OR (C)URSOR (DEF=C) ) '
      READ '(A1)',PQT
      IF(PQT.NE.'F') GO TO 8346
      CALL CLEAR
 9367 PRINT*,'ENTER,NAME OF FILE WHICH CONTAINS COORDINATES OF'
      PRINT '(A\)',' LOWER LEFT CORNER AND WIDTH OF WINDOW )'
      READ '(A20)',FORCFILE
      OPEN(43,FILE=FORCFILE,STATUS='OLD',ERR=9346)
      GO TO 9347
 9346 CALL CLEAR
      PRINT,'(A\)',' FILE ',FORCFILE,' NOT FOUND..PRESS RETURN ) '
      READ '(A1)',PQT
      CALL CLEAR
      GO TO 9367
 9347 CONTINUE
      READ(43,*) XPL,YPL,WIDT
      CLOSE(43)
      XPL=((XPL-SUBX-XMINX)/SM)+100.
      YPL=((YPL-SUBY-YMINY)/SM)+100.
      WIDT=WIDT/SM
      XPR=XPL+WIDT
      YPR=YPL
      GO TO 8345
 8346 CONTINUE
      PRINT*,'USE CURSOR TO LOCATE LOWER LEFT CORNER OF AREA'
      CALL UMOVE(400.,400.)
      CALL KURSIN(XPL,YPL,IP)
      CALL UMOVE(XPL-10.,YPL)
      CALL UPEN(XPL+10.,YPL)
      CALL UMOVE(XPL,YPL-10.)
      CALL UPEN(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(XPR-XPL).GT.0.001) YPR=YPL+(780./1039.)*(XPR-XPL)
      IF(ABS(YPR-YPL).GT.0.001) XPR=XPL+(1039./780.)*(YPR-YPL)
      XWMIN=XPL
      XWMAX=XPR
      YWMIN=YPL
      YWMAX=YPR
      ENDIF
      CALL UERASE
      CALL CLEAR
      CALL PEND
      END IF
 2301 CONTINUE
      ZMIN=9.E30
      ZMAX=-ZMIN
      DO 2307 I=1,NP
      IF(X(I).GT.XWMIN.AND.X(I).LT.XWMAX.AND.Y(I).GT.YWMIN.AND.
     +   Y(I).LT.YWMAX) THEN
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
      ENDIF
 2307 CONTINUE
      PRINT*,' (2.13)'
      PRINT*,'MIN EL.= ',ZMIN,' MAX EL.= ',ZMAX
      PRINT*,' ENTER BEGINNING CONTOUR, CONTOUR INTERVAL, HEAVY LINE'
      PRINT*,' MULTIPLE, AND NO. OF CONTOURS(0 MEANS SPAN RANGE).'
      PRINT,'(A)',
     +' OR ''N'' FOR NO CONTOURING (DEF=AUTO) (H FOR HELP) ) '
      READ '(A40)',CONVAL
      IF(CONVAL.EQ.'NONE'.OR.CONVAL.EQ.'N') CONVAL='999999,1.0,1'
      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 + OR -'
      PRINT '(A)','  CONTOUR INTERVAL.'
      PRINT*,' THE 3RD 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 (0 MEANS SPAN THE RANGE OF DATA'
      PRINT*,'I.E.  5020,-2,10,6 MEANS START AT ELEV 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,'(A)','..PRESS RETURN TO CONTINUE )'
      READ '(A1)',PQT
      GO TO 2301
      ENDIF
      IF(CONVAL.EQ.' ') THEN
      ZD=(ZMAX-ZMIN)/13.
      BLOG=LOG10(ZD)
      LOG=BLOG
      IF(BLOG.LT.0.0) LOG=LOG-1.
      IDEL=(ZD/(10.0**LOG))+1.
      JDEL=IDEL
      IF(IDEL.GT.5.) JDEL=10
      IF(IDEL.EQ.3) JDEL=4
      ZINT=JDEL*(10.0**LOG)
      HEAVY=ZINT*5
      IZ=(ZMIN/ZINT)+0.0001*ZMIN/ZINT
      ZZMIN=IZ*ZINT+ZINT
      NCM=0
      GO TO 7503
      ENDIF
      OPEN(43,FILE='M345ER')
      WRITE(43,'(A40)') CONVAL
      REWIND 43
      READ(43,*,ERR=2302) ZZMIN,ZINT,HEAVY,NCM
      IF(HEAVY.EQ.0.) GO TO 2303
      IF(HEAVY.LT.ZINT) THEN
      PRINT*,' HEAVY LINE MULTIPLE MUST BE GREATER ) OR = TO INTERVAL'
      GO TO 2302
      ENDIF
      GO TO 2303
2302  CLOSE(43,STATUS='DELETE')
      GO TO 2301
2303  CONTINUE
      CLOSE(43,STATUS='DELETE')
7503  CONTINUE
      IF(NFALT.EQ.1) MED=0
      IF(NFALT.EQ.1) GO TO 7564
      PRINT '(A)',
     +' (2.14) LEAVE BOUNDARY OF CONTOURED REGION UNDRAWN ? (Y/N) ) '
      READ(*,222) JASK
      MED=0
      IF(JASK.NE.'Y') MED=1
7564  PRINT,'(A)','(2.15) ENTER TEMPLATE FILE NAME (DEF=NONE) ) '
      READ '(A20)',TMPLT
      ITMPL=0
      IF(TMPLT.NE.' ') ITMPL=1
      PRINT '(A)',
     +' (2.17a) WANT TO SEE ELEVATIONS PLOTTED ? (Y/N) ) '
      READ(*,222) JASK
      JELV=0
      IF(JASK.EQ.'Y') JELV=1
      JLABEL=0
      IF(JELV.EQ.0) THEN
      JTEST=0
      DO 3200 I=1,NP
      IF(ALABEL(I).NE.'          ') JTEST=1
3200  CONTINUE
      IF(JTEST.EQ.1) THEN
      PRINT '(A)',
     +' (2.17b) WANT LABELS PLOTTED? (Y/N) ) '
      READ(*,222) JASK
      IF(JASK.EQ.'Y') JLABEL=1
      ENDIF
      ENDIF
      PRINT*,'(2.17c) LEAVE X-Y AXES ANNOTATIONS OFF ? (Y/N) ) '
      READ(*,222) AASK
      PRINT*,
     +'(2.18) LEAVE,LOCATION,IIC''S AT DATA PTS. OFF TOO ? (Y/N) ) '
      PRINT '(A)','  (ENTER ''A'' TO INCLUDE ''GRID'' POINTS ALSO) ) '
      READ(*,222) PASK
      JJELV=0
      IF(PASK.EQ.'Y') JJELV=1
      IF(JELV.EQ.1) JJELV=0
```

```
      PRINT*,'(<=19) ENTER TITLE OF PLOT--(IF NONE, HIT RETURN)'
      PRINT '(A)' '-->'
      READ(*,'(A48)') TITLE
      PRINT*,'                              --PRESS RETURN AT BEEPS'
      CALL UBELL
      CALL UPAUSE
      CALL PEND
      CALL PSTART(MM)
      IF(ZOOM.EQ.'C') CALL UWINDO(1.,1040.,0.,780.)
      CALL USET('COL2')
      CALL UOUTLN
      IF(ZOOM.EQ.'Z') CALL UWINDO(XPL,XPR,YPL,YPR)
      FSOFT=1.
      IF(ZOOM.EQ.'Z') FSOFT=((XPR-XPL)/1039.)
      IF(JJELV.EQ.0) THEN
      CALL UCHPEN(LPEN(2))
      DO 1929 I=1,NP
      IF(PASK.NE.'A') THEN
      IF(NCC(I).EQ.0) GO TO 1929
      IF(NCC(I).LT.1000) GO TO 1929
      ENDIF
      CALL UMOVE(X(I)-3.*FSOFT,Y(I))
      CALL UPEN(X(I)+3.*FSOFT,Y(I))
      CALL UMOVE(X(I),Y(I)-3.*FSOFT)
      CALL UPEN(X(I),Y(I)+3.*FSOFT)
1929  CONTINUE
      CALL UHOME
      CALL UBELL
      CALL UPAUSE
      ENDIF
      CALL USET('SOFT')
      CALL UPSET('VERT',30.*PSFT*FSOFT)
      CALL UPSET('HORI',15.*PSFT*FSOFT)
      IF(AASK.NE.'Y') CALL XYAXIS(XMINX,XMAXX,YMINY,YMAXY,SM,0)
      IF(JELV.EQ.1) THEN
      DO 1923 I=1,NP
      IF(PASK.NE.'A') THEN
      IF(NCC(I).EQ.0) GO TO 1923
      IF(NCC(I).LT.1000) GO TO 1923
      ENDIF
      CALL UMOVE(X(I)+6.*FSOFT,Y(I)+6.*FSOFT)
      CALL UPRNT1(Z(I),'REAL')
1923  CONTINUE
      CALL UBELL
      CALL UPAUSE
      END IF
      IF(JLABEL.EQ.1) THEN
      DO 1223 I=1,NP
      IF(NCC(I).EQ.0) GO TO 1223
      IF(NCC(I).LT.1000) GO TO 1223
      IF(ALABEL(I).EQ.'          ') GO TO 1223
      CALL UMOVE(X(I)+6.,Y(I)+6.)
      BLAB(11)='\'
      READ(ALABEL(I),'(10A1)') (BLAB(J),J=1,10)
      DO 1228 J=1,10,1
      IF(BLAB(J).NE.' ') THEN
      WRITE(ALABEL(I),'(10A1)') (BLAB(JJ),JJ=J,10)
      GO TO 1229
      ENDIF
1228  CONTINUE
1229  READ(ALABEL(I),'(10A1)') (BLAB(J),J=1,10)
      DO 1226 J=10,1,-1
      IF(BLAB(J).EQ.' ') GO TO 1226
      BLAB(J+1)='\'
      GO TO 1227
1226  CONTINUE
1227  WRITE(BLABEL,'(11A1)') (BLAB(J),J=3,11)
      IF(MM.EQ.2) THEN
      CALL UPSET('VERT',18.*PSFT*FSOFT)
      CALL UPSET('HORI',9.*PSFT*FSOFT)
      ENDIF
      CALL UPRNT1(BLABEL,'TEXT')
1223  CONTINUE
      IF(MM.EQ.2) THEN
      CALL UPSET('VERT',30.*PSFT*FSOFT)
      CALL UPSET('HORI',15.*PSFT*FSOFT)
      ENDIF
      CALL UBELL
      CALL UPAUSE
      ENDIF
      CALL USET(COLOR)
      IF(ITMPL.EQ.1) THEN
      CALL UCHPEN(LPEN(3))
```

```
      OPEN(12,FILE=TMPLT,STATUS='OLD',ERR=159)
      IPN=-1
   49 READ(12,*,END=59) X7,Y7,IPN
      X7=(X7-SUBX)*FACX
      Y7=(Y7-SUBY)*FACY
      X7=((X7-XMIN)/SM)+100.
      Y7=((Y7-YMIN)/SM)+100.
      CALL UWHERE(X77,Y77)
      ANG=ATAN2(Y7-Y77,X7-X77)
      IF(IPN.LT.0) CALL UMOVE(X7,Y7)
      IF(IPN.GE.0) CALL UPEN(X7,Y7)
      CALL UMOVE(X77+1.*FSOFT*SIN(ANG),Y77-1.*FSOFT*COS(ANG))
      IF(IPN.LT.0)CALL UMOVE(X7+1.*FSOFT*SIN(ANG),Y7-1.*FSOFT*COS(ANG))
      IF(IPN.GE.0)CALL UPEN(X7+1.*FSOFT*SIN(ANG),Y7-1.*FSOFT*COS(ANG))
      CALL UMOVE(X77-1.*FSOFT*SIN(ANG),Y77+1.*FSOFT*COS(ANG))
      IF(IPN.LT.0)CALL UMOVE(X7-1.*FSOFT*SIN(ANG),Y7+1.*FSOFT*COS(ANG))
      IF(IPN.GE.0)CALL UPEN(X7-1.*FSOFT*SIN(ANG),Y7+1.*FSOFT*COS(ANG))
      CALL UMOVE(X7,Y7)
      IPN=IPN
      GO TO 49
   59 CONTINUE
      CALL UBELL
      CALL UPAUSE
      CLOSE(12)
      END IF
  159 CONTINUE
      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.EQ.0.AND.HEAVY.NE.0.) NCM=52*HEAVY/ZINT
      LID=IPCELL
      CL(1)=ZZMIN
      NC=2
  777 CL(NC)=CL(1)+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
      IF(NC-1.EQ.NCM) GO TO 778
      NC=NC+1
      GO TO 777
  778 NC=NC-1
      IF(NC.EQ.1.AND.CL(1).LT.ZMIN) NC=0
      IF(NC.EQ.1.AND.CL(1).GT.ZMAX) NC=0
      IF(HEAVY.NE.0.) THEN
      JJ=1
      DO 7608 I=1,NC
      NUMH(I)=0
      ZZ=CL(I)
      III=(ABS(ZZ)/HEAVY)+0.01
      HH=ABS(ZZ)-III*HEAVY
      IF(ABS(HH).LT.0.1*HEAVY) THEN
      KF=26
      IF(JJ.GT.26) KF=-26
      HLET(JJ)=LET(JJ+KF)
      NUMH(I)=JJ
      JJ=JJ+1
      ENDIF
 7608 CONTINUE
      DO 7607 I=1,NC
      IF(NUMH(I).NE.0) GO TO 1159
 7607 CONTINUE
      HEAVY=0.
      ENDIF
 1159 CONTINUE
C     ..BUILD CONVEX CIRCULAR CELLS..ABOUT NPCELL POINTS PER CELL.
      IF(NCON.GT.0) THEN
      XMIN=99.E30
      XMAX=-99.E30
      YMIN=99.E30
      YMAX=-99.E30
      DO 693 I=1,NCON
      IF(X(LCON(I)).LT.XMIN) XMIN=X(LCON(I))
      IF(X(LCON(I)).GT.XMAX) XMAX=X(LCON(I))
      IF(Y(LCON(I)).LT.YMIN) YMIN=Y(LCON(I))
      IF(Y(LCON(I)).GT.YMAX) YMAX=Y(LCON(I))
  693 CONTINUE
      AVX=(XMIN+XMAX)/2.
      AVY=(YMIN+YMAX)/2.
      JID=0
      DO 661 I=1,NCON
      D=(X(LCON(I))-AVX)**2+(Y(LCON(I))-AVY)**2
      ID=SQRT(D)+1.0001
      IF(ID.GT.JID) JID=ID
  661 CONTINUE
```

```
      LID=0
      DO 662 I=1,NP
      D=(X(I)-AVX)**2+(Y(I)-AVY)**2
      ID=SQRT(D)+1.0001
      IF(ID.LE.JID) LID=LID+1
  662 CONTINUE
      END IF
      XMIN=99.E30
      XMAX=-99.E30
      YMIN=99.E30
      YMAX=-99.E30
      DO 694 I=1,NP
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
  694 CONTINUE
      IF(NCON.GT.0) GO TO 697
      AVX=(XMIN+XMAX)/2.
      AVY=(YMIN+YMAX)/2.
  697 CONTINUE
      DO 650 I=1,1300
  650 NCCELL(I)=0
      DO 660 I=1,NP
      D=(X(I)-AVX)**2+(Y(I)-AVY)**2
      ID=SQRT(D)+1.0001
  660 NCCELL(ID)=NCCELL(ID)+1
      K=2
      IRD(1)=0
      IRN(1)=0
      ND=0
      DO 670 I=1,1300
      ND=ND+NCCELL(I)
      IF(K.EQ.2) NPCELL=LID
      IF(K.NE.2) NPCELL=IPCELL
      IF(ND.GE.NPCELL) THEN
      IRD(K)=I
      IRN(K)=ND
      K=K+1
      ND=0
      END IF
  670 CONTINUE
      IF(ND.EQ.0) GO TO 680
      IRD(K)=1300
      IRN(K)=ND
      K=K+1
  680 CONTINUE
      KRD=K-1
      CALL UCHPEN(LPEN(1))
      CALL CLEAR
      PRINT*,'...WAIT...'
      IF(MM.EQ.2)CALL USET('HARD')
      IF(ZOOM.EQ.'C') THEN
      CALL UWINDO(XPL,XPR,YPL,YPR)
      XUDMIN=((XPL-1.)*(65./1039.)+10.)*PCX
      XUDMAX=((XPR-1.)*(65./1039.)+10.)*PCX
      YUDMIN=(YPL*(65./780.)+10.)*PCY
      YUDMAX=(YPR*(65./780.)+10.)*PCY
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      CALL UOUTLN
      ENDIF
      CALL CONTER(ZINT)
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
C   SAVE A MESH FILE IN FILE 'MESH.DAT'
      GO TO 8880
 8881 CONTINUE
      CALL CLEAR
      PRINT*,'(2,21) DO YOU WANT TO SAVE THIS MESH (SO THAT YOU MAY'
      PRINT,'(A)',' LATER DRAW OTHER PROFILES) ? (Y/N) '
      READ '(A1)',JASK
      IF(JASK.EQ.'Y') THEN
      CALL CLEAR
      PRINT*,'(2,22) ENTER A FILE NAME IN WHICH TO STORE THE MESH.'
      PRINT,'(A)','  (DEF=M.MSH)  '
      READ '(A12)',FIMESH
      IF(FIMESH.EQ.'            ') FIMESH='M.MSH'
      IF(FIMESH.EQ.'            ') SAVMSH='M.:::'
      CALL CLEAR
      PRINT*,'      WAIT'
      OPEN(8,FILE=FIMESH)
      WRITE(81,*) SUBX,SUBY
      WRITE(81,*) FACX,FACY
```

```
      WRITE(81,*) XQ1,XQ2,YQ1,YQ2
      WRITE(81,*) ZMAX,ZMIN,XMINX,XMAXX
      WRITE(81,*) YMINY,YMAXY,SM
      WRITE(81,*) NP,MB,MBL
      DO 3356 I=1,NP
      WRITE(81,*) X(I),Y(I),Z(I),NCC(I)
 3356 CONTINUE
      DO 3367 I=1,MB
      WRITE(81,*) (VERTEX(I,J),J=1,3)
 3367 CONTINUE
      OPEN(65,FILE='EDGECON',FORM='FORMATTED')
      READ(65,*,END=1868) NMU
      WRITE(81,*) NMU
 1867 READ(65,*,END=1868) X1,Y1,X2,Y2
      WRITE(81,*) X1,Y1,X2,Y2
      GO TO 1867
 1868 CLOSE(65,STATUS='DELETE')
      OPEN(65,FILE='OUTEDGE')
      READ(65,'(A3)',ERR=1823) ABC
      IF(ABC.EQ.' AB') THEN
      WRITE(81,*) 0
      CLOSE(65,STATUS='DELETE')
      GO TO 1824
      ENDIF
 1823 REWIND 65
      IKN=0
 1827 READ(65,'(A72)',END=1828) DUM
      IKN=IKN+1
      GO TO 1827
 1828 WRITE(81,*) IKN
      REWIND 65
 1842 READ(65,*,END=1843) X1,Y1,X2,Y2
      WRITE(81,*) X1,Y1,X2,Y2
      GO TO 1842
 1843 CLOSE(65,STATUS='DELETE')
 1824 CONTINUE
      OPEN(85,FILE='SAVCON')
      REWIND 85
 1767 READ(85,*,END=1768)MN1,MN2,MN3,MN4,MN5,MN6
     +,X1,Y1,X2,Y2
      WRITE(81,*) MN1,MN2,MN3,MN4,MN5,MN6
     +,X1,Y1,X2,Y2
      GO TO 1767
 1768 CONTINUE
      CLOSE(85)
      CLOSE(81)
      CALL CLEAR
      ENDIF
      GO TO 8882
C     .....DRAWING TITLE AND LEGEND.....
 8880 CONTINUE
      IF(MM.EQ.1.OR.MM.EQ.2) THEN
      CALL UWINDO(1.,1040.,0.,780.)
      CALL UDAREA(10.*PCX,100.*PCX,0.*PCY,100.*PCY)
      CALL USET('SOFTWARE')
      CALL UPSET('VERT',30.*PSFT)
      CALL UPSET('HORI',15.*PSFT)
      IF(MM.EQ.2) CALL USET('HARD')
      END IF
      IF(MM.EQ.3) THEN
      CALL UDAREA(1.,100.,0.,67.)
      CALL USET('SOFT')
      CALL UPSET('VERT',20.)
      CALL UPSET('HORI',15.)
      END IF
      CALL USET(COL4)
      CALL UMOVE(100.,40.)
      READ(TITLE,'(48A1)') (TITTLE(I),I=1,48)
      TITTLE(48)='$'
      WRITE(TITLE,'(48A1)') (TITTLE(I),I=1,48)
      CALL UPRNTI(TITLE,'TEXT')
      CALL CLEAR
      PRINT '(A)',
     +' (2.20) SKIP LEGEND ? (Y/N) ) '
      READ '(A1)',JASK
      CALL CLEAR
      IF(JASK.EQ.'Y') GO TO 162
      IPOW=NC
      KPOW=0
      DO 8900 I=1,NC
      IF(NUMH(I).GT.0) KPOW=KPOW+1
 8900 CONTINUE
      IF(KPOW.GT.0) IPOW=KPOW
```

```
      CALL INF(IPOW)
      PX=775.
      YF=700.
      IF(IPOW.LT.26) YF=IPOW*25.+50.
      IF(YF.LT.500.) YF=500.
      CALL UMOVE(PX,YF+25.)
      CALL UPRNT1('INT=','TEXT')
      CALL UPRNT1(ZINT,'REAL')
      INC=1
      DO 500 IC=1,NC,INC
      IF(NUMH(IC).EQ.0.AND.HEAVY.NE.0.) GO TO 500
      IF(YF.LT.51.) THEN
      YF=YNG
      PX=XNG
      ENDIF
      ACL=CL(IC)
      CALL UMOVE(PX,YF)
      KF=-26
      IF(IC.LE.26) KF=26
      IF(HEAVY.EQ.0.) CALL UPRNT1(LET(IC+KF),'TEXT')
      IF(HEAVY.NE.0.) CALL UPRNT1(HLET(NUMH(IC)),'TEXT')
      CALL UPRNT1(' ','TEXT')
      IF(ACL.GT.-0.0001.AND.ACL.LT.0.0001) ACL=0.
      IF(ABS(ACL).LT.100.) CALL UPRNT1(ACL,'REAL')
      IF(ABS(ACL).GE.100.) CALL UPRNT1(ACL,'INTE')
      IF(IC.EQ.1) THEN
      CALL UPRNT1(' ','TEXT')
      CALL UWHERE(XNG,YNG)
      ENDIF
      YF=YF-25.
  500 CONTINUE
      CALL UBELL
      CALL UPAUSE
  162 CONTINUE
      GO TO 8881
 8882 CONTINUE
C.....CREATE PROFILES.....
      IF(MM.EQ.1) DEV='I8MM'
      IF(MM.EQ.2) DEV='AP4'
      CALL PEND
      CALL PSTART(MM)
      CALL USET(COLOR)
      CALL UFLUSH
      CALL CLEAR
C.....DRAW PROFILES.....
      CALL USET(COLOR)
      CALL UFLUSH
  956 CONTINUE
      REWIND 09
      XBI=MB
      CALL CLEAR,
      PRINT '(A\)',
     +' (2,32) WANT TO GENERATE A TRIPLE TRIANGLE MESH ? (Y/N) )'
      READ(*,222) JASK
      IF(JASK.EQ.'Y') THEN
      CALL CLEAR,
      PRINT '(A\)',
     +' (2,33) LINEAR OR INVERSE SQUARE RULE ? (L OR I) DEF=L ) '
      READ(*,222) JASK
      NPWT=1
      IF(JASK.EQ.'I') NPWT=0
      PRINT*,'NEW INPUT DATA FILE NAME IS "TRIPLE"'
      PRINT*,'.....WAIT....'
      DO 516 I=MBL,MB
      II=I
      CALL CORCOR(II)
      CALL DLAREA(CX(1),CX(2),CX(3),CY(1),CY(2),CY(3),A)
      IF(ABS(A).LT.15.) GO TO 516
      VX=(CX(1)+CX(2)+CX(3))/3.
      VY=(CY(1)+CY(2)+CY(3))/3.
      S1=(VX-CX(1))**2+(VY-CY(1))**2
      S2=(VX-CX(2))**2+(VY-CY(2))**2
      S3=(VX-CX(3))**2+(VY-CY(3))**2
      VZ=(CZ(1)/S1+CZ(2)/S2+CZ(3)/S3)/(1./S1+1./S2+1./S3)
      IF(NPWT.EQ.1) CALL CONSTS(VX,VY,VZ),
      WRITE(11,*) VX,VY,VZ,0,
  516 CONTINUE
      ENDFILE 11
      REWIND 11
      CLOSE(09)
      OPEN(13,FILE='TRIPLE')
      WRITE(13,'(A14\)')                    \'
      WRITE(13,*) FACX,FACY
```

```
      WRITE(13,*) SUBX,SUBY
  817 READ(11,1,END=818) BX,BY,BZ,IZZ,ASPACE,ALABEL(1)
      BX=SM*(BX-100.)+XMINX
      BY=SM*(BY-100.)+YMINY
      WRITE(13,1) BX,BY,BZ,IZZ,ASPACE,ALABEL(1)
      GO TO 817
  818 CONTINUE
      ENDFILE 13
      CLOSE(13)
      ENDIF
      CLOSE(11)
      CLOSE(09)
      CALL CLEAR
      PRINT*,MB,' TRIANGLES FORMED  '
      PRINT '(A)',
     +'(2,34) DO YOU WANT TO DRAW THE MESH OF TRIANGLES--? (Y/N) )'
      READ(*,222) JASK
  222 FORMAT(A1)
      CALL CLEAR
      IF(JASK.EQ.'Y') THEN
      CALL UMOVE(20.,560.)
      IF(MM.EQ.1) CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      IF(MM.EQ.2) CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      IF(MM.EQ.3) CALL UDAREA(10.,75.,10.,50.)
      IF(MM.EQ.3) CALL UPSET('VERT',30.)
      IF(MM.EQ.3) CALL UPSET('HORI',24.)
C  ...DRAW TRIANGLES
      IF(MM.EQ.1.OR.MM.EQ.2) CALL USET('HARD')
      ICLR=1
      DO 510 I=MBL,MB
      II=I
      XBI=II
      CALL CORCOR(II)
      CALL UMOVE(CX(1),CY(1))
      CALL UPEN(CX(2),CY(2))
      CALL UPEN(CX(3),CY(3))
      CALL UPEN(CX(1),CY(1))
      ICLR=ICLR+1
      IF(ICLR.GT.7) ICLR=1
  510 CONTINUE
      CALL PUAIN
  513 CONTINUE
      CALL UFLUSH
      CALL UHOME
      CALL UBELL
      CALL UPAUSE
      ENDIF
      RETURN
      END
C
      SUBROUTINE XYAXIS(XMIN,XMAX,YMIN,YMAX,SM,IFLAG)
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      COMMON/DEVICE/DEV,PCX,PCY,PSET,COLOR,CLRBCK,PSIZE,LPEN(8)
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      IF(IFLAG.EQ.1) THEN
      IF(ABS(XUDMIN-10.*PCX).GT.0.001.AND.ABS(XUDMAX-75.*PCX)
     +       .GT.0.001) THEN
      CALL UWINDO(1.,1040.,0.,780.)
      CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      ENDIF
      CALL UOUTLN
      ENDIF
      XX=0.
      XXMAX=((XMAX-XMIN)/SM)+100.
      NUM=XXMAX/125.
      ZD=(XMAX-XMIN)/NUM
      BLOG=LOG10(ZD)
      LOG=BLOG
      IF(BLOG.LT.0.0) LOG=LOG-1
      IDEL=(ZD/(10.**LOG))+1
      JDEL=IDEL
      IF(IDEL.GT.5) JDEL=10
      IF(IDEL.EQ.3) JDEL=4
      XINT=JDEL*(10.**LOG)
      XS=XINT/SM
      IX=(XMIN/XINT)+0.0001*XMIN/XINT
      ZZMIN=IX*XINT
      XX=ZZMIN-XINT
   20 XX=XX+XINT
      X=((XX-XMIN)/SM)+100.
      IF(X.LT.50.) GO TO 20
```

```
      IF(X.GE.XXMAX+XS) GO TO 30
      CALL UMOVE(X,0.)
      CALL UPEN(X,10.)
      CALL UMOVE(X,34.)
      CALL USET('ACENTER')
      IF(XINT/FACX.LT.1.) CALL UPRNT1((XX/FACX)+SUBX,'REAL')
      IF(XINT/FACX.GE.1.) CALL UPRNT1((XX/FACX)+SUBX,'INTE')
      CALL USET('NOCENTER')
      CALL UMOVE(X,770.)
      CALL UPEN(X,780.)
      GO TO 20
   30 CONTINUE
      YYMAX=((YMAX-YMIN)/SM)+100.
      NUM=YYMAX/125
      ZD=(YMAX-YMIN)/NUM
      BLOG=LOG10(ZD)
      LOG=BLOG
      IF(BLOG.LT.0.0) LOG=LOG-1
      IDEL=(ZD/(10.**LOG))+1
      JDEL=IDEL
      IF(IDEL.GT.5) JDEL=10
      IF(IDEL.EQ.3) JDEL=4
      YINT=JDEL*(10.**LOG)
      YS=YINT/SM
      IY=(YMIN/YINT)+0.0001*YMIN/YINT
      ZZMIN=IY*YINT
      YY=ZZMIN-YINT
   40 YY=YY+YINT
      Y=((YY-YMIN)/SM)+100.
      IF(Y.LT.50.) GO TO 40
      IF(Y.GE.YYMAX+YS) GO TO 50
      CALL UMOVE(0.,Y)
      CALL UPEN(10.,Y)
      IF(YINT/FACY.GE.1.) CALL UPRNT1((YY/FACY)+SUBY,'INTE')
      IF(YINT/FACY.LT.1.) CALL UPRNT1((YY/FACY)+SUBY,'REAL')
      CALL UMOVE(1030.,Y)
      CALL UPEN(1040.,Y)
      GO TO 40
   50 CONTINUE
      IF(IFLAG.EQ.1) THEN
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      ENDIF
      RETURN
      END
C
      SUBROUTINE DRWPRF(MM)
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      CHARACTER COLOR*4,CLRBCK*4,PSIZE
      CHARACTER DESC*70,SEC*1,SL*1,SEC2*2,DESC2*71,YLABEL*40,YLAB2*41
      CHARACTER DPR*72,IASK*1,REFEL*40,OREFEL*40,NEWLAB*40
      CHARACTER YYQ*2,ISHADE*1,LEVOFF*1,ZEFEL*40,OZEFEL*40
      CHARACTER OSHADE*1,OEVOFF*1,JKASK*1,SYM*1,LEG*20,GEL*21,LR*1
      CHARACTER YL(40)*1
      DIMENSION X(260),Y(260),Z(260),IBOX(80,80)
      DATA SL/'\'/
      CALL UCHPEN(LPEN(1))
      IF(MM.EQ.1.OR.MM.EQ.2) THEN
      CALL USET('SOFT')
      CALL UPSET('VERT',30.*PSFT)
      CALL UPSET('HORI',20.*PSFT)
      ENDIF
      IF(MM.EQ.2) THEN
      CALL USET('SOFT')
      ENDIF
      REWIND 14
      READ(14,*) VX1,VY1
      READ(14,*) VX2,VY2
      READ(14,'(A70)') DESC
      READ(14,'(A1)') SEC
      READ(14,'(A40)') YLABEL
      WRITE(DESC2,'(A70,A1)') DESC,SL
      WRITE(SEC2,'(2A1)') SEC,SL
      WRITE(YLAB2,'(A40,A1)') YLABEL,SL
      ZMAX=-9.E09
      ZMIN=-ZMAX
      YMIN=ZMIN
      XMIN=ZMIN
      YMAX=ZMAX
      XMAX=ZMAX
      I=1
  141 READ(14,*,END=143) X(I),Y(I),Z(I)
      X(I)=X(I)-100.
```

```
      Y(I)=Y(I)-100.
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IZZ=Z(I)-0.02
      IF(IZZ.EQ.-999) THEN
      I=I+1
      GO TO 141
      ENDIF
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
      I=I+1
      GO TO 141
  143 CONTINUE
      NJ=I-1
      CLOSE(14)
      XSUB=X(1)
      YSUB=Y(1)
      CALL CLEAR
      PRINT,'(A\)',' FIRST PROFILE OF A SERIES OF PLOTS ? Y/N ) '
      READ '(A1)',JKASK
      IF(JKASK.EQ.'Y') THEN
      DO 101 I=1,80
      DO 101 J=1,80
  101 IBOX(I,J)=0
      ENDIF
      IF(JKASK.EQ.'Y') THEN
      CALL CLEAR
      PRINT*,'Vertical axis description= ',YLABEL
      PRINT,'(A\)',' Enter new description (DEF=NO CHANGE) ) '
      READ '(A40)',NEWLAB
      IF(NEWLAB.NE.' ') YLABEL=NEWLAB
      CALL CLEAR
      ENDIF
      ISYM=0
      IF(JKASK.EQ.'Y') MSYM=0
      ISYM=MSYM
      IF(JKASK.NE.'Y') THEN
      MSYM=MSYM+4
      ISYM=MSYM
      ENDIF
      IF(MSYM.GE.16) MSYM=0
      CALL CLEAR
      IF(JKASK.EQ.'Y') THEN
      PRINT '(A\)',
     +' (2.37) WANT THE PROFILE STRETCHED TO FILL PAGE ? Y/N ) '
      READ '(A1)',JASK
      ENDIF
      CALL CLEAR
      DO 200 I=1,NJ
      X(I)=(X(I)-XSUB)*0.639
  200 Y(I)=(Y(I)-YSUB)*0.639
      XF=1.0
      IF(JASK.EQ.'Y') XF=664.56/SQRT((X(NJ)-X(1))**2+(Y(NJ)-Y(1))**2)
      DO 201 I=1,NJ
      X(I)=X(I)*XF
  201 Y(I)=Y(I)*XF
      CALL CLEAR
      PRINT*,DESC
      PRINT*,'SECTION ',SEC
      PRINT 333,ZMIN,ZMAX
  333 FORMAT(9H MIN EL= ,F8.2,11H   MAX EL= ,F8.2)
 6777 CONTINUE
      IF(JKASK.NE.'Y') REFEL='S'
      IF(JKASK.EQ.'Y') THEN
      PRINT '(A\)',
     +' (2.38) ENTER ZMIN AND ZMAX (DEF=AUTO) ) '
      READ '(A40)',REFEL
      ENDIF
      XSET=200.
      YSET=100.
      YGAP=125.
      CALL CLEAR
      IF(REFEL.EQ.'S') REFEL=OREFEL
      OREFEL=REFEL
      IF(REFEL.EQ.' ') THEN
      ZD=(ZMAX-ZMIN)
      BLOG=LOG10(ZD)
      LOG=BLOG
      IF(BLOG.LT.0.0) LOG=LOG-1
      IDEL=(ZD/(10.0**LOG))+1.
      JDEL=IDEL
```

```
      IF(IDEL.GE.6) IDEL=10
      IF(IDEL.EQ.3) IDEL=4
      ZINT=IDEL*(10.0**LOG)
      IZ=(ZMIN/ZINT)+0.0001*ZMIN/ZINT
      ZZMIN=IZ*ZINT
      ZMIN=ZZMIN
      ZZMAX=ZMIN
   38 ZZMAX=ZZMAX+ZINT
      IF(ZZMAX.GT.ZMAX) GO TO 89
      GO TO 88
   89 ZMAX=ZZMAX
      ZPLOT=(ZMAX-ZMIN)/5.
      GO TO 94
      ENDIF
      IF(REFEL.NE.' ') THEN
      OPEN(43,FILE='M44334')
      WRITE(43,'(A40)') REFEL
      REWIND 43
      READ(43,*,ERR=6777) ZMIN,ZMAX
      CLOSE(43,STATUS='DELETE')
      ENDIF
  418 CONTINUE
      IF(JKASK.NE.'Y') ZEFEL='S'
      IF(JKASK.EQ.'Y') THEN
      PRINT '(A\)',
     +' (2.39) HOW MANY Z UNITS PER PLOT INCH (DEF=AUTO) ) '
      READ '(A40)',ZEFEL
      ENDIF
      CALL CLEAR
      IF(ZEFEL.EQ.'S') ZEFEL=OZEFEL
      OZEFEL=ZEFEL
      IF(ZEFEL.EQ.' ') THEN
      ZPLOT=(ZMAX-ZMIN)/5.
      GO TO 94
      ENDIF
      IF(ZEFEL.NE.' ') THEN
      OPEN(43,FILE='M44343')
      WRITE(43,'(A40)') ZEFEL
      REWIND 43
      READ(43,*,ERR=418) ZPLOT
      CLOSE(43,STATUS='DELETE')
      ENDIF
   94 CONTINUE
      IF((ZMAX-ZMIN)/ZPLOT.GT.5.05) THEN
      CALL CLEAR
      PRINT*,'USING ',ZPLOT,' WILL PLOT TOO TALL. TRY A LARGER VALUE.'
      GO TO 418
      ENDIF
      CALL CLEAR
      IF(JKASK.NE.'Y') ISHADE='S'
      IF(JKASK.EQ.'Y') THEN
      PRINT '(A\)',' WANT PROFILE SHADING REMOVED Y/N ?(S FOR SAME) )'
      READ '(A1)',ISHADE
      ENDIF
      IF(ISHADE.EQ.'S') ISHADE=OSHADE
      OSHADE=ISHADE
      CALL CLEAR
      IF(JKASK.NE.'Y') LEVOFF='Y'
      IF(JKASK.EQ.'Y') THEN
      PRINT '(A\)',' AXIS LABELS, ETC LEFT OFF Y/N ?  ) '
      READ '(A1)',LEVOFF
      ENDIF
      IF(LEVOFF.EQ.'S') LEVOFF=OEVOFF
      OEVOFF=LEVOFF
      CALL CLEAR
 8855 PRINT*,'SYMBOL ON PROFILE LINE (DEF=NONE, S=SQUARE, C=CIRCLE'
      PRINT '(A\)',' T=TRIANGLE, D=DIAMOND. *=* ) '
      READ '(A1)',SYM
      CALL CLEAR
      IF(SYM.EQ.' '.OR.SYM.EQ.'S'.OR.SYM.EQ.'C'.OR.SYM.EQ.'T'.
     +   OR.SYM.EQ.'D'.OR.SYM.EQ.'*') GO TO 8866
      CALL CLEAR
      PRINT*,'  ',SYM,' NOT A VALID SYMBOL....TRY AGAIN'
      GO TO 8855
 8866 CONTINUE
      IF(SYM.NE.' ') THEN
      CALL CLEAR
      PRINT '(A\)',' LEGEND FOR SYMBOL ? (20 CHARS MAX) ) '
      READ '(A20)',LEG
      WRITE(GEL,'(A20,A1)') LEG,SL
      IF(JKASK.EQ.'Y') THEN
      CALL CLEAR
      PRINT '(A\)',' LEGEND PRINTED ON LEFT (DEF) OR RIGHT ? L/R ) '
```

```
      READ '(A1)',LR
      IF(LR.EQ.' ') LR='L'
      ENDIF
      ENDIF
      CALL CLEAR
      ZDIF=ZPLOT/YGAP
      IF(ZDIF.EQ.0.) RETURN
      IF(JKASK.EQ.'Y') ILEG=0
      IF(JKASK.EQ.'Y') CALL UERASE
      CALL CLEAR
      IF(JKASK.EQ.'Y') CALL UBACKG(CLRBCK)
      IF(JKASK.EQ.'Y') CALL UOUTLN
      IF(ISHADE.EQ.'Y') GO TO 412
      DO 400 I=1,NJ
      IZZ=Z(I)-0.02
      IF(IZZ.EQ.-999) GO TO 400
      ZZ=(Z(I)-ZMIN)/ZDIF
      EX=SQRT(X(I)**2+Y(I)**2)+XSET
      IF(ZMIN.LT.0.) THEN
      IF(I-(I/2)*2.EQ.0) THEN
      CALL UMOVE(EX,(-ZMIN/ZDIF)+YSET)
      CALL UPEN(EX,ZZ+YSET)
      ENDIF
      IF(I-(I/2)*2.NE.0) THEN
      CALL UMOVE(EX,ZZ+YSET)
      CALL UPEN(EX,(-ZMIN/ZDIF)+YSET)
      ENDIF
      ENDIF
      IF(ZMIN.GE.0.) THEN
      IF(I-(I/2)*2.EQ.0) THEN
      CALL UMOVE(EX,YSET)
      CALL UPEN(EX,ZZ+YSET)
      ENDIF
      IF(I-(I/2)*2.NE.0) THEN
      CALL UMOVE(EX,ZZ+YSET)
      CALL UPEN(EX,YSET)
      ENDIF
      ENDIF
  400 CONTINUE
  412 CONTINUE
      IB=0
      DO 402 I=2,NJ
      EX2=SQRT(X(I)**2+Y(I)**2)+XSET
      EX1=SQRT(X(I-1)**2+Y(I-1)**2)+XSET
      ZZ2=(Z(I)-ZMIN)/ZDIF
      ZZ1=(Z(I-1)-ZMIN)/ZDIF
      IZ1=Z(I)-0.02
      IZ2=Z(I-1)-0.02
      IF(I.EQ.2) THEN
      G1LAST=EX1
      G2LAST=ZZ1+YSET
      ENDIF
      IF(IZ1.EQ.-999.OR.IZ2.EQ.-999) GO TO 402
      ISYM=ISYM+1
      IF(2.EQ.2) THEN
      G1=(EX1+EX2)/2.
      G2=(ZZ1+ZZ2+YSET+YSET)/2.
      CALL XYMBOL(G1,G2,SYM)
      IF(SQRT((G1LAST-G1)**2+(G2LAST-G2)**2).GT.25.) THEN
      CALL XYMBOL((G1+G1LAST)/2.,(G2+G2LAST)/2.,SYM)
      ENDIF
      G1LAST=G1
      G2LAST=G2
      IB=IB-MSYM*5
      ISYM=0
      ENDIF
      IF(ISHADE.NE.'Y') THEN
      CALL UMOVE(EX1,ZZ1+YSET)
      CALL UPEN(EX2,ZZ2+YSET)
      ENDIF
  402 CONTINUE
      IF(LEVOFF.NE.'Y') THEN
      CALL UCHPEN(LPEN(4))
      SQ=SQRT((X(NJ))**2+(Y(NJ))**2)
      SH=SQ
      IF(JASK.NE.'Y') SH=SQRT((X(NJ)-X(1))**2+(Y(NJ)-Y(1))**2)
      CALL UMOVE(100.,YSET)
      CALL UPRNTI(ZMIN,'REAL')
      CALL UMOVE(XSET-10.,YSET)
      CALL UPEN(XSET,YSET)
      CALL UMOVE(XSET,YSET)
      CALL UPEN(SQ+XSET,YSET)
      ZS=0.0
```

```
      Z1=ZMIN
500   ZS=ZS+YGAP
      Z1=Z1+ZPLOT
      CALL UMOVE(100.,ZS+YSET)
      CALL UPRNT1(Z1,'REAL')
      CALL UMOVE(XSET-10.,ZS+YSET)
      CALL UPEN(XSET,ZS+YSET)
      CALL UMOVE(XSET,ZS-YGAP+YSET)
      CALL UPEN(XSET,ZS+YSET)
      CALL UWHERE(R1,R2)
      CALL UMOVE(XSET,ZS+YSET)
      CALL UPEN(SQ+XSET,ZS+YSET)
      CALL UWHERE(R3,R4)
      CALL UMOVE(SQ+XSET,ZS-YGAP+YSET)
      CALL UPEN(SQ+XSET,ZS+YSET)
      IF(Z1.GE.ZMAX) GO TO 510
      GO TO 500
510   CONTINUE
      CALL UFLUSH
      CALL UMOVE(R1+60.,R2+20.)
      DPR=DESC2
      CALL UPRNT1(DPR,'TEXT')
      DPR=SEC2
      CALL UMOVE(SQRT(X(1)**2+Y(1)**2)+XSET,YSET)
      CALL UWHERE(G1,G2)
      CALL UMOVE(G1+10.,R2-30.)
      CALL UPRNT1(DPR,'TEXT')
      CALL UMOVE(SQRT(X(NJ)**2+Y(NJ)**2)+XSET,YSET)
      CALL UWHERE(G1,G2)
      CALL UMOVE(G1-35.,R4-30.)
      CALL UPRNT1(DPR,'TEXT')
      IF(SEC.NE.' ') CALL UPRNT1('*\','TEXT')
      CALL UMOVE(XSET+SQ/2.,-100.,YSET-70.)
      CALL UPRNT1('L= \','TEXT')
      DI=SQRT((VX1-VX2)**2+(VY1-VY2)**2)
      IF(JASK.NE.'Y') THEN
      DI=DI*SQ/SH
      ENDIF
      IF(DI.GT.100.) CALL UPRNT1(DI+.05,'INTE')
      IF(DI.LE.100.) CALL UPRNT1(DI,'REAL')
      DO 600 I=1,11
      G3=(I-1)/10.
      CALL UMOVE(XSET+(I-1)*SQ/10.,YSET)
      CALL UPEN(XSET+(I-1)*SQ/10.,YSET-10.)
      CALL UWHERE (G1,G2)
      IF(MM.EQ.1) CALL UMOVE(G1-35.,G2-30.)
      IF(MM.EQ.2) CALL UMOVE(G1-15.,G2-30.)
      IF(I.GE.2.AND.I.LE.10) THEN
      CALL UPRNT1(G3,'REAL')
      CALL UPRNT1('L\','TEXT')
      ENDIF
      IF(I.EQ.1) CALL UPRNT1('.0\','TEXT')
      IF(I.EQ.11) CALL UPRNT1(' L\','TEXT')
600   CONTINUE
      RX=R3+20.
      RY=R4-30.
      READ(YLABEL,'(40A1)') (YL(I),I=1,40)
      DO 850 I=40,1,-1
      IF(YL(I).NE.' ') GO TO 851
850   CONTINUE
      I=40
851   CONTINUE
      DY=(RY-YSET+10.)/I
      IF(DY.GT.30.) DY=30.
      CALL USET('SOFT')
      CALL UPSET('VERT',0.75*DY*PSFT)
      CALL UPSET('HORI',0.50*DY*PSFT)
      J=I
      DO 700 I=1,J
      WRITE(YYQ,'(2A1)') YL(I),SL
      IF(RY.LT.YSET) GO TO 700
      CALL UMOVE(RX,RY)
      CALL UPRNT1(YYQ,'TEXT')
      RY=RY-DY
700   CONTINUE
      ENDIF
      CALL UPSET('VERT',.30.*PSFT)
      CALL UPSET('HORI',.20.*PSFT)
      IF(SYM.NE.' ') THEN
      IF(LEG.NE.'                    ') THEN
      YLEG=R4-30.*(ILEG+1)
      ILEG=ILEG+1
      XXSS=XSET
```

```
      IF(LR.EQ.'R') XXSS=XXSS+400.
      CALL XYMBOL(XXSS+50.,YLEG+10.,SYM)
      CALL UMOVET(XXSS+75.,YLEG)
      CALL UPRNT1(GEL,'TEXT')
      ENDIF
      ENDIF
      CALL UMOVE(0.,0.)
      CALL UFLUSH
      CALL USETI('SOFT')
      RETURN
      END
C
      SUBROUTINE FENFIL(ZMX,ZMN,MM,X1,X2,Y1,Y2,PX,RDASK,ISKIP,XMINX,
     +                  XMAXX,YMINY,YMAXY,SM,IVC,FTYPE)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/TRI/MBL,MB
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      DIMENSION XT(260),YT(260),ZT(260)
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER DEV*4,COLOR*4,CLRBCK*4,PSIZE*1
      CHARACTER IP*1,JASK*1,FILEN*12,DESC*80,SLASH*1,IPP*2,YLABEL*40
      CHARACTER RDASK*1,FTYPE*1,ABC*2
      DATA SLASH/'\'/
      IIPEN=LPEN(2)
      IF(ISKIP.EQ.0) GO TO 1900
      IPP=' ,'
      IP=' '
      CALL CLEAR
      PRINT*,'...WAIT..'
      IF(2.NE.2) THEN
      EX=X1
      EY=Y1
      TX=X2
      TY=Y2
      IIPEN=LPEN(2)
      IF(ABS(X2-X1).GE.0.1) THEN
      IIPEN=LPEN(4)
      IF(X2.LT.X1) THEN
      X2=EX
      Y2=EY
      X1=TX
      Y1=TY
      GO TO 123
      ENDIF
      ENDIF
      IF(ABS(X2-X1).LT.0.1.AND.Y2.LT.Y1) THEN
      X2=TX
      Y2=EY
      X1=TX
      Y1=TY
      ENDIF
  123 CONTINUE
      ENDIF
      XB1=X1
      YB1=Y1
      XE1=X2
      YE1=Y2
      IF(2.NE.2) THEN
      DO 333 I=1,1041,2
      IF(XB1.LE.0.0.AND.XE1.LE.0.0) GO TO 335
      IF((X2-X1).NE.0.0) F=(I-1)/(X2-X1)
      IF((X2-X1).EQ.0.0) F=(I-1)/(Y2-Y1)
      IF(XB1.GT.0.) THEN
      XB1=X1+F*(X2-X1)
      YB1=Y1+F*(Y2-Y1)
      ENDIF
      IF(XE1.GT.0.0) THEN
      XE1=X2-F*(X2-X1)
      YE1=Y2-F*(Y2-Y1)
      ENDIF
      DO 334 J=1,MB
      CALL CORCOR(J)
      IF(XB1.GT.0.) THEN
      CALL AREA(CX(1),CX(2),XB1,CY(1),CY(2),YB1,A1)
      CALL AREA(CX(2),CX(3),XB1,CY(2),CY(3),YB1,A2)
      CALL AREA(CX(3),CX(1),XB1,CY(3),CY(1),YB1,A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      XB1=-XB1
```

```
      YB1=-YB1
      ENDIF
      ENDIF
      IF(XE1.GT.0.) THEN
      CALL AREA(CX(1),CX(2),XE1,CY(1),CY(2),YE1,A1)
      CALL AREA(CX(2),CX(3),XE1,CY(2),CY(3),YE1,A2)
      CALL AREA(CX(3),CX(1),XE1,CY(3),CY(1),YE1,A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      XE1=-XE1
      YE1=-YE1
      ENDIF
      ENDIF
  334 CONTINUE
  333 CONTINUE
  335 CONTINUE
      X1=-XB1
      Y1=-YB1
      X2=-XE1
      Y2=-YE1
      ENDIF
      CALL USET(COL1)
      IF(MM.EQ.1) CALL UCRCLE(X1,Y1,5.)
      IF(MM.EQ.1) CALL UCRCLE(X2,Y2,5.)
      CALL UFLUSH
      NJ=(X2-X1)/5.
      IF(NJ.EQ.0) NJ=(Y2-Y1)/5.
      NJ=101
      DO 100 I=1,NJ
      OI=I
      ONJ=NJ
      F=(OI-1)/(ONJ-1)
      XT(I)=X1+F*(X2-X1)
  100 YT(I)=Y1+F*(Y2-Y1)
      ZMIN=99.E30
      ZMAX=-ZMIN
      II=1
      JFND=0
      LFND=0
      LBB=1
      LBJ=101
      LBC=100
      LTIME=0
  315 CONTINUE
      DO 210 J=LBB,LBJ,LBC
      ZT(J)=-999
      IF(LFND.EQ.1) GO TO 210
C     FIND TRIANGLE POINT IS IN.....
      CALL UALPHA
      DO 200 IG=II,MB+II-1
      KK=0
      IF(IG.GT.MB) KK=MB
      I=IG-KK
      IF(I.LT.MBL) GO TO 200
      CALL CORCOR(I)
      CALL AREA(CX(1),CX(2),XT(J),CY(1),CY(2),YT(J),A1)
      CALL AREA(CX(2),CX(3),XT(J),CY(2),CY(3),YT(J),A2)
      CALL AREA(CX(3),CX(1),XT(J),CY(3),CY(1),YT(J),A3)
      IF(A1.LE.0.AND.A2.LE.0.AND.A3.LE.0.) THEN
      CALL UALPHA
      IF(J.EQ.LBB.AND.LTIME.EQ.1) JFND=1
C     FOUND IT
      CALL CONSTS(XT(J),YT(J),ZT(J))
      IF(ZT(J).GT.ZMAX) ZMAX=ZT(J)
      IF(ZT(J).LT.ZMIN) ZMIN=ZT(J)
      II=MAX(1,I-(MB/10))
      GO TO 210
      ENDIF
  200 CONTINUE
      IF(JFND.EQ.1) LFND=1
  210 CONTINUE
      IF(ZT(1).LE.-998.0.AND.ZT(101).LE.-998.0) LFND=1
      LBB=1
      LBJ=NJ
      LBC=1
      IF(ZT(1).LE.-998.0.AND.ZT(101).GT.-998.0) THEN
      LBB=NJ
      LBJ=1
      LBC=-1
      ENDIF
      LTIME=LTIME+1
      IF(LTIME.EQ.1) GO TO 315
   88 CALL CLEAR
      IF(ISKIP.EQ.1) THEN
```

```
      DO 603 I=1,NJ
  603 WRITE(23,*) XT(I),YT(I),ZT(I)
      ENDIF
      JASK='D'
      ZZMIN=ZMN
      ZZMAX=ZMX
 7902 CONTINUE
 1900 CONTINUE
      IF(MM.EQ.2) THEN
      CALL PEND
      DEV='HP4'
      CALL PSTART(2)
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL UCHPEN(IIPEN)
      IF(IIPEN.EQ.0) CALL UCHPEN(LPEN(1))
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      ENDIF
      IF(RDASK.EQ.'Y') THEN
      CALL USET(COLOR)
      CALL UCHPEN(LPEN(1))
      CALL RDCON(ZMX,ZMN,0,XMINX,XMAXX,YMINY,YMAXY,SM)
      CALL UCHPEN(IIPEN)
      RDASK='N'
      CALL USET(COL1)
      ENDIF
      CLOSE(81)
      IF(ISKIP.EQ.0) RETURN
      XVC=IVC-1
      IF(MM.EQ.2) THEN
      CALL USET('SOFT')
      CALL UPSET('VERT',24.*PSFT*FSOFT)
      CALL UPSET('HORI',12.*PSFT*FSOFT)
      ENDIF
      IF(FTYPE.EQ.'P') THEN
      ANG=ATAN2(Y2-Y1,X2-X1)
      CALL UMOVE(X1-10.*COS(ANG),Y1-10.*SIN(ANG))
      CALL UPEN(X1-30.*COS(ANG),Y1-30.*SIN(ANG))
      CALL UMOVE(X1-60.*COS(ANG),Y1-60.*SIN(ANG))
      CALL UWHERE(XS,YS)
      IF(XVC.GT.9.5) CALL UMOVE(XS-20.*PSFT,YS)
      CALL UPRNT1(XVC,'INTE')
      ABC='\'
      CALL UPRNT1(ABC,'TEXT')
      ENDIF
      IF(FTYPE.EQ.'N') THEN
      ANG=ATAN2(Y2-Y1,X2-X1)
      CALL UMOVE(X2+10.*COS(ANG),Y2+10.*SIN(ANG))
      CALL UPEN(X2+30.*COS(ANG),Y2+30.*SIN(ANG))
      CALL UMOVE(X2+50.*COS(ANG),Y2+50.*SIN(ANG))
      CALL UWHERE(XS,YS)
      IF(XVC.GT.9.5) CALL UMOVE(XS-20.*PSFT,YS)
      CALL UPRNT1(XVC,'INTE')
      ABC='\'
      CALL UPRNT1(ABC,'TEXT')
      ENDIF
      CALL UMOVE(X1,Y1)
      CALL UPEN(X2,Y2)
      CALL UHOME
      CALL UFLUSH
C     DRAW THE PROFILE
      IF(PX.GT.0.01) THEN
      ZDIF=(ZZMAX-ZZMIN)/PX
      IF(ZDIF.EQ.0.) RETURN
      XP=0.
      THE=ATAN2(Y2-Y1,X2-X1)
      CALL USET(COL3)
      DO 300 I=2,NJ
      IF(ZT(I).EQ.-999.) GO TO 300
      IF(ZT(I-1).EQ.-999.) GO TO 300
      IF(ZT(I).GT.ZZMAX) ZT(I)=ZZMAX
      IF(ZT(I).LT.ZZMIN) ZT(I)=ZZMIN
      ZZ=(ZT(I)-ZZMIN)/ZDIF
      CALL UMOVE(XT(I),YT(I))
      CALL UPEN(XT(I)-ZZ*SIN(THE),YT(I)+ZZ*COS(THE))
      CALL UPEN(XT(I-1)-ZZ*SIN(THE),YT(I-1)+ZZ*COS(THE))
      CALL UPEN(XT(I-1),YT(I-1))
      CALL UPEN(XT(I),YT(I))
      CALL UMOVE(XT(I),YT(I))
  300 CONTINUE
      ENDIF
      CALL UHOME
      CALL UFLUSH
```

```
      CALL USET(COLOR)
      RETURN
      END
C
      SUBROUTINE FENCE(ZMAX,ZMIN,MM,XMINX,XMAXX,YMINY,YMAXY,SM,IW,KW)
      INTEGER SIDE,VERTEX
      COMMON/COOR/NCC(999),X(999),Y(999),Z(999),CX(3),CY(3),CZ(3),
     +            VERTEX(2000,3),SIDE(2000,3),NP
      COMMON/DVICE/DEV,PCX,PCY,PSFT,COLOR,CLRBCK,PSIZE,LPEN(8)
      COMMON/TRI/MBL,MB
      COMMON/KOLOR/COL1,COL2,COL3,COL4
      COMMON/STUFF/ZIT(4)
      COMMON/SCALES/FACX,FACY,SUBX,SUBY
      COMMON/WINDO/XWMIN,XWMAX,YWMIN,YWMAX
      COMMON/DAREA/XUDMIN,XUDMAX,YUDMIN,YUDMAX,FSOFT
      CHARACTER COL1*4,COL2*4,COL3*4,COL4*4
      CHARACTER COLOR*1,CLRBCK*1,PSIZE*1,DEV*4
      CHARACTER JASK*1,FIMESH*12,FTYPE*1,HIMESH*12,PASK*1,RDASK*1
      CHARACTER EXAG*10,REFEL*10,F(12)*1,E(12)*1,OLDMSH*12,PIF*60
      CHARACTER EXTEN(25)*3,FNME*12,PROFI*22,GWEL*22,FROOT*8,PORN*1
      CHARACTER NEWZS*40,ZOOM*1,IP*1,FORC*1,FORCFILE*20,POT*1
      DATA EXTEN/'  0','  1','  2','  3','  4','  5','  6','  7',
     +     '  8','  9',' 10',' 11',' 12',' 13',' 14',' 15',' 16',
     +     ' 17',' 18',' 19',' 20',' 21',' 22',' 23',' 24'/
      DATA OLDMSH/'M.MSH        '/
      DATA HIMESH/'             '/
      CALL CLEAR
      IF(KW.EQ.0),THEN
      PRINT '(A1)','     (2.41) WANT TO DRAW GRID BOX PROFILES ? (Y/N) ) '
      READ '(A1)',JASK
      ENDIF
      IF(KW.EQ.1) JASK='Y'
      ZOOM='
      XUDMIN=10.*PCX
      XUDMAX=75.*PCX
      YUDMIN=10.*PCY
      YUDMAX=75.*PCY
      XWMIN=1.
      XWMAX=1040.
      YWMIN=0.
      YWMAX=780.
      FSOFT=1.
 1456 CONTINUE
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      IF(JASK.EQ.'Y') THEN
      IF(IW.EQ.0) THEN
  102 PRINT*,'(2.40) ENTER NAME OF "MESH" FILE (DEF= ',OLDMSH,')'
      PRINT '(A1)','     OR  "Q"  TO QUIT )'
      READ '(A12)',FIMESH
      IF(FIMESH.EQ.'Q') RETURN
      IF(FIMESH.EQ.' ') FIMESH=OLDMSH
      OPEN(81,FILE=FIMESH,STATUS='OLD',ERR=100)
      OLDMSH=FIMESH
      GO TO 103
  100 PRINT*,'FILE NOT FOUND,  TRY AGAIN'
      GO TO 102
  103 READ(81,*) SUBX,SUBY
      READ(81,*) FACX,FACY
      READ(81,*) XMIN,XMAX,YMIN,YMAX
      READ(81,*) ZMAX,ZMIN,XMINX,XMAXX
      READ(81,*) YMINY,YMAXY,SM
      READ(81,*) NP,MB,MBL
      QXMIN=9.E30
      QYMIN=QXMIN
      QXMAX=-QXMIN
      QYMAX=QXMAX
      DO 105 I=1,NP
      READ(81,*) X(I),Y(I),Z(I),NCC(I)
      IF(X(I).LT.QXMIN) QXMIN=X(I)
      IF(X(I).GT.QXMAX) QXMAX=X(I)
      IF(Y(I).LT.QYMIN) QYMIN=Y(I)
      IF(Y(I).GT.QYMAX) QYMAX=Y(I)
  105 CONTINUE
      DO 106 I=1,MB
  106 READ(81,*) (VERTEX(I,J),J=1,3)
      ENDIF
      CALL CLEAR
      IF(KW.EQ.0),THEN
      PRINT '(A1)',
     +  '     (2.24a) WANT TO REDRAW THE CONTOURS ? (Y/N) ) '
      READ '(A1)',RDASK
      ENDIF
```

```
      IF(KW.EQ.1) RDASK='Y'
      CALL CLEAR
      PRINT '(A\)',
     +' (2.24b) WANT THE VOLUME UNDER THE MESH ? (Y/N) ) '
      READ '(A1)',PASK
      IF(PASK.EQ.'Y') THEN
      CALL CLEAR
      PRINT*,'MINIMUM ELEVATION= ',ZMIN
 4455 PRINT '(A\)',
     +' (2.24c) ENTER REFERENCE ''BASE'' ELEVATION (DEF=0.0) ) '
      READ '(A10)',REFEL
      IF(REFEL.EQ.' ') REFELE=0.0
      IF(REFEL.NE.' ') THEN
      OPEN(43,FILE='M44334')
      WRITE(43,'(A10)') REFEL
      REWIND 43
      READ(43,*,ERR=4455) REFELE
      CLOSE(43,STATUS='DELETE')
      ENDIF
      ATOT=0.0
      VOL=0.0
      VOL1=0.0
      VOL2=0.0
      DO 540 I=MBL,MB
      CALL CORCOR(I)
      CALL AREA(CX(1),CX(2),CX(3),CY(1),CY(2),CY(3),A)
      A=SM*SM*ABS(A)/2.
      ATOT=ATOT+A
      ZAV=(CZ(1)+CZ(2)+CZ(3))/3.
      VOL=VOL+A*(ZAV-REFELE)
      VOL1=VOL1+A*(ZAV-ZMIN)
      VOL2=VOL2+A*(ZAV-ZMAX)
  540 CONTINUE
      REF=(-VOL1*(ZMAX-ZMIN)/(VOL2-VOL1))+ZMIN
      ENDIF
      XDF=XMAX-XMIN
      YDF=YMAX-YMIN
      XMIN=XMIN+0.1*XDF
      YMIN=YMIN+0.1*YDF
      XMAX=XMAX-0.1*XDF
      YMAX=YMAX-0.1*YDF
      OANGG=0.
      OXBX=(XMIN-100.)*SM+SUBX+XMINX
      OYBX=(YMIN-100.)*SM+SUBY+YMINY
      OXLEN=(XMAX-XMIN)*SM
      OXHT=(YMAX-YMIN)*SM
      IF(FIMESH.NE.HIMESH) THEN
      CALL UERASE
      CALL UBACKG(CLRBCK)
      HIMESH=FIMESH
      ENDIF
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      IF(MM.EQ.2) THEN
      CALL PEND
      DEV='I8MM'
      CALL PSTART(1)
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL USET(COL2)
      CALL UOUTLN
      ENDIF
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
C     DRAW POINTS
      GO TO 1939
 1938 CONTINUE
      CALL UERASE
      CALL UBACKG(CLRBCK)
      CALL UOUTLN
 1939 CONTINUE
      CALL USET(COL2)
      DO 1931 I=1,NP
      CALL UMOVE(X(I)-3.*FSOFT,Y(I))
      CALL UPEN(X(I)+3.*FSOFT,Y(I))
      CALL UMOVE(X(I),Y(I)-3.*FSOFT)
      CALL UPEN(X(I),Y(I)+3.*FSOFT)
 1931 CONTINUE
      CALL USET(COLOR)
      XPL=XWMIN
      XPR=XWMAX
      YPL=YWMIN
      YPR=YWMAX
      CALL CLEAR
      IF(ZOOM.EQ.' ') THEN
```

```
      PRINT,'(A\)',' WANT TO (C)LIP OR (Z)OOM AREA (DEF=N) ?(C OR Z) )'
      READ '(A1)',ZOOM
      IF(ZOOM.EQ.'C'.OR.ZOOM.EQ.'Z') THEN
      CALL CLEAR
      PRINT,'(A\)',' WINDOW FROM (F)ILE OR (C)URSOR (DEF=C) ? '
      READ '(A1)',FORC
      IF(FORC.NE.'F') GO TO 8346
      CALL CLEAR
9367  PRINT*,'ENTER NAME OF FILE WHICH CONTAINS COORDINATES OF'
      PRINT,'(A)','  LOWER LEFT CORNER AND WIDTH OF WINDOW )'
      READ '(A20)',FORCFILE
      OPEN(43,FILE=FORCFILE,STATUS='OLD',ERR=9346)
      GO TO 9347
9346  CALL CLEAR
      PRINT,'(A)',' FILE',FORCFILE,' NOT FOUND ...PRESS RETURN ) '
      READ '(A1)',PQT
      CALL CLEAR
      GO TO 9367
9347  CONTINUE
      READ(43,*) XPL,YPL,WIDT
      CLOSE(43)
      XPL=((XPL-SUBX-XMINX)/SM)+100.
      YPL=((YPL-SUBY-YMINY)/SM)+100.
      WIDT=WIDT/SM
      XPR=XPL+WIDT
      YPR=YPL
      GO TO 8345
8346  CONTINUE
      CALL CLEAR
      PRINT*,'USE CURSOR TO LOCATE LOWER LEFT CORNER OF AREA'
      CALL UMOVE(400.,400.)
      CALL KURSIN(XPL,YPL,IP)
      CALL UMOVE(XPL-10.,YPL)
      CALL UPEN(XPL+10.,YPL)
      CALL UMOVE(XPL,YPL-10.)
      CALL UPEN(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(XPR-XPL).GT.0.001) YPR=YPL+(780./1039.)*(XPR-XPL)
      IF(ABS(YPR-YPL).GT.0.001) XPR=XPL+(1039./780.)*(YPR-YPL)
      XWMIN=XPL
      XWMAX=XPR
      YWMIN=YPL
      YWMAX=YPR
      ENDIF
      IF(ZOOM.EQ.'C') THEN
      CALL UWINDO(XPL,XPR,YPL,YPR)
      XUDMIN=((XPL-1.)*(65./1039.)+10.)*PCX
      XUDMAX=((XPR-1.)*(65./1039.)+10.)*PCX
      YUDMIN=(YPL*(65./780.)+10.)*PCY
      YUDMAX=(YPR*(65./780.)+10.)*PCY
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      GO TO 1938
      ENDIF
      IF(ZOOM.EQ.'Z') THEN
      CALL UWINDO(XPL,XPR,YPL,YPR)
      FSOFT=(XPR-XPL)/1039.
      GO TO 1938
      ENDIF
      ENDIF
      CALL UPSET('VERT',30.*FSOFT*PSFT)
      CALL UPSET('HORI',20.*FSOFT*PSFT)
      IF(IJASK.EQ.'Y') THEN
      IF(KW.EQ.1) THEN
      CALL FENFIL(ZMAX,ZMIN,MM,X1,X2,Y1,Y2,EX,RDASK,0,XMINX,XMAXX,
     + YMINY,YMAXY,SM,0,FTYPE)
      CLOSE(81)
      GO TO 79
      ENDIF
8869  CALL CLEAR
      XBX=OXBX
      YBX=OYBX
      XLEN=OXLEN
      XHT=OXHT
      ANGG=OANGG
      PRINT*,'(2.41a)'
      PRINT*,'ENTER PROFILE GRID BOX ORIGIN OF COORDINATES, ANGLE OF'
      PRINT*,'BOX TO HORZ, LENGTH OF BOX, & HEIGHT. (X,Y,ANG,L,H)'
      PRINT*,'OR ENTER  ''S''  FOR SAME AS LAST TIME (DEF=AUTO)'
      READ '(A60)',PIF
```

```
      CALL CLEAR
      IF(PIF.EQ.' ') GO TO 8828
      IF(PIF.EQ.'S') THEN
      OPEN(84,FILE='PIFFEL',STATUS='OLD',ERR=8869)
      READ(84,*) XBX,YBX,ANGG,XLEN,XHT
      CLOSE(84)
      GO TO 8868
      ENDIF
      OPEN(85,FILE='M33RR4')
      WRITE(85,'(2X,A60)') PIF
      REWIND 85
      READ(85,*,ERR=8863) XBX,YBX,ANGG,XLEN,XHT
      OPEN(84,FILE='PIFFEL')
      WRITE(84,'(5F15.4)') XBX,YBX,ANGG,XLEN,XHT
      CLOSE(84)
      CLOSE(85,STATUS='DELETE')
      GO TO 8868
 8863 OPEN(84,FILE=PIF,STATUS='OLD',ERR=8869)
      READ(84,*,ERR=8869) XBX,YBX,ANGG,XLEN,XHT
      CLOSE(84)
      CLOSE(85,STATUS='DELETE')
 8828 OPEN(84,FILE='PIFFEL')
      WRITE(84,'(5F15.4)') XBX,YBX,ANGG,XLEN,XHT
      CLOSE(84)
 8868 CONTINUE
      CALL CLEAR
      XA=((XBX-SUBX-XMINX)/SM)+100.
      YA=((YBX-SUBY-YMINY)/SM)+100.
      P1=XA
      P2=YA
      ANG=ANGG/57.29578
      P3=((XBX-SUBX-XMINX+XLEN)/SM)+100.
      P4=P2
      P5=P3
      P6=((YBX-SUBY-YMINY+XHT)/SM)+100.
      P7=P1
      P8=P6
      CALL ROTATE(P1,XA,P2,YA,ANG)
      CALL ROTATE(P3,XA,P4,YA,ANG)
      CALL ROTATE(P5,XA,P6,YA,ANG)
      CALL ROTATE(P7,XA,P8,YA,ANG)
      CALL UMOVE(P1,P2)
      CALL UPEN(P3,P4)
      CALL UPEN(P5,P6)
      CALL UPEN(P7,P8)
      CALL UPEN(P1,P2)
      CALL CLEAR
      PRINT 8855,XBX,YBX,ANGG,XLEN,XHT
 8855 FORMAT(1X,2HX=,F12.2,3H Y=,F12.2,5H ANG=,F6.2,3H L=,F10.2,
     +3H H=,F12.2)
      PRINT,'(A\)',' REDO BOX ? (Y/N) ) '
      READ '(A1)',FTYPE
      CALL CLEAR
      IF(FTYPE.EQ.'Y') THEN
      CALL USET(CLRBCK)
      CALL UMOVE(P1,P2)
      CALL UPEN(P3,P4)
      CALL UPEN(P5,P6)
      CALL UPEN(P7,P8)
      CALL UPEN(P1,P2)
      CALL USET(COL1)
      GO TO 8869
      ENDIF
   19 PRINT*,'(2,42) WANT PARALLEL (TO BASE) OR NORMAL GRIDS ? '
      PRINT,'(A\)',' INPUT P OR N (DEF=P)  )
      READ '(A1)',FTYPE
      CALL CLEAR
      IF(FTYPE.EQ.' ') FTYPE='P'
      IF(FTYPE.NE.'P' AND.FTYPE.NE.'N') GO TO 19
 5578 PRINT,'(A\)',' (2.43) HOW MANY PROFILES ON THE PAGE ?(DEF=11) ) '
      READ '(A10)',EXAG
      IF(EXAG.EQ.' ') MANY=11
      IF(EXAG.NE.' ') THEN
      OPEN(43,FILE='M33443')
      WRITE(43,'(A10)') EXAG
      REWIND 43
      READ(43,*,ERR=5578) MANY
      CLOSE(43,STATUS='DELETE')
      ENDIF
      IF(MANY.NE.0) THEN
      PX=MIN(XLEN/SM,XHT/SM)/MANY
      CALL CLEAR
      PRINT '(A\)',' (2.44) ENTER SCALE EXAG. FACTOR (DEF=1.0) ) '
```

```
      READ '(A10)',EXAG
      READ(EXAG,'(F10.0)') EX
      IF(EX.EQ.0.) EX=1.0
      IF(EXAG.EQ.'0') EX=0.
      JF=0
      ENDIF
      IF(EX.GE.0.1) THEN
 3366 CALL CLEAR
      PRINT*,'MIN EL= ',ZMIN,' MAX EL= ',ZMAX,' FOR PROFILE SCALES'
      PRINT,'(A)',' ENTER NEW MIN & MAX EL''S. (DEF=NO CHANGE) '
      READ '(A4J)',NEWZS
      IF(NEWZS.NE.' ') THEN
      OPEN(43,FILE='M3345R')
      REWIND 43
      WRITE(43,'(A40)') NEWZS
      REWIND 43
      READ(43,*,ERR=3366) ZMIN,ZMAX
      CLOSE(43,STATUS='DELETE')
      ENDIF
      ENDIF
      EX=EX*PX
      IF(MANY.EQ.0) THEN
      CALL FENFIL(ZMAX,ZMIN,MM,X1,X2,Y1,Y2,EX,RDASK,0,XMINX,XMAXX,
     + YMINY,YMAXY,SM,0,FTYPE)
      CLOSE(81)
      GO TO 79
      ENDIF
      CALL CLEAR
      PRINT*,'THE "GRIDDED" PAGE PLOT FILES ARE NAMED WITH THE SAME'
      PRINT*,'ROOT AS THE .MSH FILE. EXTENSION ARE .OP, .1P, '
      PRINT,'(A)','  .ON, .1N, ETC   PRESS RETURN TO CONTINUE ) '
      READ '(A1)',E(1)
      READ(FIMESH,'(8A1)') (F(J),J=1,8)
      M=8
      DO 277 J=1,8
      IF(F(J).EQ.' '.OR.F(J).EQ.'.') THEN
      M=J-1
      GO TO 278
      ENDIF
  277 CONTINUE
  278 CONTINUE
      FROOT='
      WRITE(FROOT,'(8A1)') (F(J),J=1,M)
      IF(MANY.EQ.1) JF=1
      DO 77 I=1,MANY
      WRITE(FNME,'(A8,A3,A1)') FROOT,EXTEN(I),FTYPE
      READ(FNME,'(12A1)') (F(J),J=1,12)
      M=1
      DO 177 J=1,12
      E(J)=
      IF(F(J).NE.' ') THEN
      E(M)=F(J)
      M=M+1
      ENDIF
  177 CONTINUE
      WRITE(FNME,'(12A1)') (E(J),J=1,12)
      OPEN(23,FILE=FNME)
      IF(FTYPE.EQ.'P') THEN
      IF(MANY.NE.1) FD=(XHT)/(MANY-1)
      IF(MANY.EQ.1) FD=(XHT)/2.
      X1=XBX-SUBX
      X2=XBX+XLEN-SUBX
      Y1=(YBX-SUBY)+(I-1+JF)*FD
      Y2=Y1
      X1=((X1-XMINX)/SM)+100.
      X2=((X2-XMINX)/SM)+100.
      Y1=((Y1-YMINY)/SM)+100.
      Y2=((Y2-YMINY)/SM)+100.
      ANG=ANGG/57.29578
      XA=XBX-SUBX
      XA=((XA-XMINX)/SM)+100.
      YA=(YBX-SUBY)
      YA=((YA-YMINY)/SM)+100.
      CALL ROTATE(X1,XA,Y1,YA,ANG)
      CALL ROTATE(X2,XA,Y2,YA,ANG)
      ENDIF
      IF(FTYPE.EQ.'N') THEN
      IF(MANY.NE.1) FD=(XLEN)/(MANY-1)
      IF(MANY.EQ.1) FD=(XLEN)/2.
      X1=(XBX-SUBX)+(I-1+JF)*FD
      X2=X1
      Y1=YBX-SUBY
      Y2=YBX+XHT-SUBY
```

```
      X1=((X1-XMINX)/SM)+100.
      X2=((X2-XMINX)/SM)+100.
      Y1=((Y1-YMINY)/SM)+100.
      Y2=((Y2-YMINY)/SM)+100.
      ANG=ANGG/57.29578
      XA=(XBX-SUBX)
      XA=((XA-XMINX)/SM)+100.
      YA=YBX-SUBY
      YA=((YA-YMINY)/SM)+100.
      CALL ROTATE(X1,XA,Y1,YA,ANG)
      CALL ROTATE(X2,XA,Y2,YA,ANG)
      ENDIF
      VX=SM*(X1-100.)+XMINX
      VY=SM*(Y1-100.)+YMINY
      WRITE(23,*) VX,VY
      VX=SM*(X2-100.)+XMINX
      VY=SM*(Y2-100.)+YMINY
      WRITE(23,*) VX,VY
      PORN=
      IF(FTYPE.EQ.'N') PORN='*'
      IF(FTYPE.EQ.'P') PROFI='         GRID PROFILE '
      IF(FTYPE.EQ.'N') PROFI='         GRID PROFILE '
      WRITE(23,'(A22,I4,A1)') PROFI,I-1,PORN
      WRITE(23,*)
      GWEL='WATER TABLE ELEVATION '
      WRITE(23,'(A22)') GWEL
      IVC=1
      CALL CLEAR
      CALL FENFIL(ZMAX,ZMIN,MM,X1,X2,Y1,Y2,EX,RDASK,1,XMINX,XMAXX,
     +  YMINY,YMAXY,SM,IVC,FTYPE)
      CLOSE(23)
   77 CONTINUE
   79 CONTINUE
      IF(PASK.EQ.'Y') THEN
      IF(ZOOM.EQ.'C') CALL UDAREA(10.*PCX,75.*PCX,10.*PCY,75.*PCY)
      IF(ZOOM.EQ.'C') CALL UWINDO(1.,1040.,0.,780.)
      CALL USETT('SOFT')
      CALL UPSET('VERT',30.*PSFT)
      CALL UPSET('HORI',15.*PSFT)
      CALL UMOVE(100.,50.)
      CALL UPRNT1('VOL ABV Z= \','TEXT')
      CALL UPRNT1(REFELE,'REAL')
      CALL UPRNT1(' =\','TEXT')
      CALL UPRNT1(VOL,'REAL')
      CALL UPRNT1('  AV Z= \','TEXT')
      CALL UPRNT1(REF,'REAL')
      CALL UPRNT1('  AREA=\','TEXT')
      CALL UPRNT1(ATOT,'REAL')
      IF(MM.EQ.2) CALL USETT('HARD')
      CALL UDAREA(XUDMIN,XUDMAX,YUDMIN,YUDMAX)
      CALL UWINDO(XWMIN,XWMAX,YWMIN,YWMAX)
      CALL CLEAR
      PRINT*,'VOL ABOVE ',REFELE,'=',VOL,'  AREA= ',ATOT
      PRINT*,'AV Z = ',REF
      CALL UBELL
      CALL UPAUSE
      ENDIF
      CALL CLEAR
      IF(KW.EQ.0),THEN
      PRINT '(A\)',
     +'(2,41) WANT TO DRAW ANOTHER GRID BOX PROFILE ? (Y/N) ) '
      READ '(A1)',JASK
      IF(JASK.EQ.'Y') GO TO 1456
      ENDIF
      IF(KW.EQ.1),THEN
      PRINT '(A\)',
     +'(2,24a),WANT TO REDRAW THE CONTOURS ? (Y/N) '
      READ '(A1)',JASK
      CALL UERASE
      CALL UBACKG(CLRBCK)
      IF(JASK.EQ.'Y') GO TO 1456
      ENDIF
      ENDIF
      END IF
      CALL UERASE
      CALL UBACKG(CLRBCK)
      RETURN
      END
C
      SUBROUTINE XYMBOL(X,Y,SYM)
      CHARACTER SYM*1
      IF(SYM.EQ.' ') RETURN
      IF(SYM.EQ.'*') THEN
```

```
      CALL LINE(SYM,X,Y,1,6)
      CALL LINE(SYM,X,Y,3,4)
      CALL LINE(SYM,X,Y,2,5)
      CALL LINE(SYM,X,Y,7,8)
      ENDIF
      IF(SYM.EQ.'S') THEN
      CALL LINE(SYM,X,Y,1,3)
      CALL LINE(SYM,X,Y,3,6)
      CALL LINE(SYM,X,Y,6,4)
      CALL LINE(SYM,X,Y,4,1)
      ENDIF
      IF(SYM.EQ.'T') THEN
      CALL LINE(SYM,X,Y,1,3)
      CALL LINE(SYM,X,Y,3,5)
      CALL LINE(SYM,X,Y,5,1)
      ENDIF
      IF(SYM.EQ.'C') THEN
      CALL UCRCLE(X,Y,2.5)
      ENDIF
      IF(SYM.EQ.'D') THEN
      CALL LINE(SYM,X,Y,2,7)
      CALL LINE(SYM,X,Y,2,8)
      CALL LINE(SYM,X,Y,5,7)
      CALL LINE(SYM,X,Y,5,8)
      ENDIF
      RETURN
      END
C
      SUBROUTINE LINE(SYM,X,Y,I,J)
      DIMENSION U(8),V(8)
      CHARACTER SYM*1
      DATA U/-2.5,0.,2.5,-2.5,0.,2.5,-2.5,2.5/
      DATA V/-2.5,-2.5,-2.5,2.5,2.5,2.5,0.,0./
      W=1.0
      F=1
      IF(SYM.EQ.'D') W=1.25
      CALL UMOVE(X,Y)
      DO 100 K=1,1
      CALL UMOVE(X+F*W*U(I),Y+F*W*V(I))
  100 CALL UPEN(X+F*W*U(J),Y+F*W*V(J))
      RETURN
      END
C
      SUBROUTINE ROTATE(X,XA,Y,YA,ANG)
      XN=(X-XA)*COS(ANG)-(Y-YA)*SIN(ANG)
      YN=(Y-YA)*COS(ANG)+(X-XA)*SIN(ANG)
      X=XN+XA
      Y=YN+YA
      RETURN
      END
```

## APPENDIX C. FORTRAN LISTING FOR McGRID

```
DIMENSION X(2000),Y(2000),Z(2000),IC(20),FDUM(20),ZZV(2000)
DIMENSION IBB(40)
CHARACTER FILLE*24,ISG*1,HONOR*1,ZCHAR*20,ICBOR*1,FF(14)*1
CHARACTER IASK*1,TYP*4,SLASH*1,ASIS*1,DUM*200,FUM(1)*200
CHARACTER VORMAT*182,CORMAT*182,VOR1*90,VOR2*90,COR1*90,COR2*90
CHARACTER CDUM(20)*20,IFDUM(200)*1,DUMM*24
CHARACTER VOR(20)*9,COR(20)*9,L1*1,L2*6,L3*2,OPREN*1,CPREN*1
CHARACTER IOR(20)*9,M1*4,M2*1,M3*2,IOR1*90,IOR2*90,IORMAT*185
CHARACTER ONEX*3,DEFAUL*5
CHARACTER GX(200)*1,KEEP(10)*20,NKEEP*20,PCDEN*20,CID*20
CHARACTER CD(2000)*20,LLBL*10,TRASH*1
DATA ZZV/2000*1./
DATA GX/200*' '/
DATA ONEX/'1X,'/
DATA OPREN/'('/
DATA CPREN/')'/
DATA VOR/20*'          '/
DATA COR/20*'          '/
DATA IOR/20*'          '/
DATA CDUM/20*'XXXXXXXXXXXXXXXXXXX'/
DATA SLASH/'\'/
PRINT*,'*********************  HI ! ***************************'
PRINT*,'**THIS IS McCON---A GENERAL CONTOURING CODE****'
PRINT*,'.....TROUBLED....  CALL JOHN PALMERTON (601) 634-3357..'
PRINT*,'.....OR WRITE TO:  USAE WATERWAYS EXPERIMENT STATION'
PRINT*,'                   ATTN: WESGS-R'
PRINT*,'                   GEOTECHNICAL LABORATORY'
PRINT*,'                   SOIL AND ROCK MECHANICS DIVISION'
PRINT*,'                   3909 HALLS FERRY ROAD'
PRINT*,'                   VICKSBURG, MS 39180-0631'
PRINT*,'*************************************************************'
PRINT*,'  '
PRINT '(A\)',' PRESS ANY KEY TO CONTINUE ) '
READ '(A1)',IASK
PRINT*,' '
PRINT*,'THIS SECTION OF THE CONTOUR PACKAGE PROVIDES SEVERAL'
PRINT*,'OPTIONS FOR MANIPULATING YOUR DATA.  ESSENTIALLY, THIS'
PRINT*,'ROUTINE ALLOWS YOU TO NAME AN INPUT DATA FILE AND THEN'
PRINT*,'CREATES A FILE NAMED  ''MAP.DAT''  THAT IS USED BY THE'
PRINT*,'SUBSEQUENT CONTOURING ROUTINES.  IF THE ''MAPDAT'' FILE'
PRINT*,'YOU WANT WAS JUST PREVIOUSLY CREATED AND YOU WANT TO'
PRINT*,'GO DIRECTLY TO  ''CONTOURING''  USING THAT FILE THEN'
PRINT*,'   PRESS ''C'' NOW TO GO TO CONTOURING'
PRINT*,'OR PRESS ''V'' TO GET MORE VERBAGE,'
PRINT '(A)',' OR PRESS ''RETURN'' TO BEGIN INPUT PROCESS. ) '
READ '(A1)',IASK
IF(IASK.EQ.'C') STOP '..GOING TO CONTOURING...'
IF(IASK.EQ.' ') GO TO 17
PRINT*,' '
PRINT*,'..A WORD ABOUT THE INPUT FILE.  THIS ROUTINE WILL ASK'
PRINT*,'YOU FOR THE NAME OF A DATA FILE WHICH YOU HAVE PRE-'
PRINT*,'VIOUSLY PREPARED,, THIS DATA FILE MAY EITHER BE'
PRINT*,'  ''FREEFORM''  OR  ''COLUMNAR''.  IF ''FREEFORM'', THE'
PRINT*,'DATA MUST ALL BE NUMERIC (NO LETTERS ANYWHERE).  EACH'
PRINT*,'LINE OF THE FILE MUST CONTAIN THE X & Y  COORDINATES'
PRINT*,'OF A DATA POINT.  THE LINE MAY ALSO CONTAIN MULTIPLE'
PRINT*,'''Z''  (OR ELEVATION) VALUES FOR EACH DATA POINT.  IT'
PRINT*,'MAY ALSO INCLUDE A POINT IDENTIFIER COLUMN (NUMERIC)'
PRINT*,'AND A  ''SELECTOR'' COLUMN (USED TO SKIP OVER CERTAIN'
PRINT*,'DATA POINTS).  THE ROUTINE PRINTS THE FIRST FEW LINES'
PRINT*,'AND THEN ASKS YOU TO SELECT THE MEANING OF THE COLUMNS'
PRINT '(A\)',' PRESS ANY KEY TO CONTINUE ) '
READ '(A1)',IASK
PRINT*,' '
PRINT*,'..IF THE INPUT FILE IS ''FREEFORM''.'
PRINT*,'  THERE MUST BE THE SAME NUMBER OF ENTRIES ON EACH LINE'
PRINT*,'  EACH ENTRY MUST BE SEPARATED BY A COMMA OR BY ONE OR'
PRINT*,'  BLANKS.'
PRINT*,' '
PRINT*,'..IF THE INPUT FILE IS ''COLUMNAR''.'
PRINT*,'  ALL OF THE DATA COLUMNS MUST BE "RIGHT-JUSTIFIED"'
PRINT*,'  SUPPOSE THAT THE DATA LOOKS LIKE..'
PRINT*,' '
PRINT*,'    12002  1.3 231.5 22 156 0  23.4'
PRINT*,'  ,  12003 12.4  23.2  1     1 124.6'
PRINT*,' '
PRINT*,'  IF THE FILE'
PRINT*,'  CONTAINS FIELDS THAT ARE BLANK (5TH COLUMN OF 2ND'
PRINT*,'  LINE ABOVE, THEN THE INFORMATION IN THAT LINE IS'
PRINT*,'  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*,' '
      PRINT '(A\)','       PRESS ANY KEY TO CONTINUE ) '
      READ '(A1)',IASK
      PRINT*,' '
      PRINT*,'...ONE MORE THING.....'
      PRINT*,'IN THE EVENT THAT THE IDENTIFIER (WHEN SPECIFIED) IS'
      PRINT*,'REPEATED,,ONLY THE LAST LINE WITH THE SAME VALUE IS'
      PRINT*,'RETAINED.'
      PRINT*,' '
      PRINT '(A\)','       PRESS ANY KEY TO CONTINUE  ) '
      READ '(A1)',IASK
      PRINT*,' '
   17 PRINT*,'  '
      PRINT*,'  '
      PRINT '(A\)',
     +' (1.01a) ENTER (ORIGINAL) DATA FILE NAME (DEF=MCDEMO.DAT) ) '
      READ '(A24)',FILLE
      IF(FILLE.EQ.' ') FILLE='MCDEMO.DAT'
      OPEN(09,FILE=FILLE,STATUS='OLD',ERR=6)
      GO TO 7
    6 PRINT*,'FILE ',FILLE,' NOT FOUND....TRY AGAIN.'
      GO TO 17
    7 CONTINUE
      OPEN(10,FILE='TEM',STATUS='OLD',ERR=7390)
      CLOSE(10,STATUS='DELETE')
 7390 OPEN(10,FILE='TEM',STATUS='NEW')
      PRINT '(A\)',
     +' (1.01b) IS FILE "FREE-FIELD" OR "COLUMNAR" (F(DEF) OR C) ? ) '
      FREFOR=0
      READ '(A1)',ISG
      IF(ISG.EQ.'C') FREFOR=1
      PRINT '(A\)',
     +' (1.02) WANT INVERSE POWER GRIDDING ?(Y,N OR ?) ) '
      READ '(A1)',ISG
      TYP='TRI'
      IF(ISG.EQ.'?') THEN
      PRINT*,' '
      PRINT*,'THIS OPTION WILL CREATE A RECTANGULAR GRID OF DATA'
      PRINT*,'POINTS WHICH WILL ENCOMPASS THE ORIGINAL POINTS (WITH'
      PRINT*,'A SMALL BORDER).  THE CONTOURABLE 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*,' '
      PRINT*,'THE FORMULA FOR INVERSE POWER AVERAGING IS:'
      PRINT*,' '
      PRINT*,' Zg(i)=SUM[(Zo(j)/(1.+D(ij))**Q] / SUM[1./(1.+D(ij))**Q]'
      PRINT*,'   WHERE Zg(i) IS THE SOUGHT ELEVATION AT GRID POINT i'
      PRINT*,'         Zo(j) IS THE ELEVATION AT AN ORIGINAL POINT j'
      PRINT*,'         D(ij) IS THE DISTANCE FROM i TO j.'
      PRINT*,'         Q IS THE POWER (USUALLY 2)'
      PRINT*,'  '
      GO TO 17
      ENDIF
      HONOR='Y'
      ICBOR='N'
      Q=3
      ASIS='N'
      IF(ISG.EQ.'Y'.OR.ISG.EQ.'y') GO TO 814
      PRINT*,' (1.03) WANT ORIGINAL DATA POINTS "AS IS" ? (NO FILLING '
      PRINT '(A\)',' OUT TO EDGES OF PLOT) (Y/N) ) '
      READ '(A1)',ASIS
  814 CONTINUE
      IF(ASIS.NE.'Y'.AND.ASIS.NE.'y') THEN
      IF(ISG.EQ.'Y'.OR.ISG.EQ.'y') THEN
      TYP='GRD'
      PRINT '(A\)',
     +' (1.04) ENTER VALUE FOR Q (POWER FOR INVERSE FITTING) DEF=2 )'
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,NMBER,2)
      Q=NMBER
      ENDIF
      PRINT '(A\)',
     +' (1.05) REMOVE APRON FROM AROUND DATA POINTS ? (Y/N) ) '
      READ '(A1)',IASK
      ICBOR='Y'
      IF(IASK.EQ.'Y'.OR.IASK.EQ.'y') ICBOR='N'
```

```
      IBOR=0
      IF(ICBOR.EQ.'Y'.OR.ICBOR.EQ.'y') IBOR=2
      HONOR='Y'
      IF(ISG.EQ.'Y'.OR.ISG.EQ.'y') THEN
   18 PRINT '(A\)',
     +' (1.06) DO NOT HONOR ORIGINAL DATA POINTS (Y,N OR ? ) '
      READ '(A1)',IASK
      HONOR='Y'
      IF(IASK.EQ.'Y'.OR.IASK.EQ.'y') HONOR='N'
      IF(IASK.EQ.'?') THEN
      PRINT*
      PRINT*,'ORIGINAL DATA POINTS, IF HONORED, WILL BE ADDED TO THE'
      PRINT*,'GRIDDED POINTS.  THIS MAY BE DISIRABLE TO AVOID'
      PRINT*,'SMOOTHING OF PEAKS AND VALLEYS WHICH OFTEN RESULTS FROM'
      PRINT*,'INVERSE POWER AVERAGING.'
      PRINT*
      GO TO 18
      ENDIF
      ENDIF
      ENDIF
      VORMAT='(*)'
      CORMAT='(*)'
      IF(FREFOR.EQ.0) GO TO 8447
      REWIND 09
 8879 READ(09,'(A200)',END=8877) DUM
      READ(DUM,'(200A1)') (IFDUM(I),I=1,200)
      DO 8878 I=1,200
      IF(IFDUM(I).NE.' ') GX(I)='X'
 8878 CONTINUE
      GO TO 8879
 8877 CONTINUE
      REWIND 09
      DO 8876 I=1,200
 8876 IFDUM(I)=GX(I)
      PRINT '(1X,80A1)',(GX(I),I=1,80)
      DO 8450 I=200,1,-1
      IF(IFDUM(I).NE.' ') GO TO 8451
 8450 CONTINUE
 8451 ILAST=I+1
      J=2
      IBB(1)=0
      DO 8460 I=2,ILAST
      IF(IFDUM(I).EQ.' '.AND.IFDUM(I-1).NE.' ') THEN
      IBB(J)=I-1
      J=J+1
      ENDIF
 8460 CONTINUE
      J=J-1
      DO 8777 I=2,J
      L1='F
      L2='.0,1X,'
      IF(I.EQ.J) L2=',0    '
      NUMB=IBB(I)-IBB(I-1)
      IF(I.GT.2) NUMB=NUMB-1
      WRITE(L3,'(I2)') NUMB
      VOR(I-1)=L1//L3//L2
      L1='A'
      L2=',1X,'
      IF(I.EQ.J) L2='      '
      COR(I-1)=L1//L3//L2
      IF(I.EQ.2) WRITE(M3,'(I2)') NUMB
      IF(I.GT.2) WRITE(M3,'(I2)') NUMB+1
      M1='I'
      M2=','
      IF(I.EQ.J) M2=' '
      IOR(I-1)=M1//M3//M2
 8777 CONTINUE
      VOR1=VOR(1)//VOR(2)//VOR(3)//VOR(4)//VOR(5)//VOR(6)//
     +     VOR(7)//VOR(8)//VOR(9)//VOR(10)
      VOR2=VOR(11)//VOR(12)//VOR(13)//VOR(14)//VOR(15)//VOR(16)//
     + VOR(17)//VOR(18)//VOR(19)//VOR(20)
      COR1=COR(1)//COR(2)//COR(3)//COR(4)//COR(5)//COR(6)//
     +     COR(7)//COR(8)//COR(9)//COR(10)
      COR2=COR(11)//COR(12)//COR(13)//COR(14)//COR(15)//COR(16)//
     + COR(17)//COR(18)//COR(19)//COR(20)
      IOR1=IOR(1)//IOR(2)//IOR(3)//IOR(4)//IOR(5)//IOR(6)//
     +     IOR(7)//IOR(8)//IOR(9)//IOR(10)
      IOR2=IOR(11)//IOR(12)//IOR(13)//IOR(14)//IOR(15)//IOR(16)//
     +     IOR(17)//IOR(18)//IOR(19)//IOR(20)
      VORMAT=OPREN//VOR1//VOR2//CPREN
      CORMAT=OPREN//COR1//COR2//CPREN
      IORMAT=OPREN//ONEX//IOR1//IOR2//CPREN
 8447 CONTINUE
```

```
      PRINT*,' '
      PRINT*,'...FIRST FEW LINES OF DATA FILE  ',FILLE,' LOOKS LIKE...'
      PRINT*,' '
      IF(FREFOR.EQ.1) PRINT IORMAT,(I,I=1,J-1)
      PRINT '(1X,79A1)',(GX(I),I=1,79)
      REWIND 09
      DO 666 I=1,4
      READ(09,'(A200)',END=667) DUM
  666 PRINT '(1X,A79)', DUM
  667 CONTINUE
      IZ1=0
      IZ2=0
      IZ3=0
      IZ4=0
      IZ5=0
      DO 8899 JJ=1,10
 8899 KEEP(JJ)='
      JDEN=1
      IDEN=1
      ISEL=0
      IF(FREFOR.EQ.1) THEN
      PRINT '(A\)',
     +' (1.07a) WHICH COL IS "IDENTIFIER" ? (0 IF NONE, DEF=0) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,NMBER,0)
      IDEN=NMBER
      JDEN=0
      IF(IDEN.EQ.0) JDEN=1
      IF(IDEN.EQ.0) IDEN=1
      IELIM=0
      PRINT '(A\)',
     +' (1.07b) DO NOT REMOVE DUPLICATE LABEL ENTRYS ? (Y/N) ) '
      READ '(A1)',IASK
      IF(IASK.EQ.'y'.OR.IASK.EQ.'Y') IELIM=1
      PRINT '(A\)',
     +' (1.08) WHICH COL IS "SELECTOR" ? (0 IF NONE, DEF=0) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,NMBER,0)
      ISEL=NMBER
      IF(ISEL.NE.0),THEN
      PRINT '(A\)',' (1.09) ENTER 1ST SELECTOR "KEEP VALUE"(DEF=CR) ) '
      READ '(A20)',KEEP(1)
      JJ=1
 6693 IF(KEEP(JJ).NE.'                    ') THEN
      JJ=JJ+1
      PRINT '(A\)',' (1.09) ENTER NEXT SELECTOR "KEEP VALUE"(DEF=CR)) '
      READ '(A20)',KEEP(JJ)
      GO TO 6693
      ENDIF
      JJ=JJ-1
      ENDIF
      ENDIF
      PRINT '(A\)',' (1.10) WHICH COL IS "X" ? (DEF=1) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IX,1)
      PRINT '(A\)',' (1.11) WHICH COL IS "Y" ? (DEF=2) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IY,2)
      PRINT '(A\)',
     +' (1.12) WHICH COL IS "Z1" ? (-)SIGN IF DESIRED (DEF=3) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IZ1,3)
      PRINT '(A\)',
     +' (1.13) WHICH COL IS "Z2" ?(FOR Z1+OR-Z2)OR 0(DEF)WHEN DONE ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IZ2,0)
      IF(IZ2.NE.0) THEN
      PRINT '(A\)',
     +' (1.13) WHICH COL IS "Z3" ?(Z1+-Z2+-Z3)OR 0(DEF) WHEN DONE ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IZ3,0)
      ENDIF
      IF(IZ3.NE.0) THEN
      PRINT '(A\)',' (1.13) WHICH COL IS "Z4" ? (DEF=0) (ETC)  ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IZ4,0)
      ENDIF
      IF(IZ4.NE.0) THEN
      PRINT '(A\)',' (1.13) WHICH COL IS "Z5" ? (DEF=0) ETC) ) '
      READ '(A5)',DEFAUL
      CALL DEFALL(DEFAUL,IZ5,0)
      ENDIF
      IZMAX=MAX(ABS(IX),ABS(IY),ABS(IZ1),ABS(IZ2),ABS(IZ3),ABS(IZ4),
```

```
     +          ABS(IZ5),ABS(ISEL),ABS(IDEN))
      REWIND 09
      XCDEN=0
   14 CONTINUE
      XCDEN=XCDEN+1
      IF(FREFOR.EQ.1) READ(09,CORMAT,END=15,ERR=38)(CDUM(I),I=1,IZMAX)
      IF(FREFOR.EQ.0) READ(09,*,END=15,ERR=38) (FDUM(I),I=1,IZMAX)
      IF(FREFOR.EQ.1) THEN
      DO 6655 I=1,IZMAX
      FDUM(I)=0.0
      IF(I.EQ.ABS(IX).OR.I.EQ.ABS(IY).OR.I.EQ.ABS(IZ1).OR.I.EQ.
     +ABS(IZ2).OR.I.EQ.ABS(IZ3).OR.I.EQ.ABS(IZ4).OR.I.EQ.ABS(IZ5))THEN
      READ(CDUM(I),'(F20.0)') FDUM(I)
      ENDIF
 6655 CONTINUE
      ENDIF
      GO TO 39
   38 LCDEN=XCDEN
      PRINT*,'...DATA ERROR LINE ',LCDEN,' PROBABLY NON-NUMERIC DATA'
      PRINT*,'  LINE ',LCDEN,' IGNORED.'
      READ(09,'(A1)') IASK
      XCDEN=XCDEN-1.
      GO TO 14
   39 CONTINUE
      CID=CDUM(IDEN)
      IF(ISEL.GT.0) THEN
      NKEEP=CDUM(ISEL)
      DO 7733 II=1,JJ
      IF(NKEEP.EQ.KEEP(II)) GO TO 1403
 7733 CONTINUE
      GO TO 14
 1403 CONTINUE
      ENDIF
      CX=FDUM(ABS(IX))*SIGN(1.,IX)
      CY=FDUM(ABS(IY))*SIGN(1.,IY)
      CZ1=FDUM(ABS(IZ1))*SIGN(1.,IZ1)
      IF(ABS(CZ1).EQ.999) GO TO 14
      IF(CDUM(ABS(IZ1)).EQ.'          ') GO TO 14
      CZ=CZ1
      CZ2=0.
      CZ3=0.
      CZ4=0.
      CZ5=0.
      IF(IZ2.EQ.0) GO TO 171
      CZ2=FDUM(ABS(IZ2))*SIGN(1.,IZ2)
      IF(ABS(CZ2).EQ.999) GO TO 14
      IF(CDUM(ABS(IZ2)).EQ.'          ') GO TO 14
      CZ=CZ+CZ2
      IF(IZ3.EQ.0) GO TO 171
      CZ3=FDUM(ABS(IZ3))*SIGN(1.,IZ3)
      IF(ABS(CZ3).EQ.999) GO TO 14
      IF(CDUM(ABS(IZ3)).EQ.'          ') GO TO 14
      CZ=CZ+CZ3
      IF(IZ4.EQ.0) GO TO 171
      CZ4=FDUM(ABS(IZ4))*SIGN(1.,IZ4)
      IF(ABS(CZ4).EQ.999) GO TO 14
      IF(CDUM(ABS(IZ4)).EQ.'          ') GO TO 14
      CZ=CZ+CZ4
      IF(IZ5.EQ.0) GO TO 171
      CZ5=FDUM(ABS(IZ5))*SIGN(1.,IZ5)
      IF(ABS(CZ5).EQ.999) GO TO 14
      IF(CDUM(ABS(IZ5)).EQ.'          ') GO TO 14
      CZ=CZ+CZ5
  171 CONTINUE
      IF(JDEN.EQ.0) WRITE(10,2000) CX,CY,CZ,CID
      WRITE(PCDEN,'(F5.0)') XCDEN
      IF(JDEN.NE.0) WRITE(10,2000) CX,CY,CZ,PCDEN
      GO TO 14
   15 CONTINUE
 2000 FORMAT(1X,F19.9,1X,F19.9,1X,F19.9,1X,A20)
      REWIND 10
      CLOSE(09)
      OPEN(11,FILE='MAP.DAT')
      OPEN(8,FILE='MCINFO.DAT')
      READ(FILLE,'(14A1)') (FF(I),I=1,14)
      DO 58 I=1,14
      IF(FF(I).EQ.':') THEN
      K=1
      DO 19 J=I+1,14
      FF(K)=FF(J)
      K=K+1
   19 CONTINUE
      DO 77 J=K,14
```

```
 77 FF(I)=' '
    GO TO 53
    ENDIF
 58 CONTINUE
 53 CONTINUE
    FF(14)='\'
    WRITE(8,'(14A1)') (FF(I),I=1,14)
    PRINT*,' '
    PRINT*,' **********IMPORTANT**********************************'
    PRINT*,'OUTPUT DATA FILE IS `MAP.DAT`'
    PRINT*,' '
    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(THE MAXIMUM DISTANCE BETWEEN'
    PRINT*,'POINTS IS 1000 SCREEN UNITS). THEREFORE, ALL POINTS'
    PRINT*,'WITHIN 7 SCREEN UNITS OF EACH OTHER WILL BE COMBINED.'
    PRINT '(A\)'
   + ' (1,14) ENTER NEW CHECKING DISTANCE IF DESIRED (DEF=7) ) '
    READ '(A5)',DEFAUL
    CALL DEFALL(DEFAUL,MSET,7)
    DSET=MSET
    PRINT*,'.......WAIT'
    IGRID=12
    J=1
    XMIN=99.E30
    XMAX=-99.E30
    YMIN=XMIN
    YMAX=XMAX
    ZMIN=XMIN
    ZMAX=XMAX
    I=1
137 READ(10,2000,END=139) X(I),Y(I)
    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)
    I=I+1
    GO TO 137
139 CONTINUE
    NTOT=I-1
    REWIND 10
    SSX=(XMAX-XMIN)/800.
    SSY=(YMAX-YMIN)/800.
    SM=SSX
    IF(SSY.GT.SSX) SM=SSY
    IDUP=0
    LDUP=0
    I=1
 23 READ(10,2000,END=44) X(J),Y(J),Z(J),CD(J)
    DO 29 I=1,J-1
    DIST=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)/SM
    IF(DIST.LE.DSET) IDUP=IDUP+1
    IF(CD(I).EQ.CD(J).AND.IELIM.EQ.0) LDUP=LDUP+1
    IF((CD(I).EQ.CD(J).AND.IELIM.EQ.0).OR.DIST.LE.DSET) THEN
    X(I)=X(J)
    Y(I)=Y(J)
    IF(CD(I).NE.CD(J)) Z(I)=(Z(J)+Z(I))
    IF(CD(I).NE.CD(J)) ZZV(I)=ZZV(I)+1.
    IF(CD(I).EQ.CD(J).AND.IELIM.EQ.0) Z(I)=Z(J)
    IF(CD(I).EQ.CD(J).AND.IELIM.EQ.0) ZZV(I)=1.
    CD(I)=CD(J)
    GO TO 23
    ENDIF
 29 CONTINUE
    J=J+1
    GO TO 23
 44 J=J-1
    DO 445 I=1,J
445 Z(I)=Z(I)/ZZV(I)
    IF(IDUP.GT.0) THEN
    PRINT*,' '
    PRINT 4455,IDUP+1,MSET
4455 FORMAT(I12,15H POINTS WITHIN ,I4,27H SCREEN UNITS OF EACH OTHER)
    PRINT*,'ELEVATIONS AT THESE POINTS AVERAGED AND ASSIGNED TO'
    PRINT*,'POSITION OF FIRST (INVOLVED) POINT(S) FOUND IN FILE.'
    PRINT,'(A\)','.....PRESS RETURN TO CONTINUE'
    READ '(A1)',TRASH
    PRINT*,' '
    ENDIF
    IF(LDUP.GT.0) THEN
    PRINT*,' '
```

```
      PRINT 4456,LOUP+1
 4456 FORMAT(I12,17H DUPLICATE LABELS)
      PRINT*,'INFO FOR LAST DUPLICATE LABEL IN FILE WAS USED.'
      PRINT*
      PRINT '(A\)','.....PRESS RETURN TO CONTINUE'
      READ '(A1)',TRASH
      PRINT*
      ENDIF
      XMAX=-99.E30
      XMIN=-XMAX
      YMAX=XMAX
      YMIN=XMIN
      DO 33 I=1,J
      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)
      IF(Z(I).GT.ZMAX) ZMAX=Z(I)
      IF(Z(I).LT.ZMIN) ZMIN=Z(I)
   33 CONTINUE
      SUBX=0.
      SUBY=0.
      IF(ABS(XMIN).GT.100.*(XMAX-XMIN).OR.
     +   ABS(YMIN).GT.100.*(YMAX-YMIN)) THEN
      LOG=LOG10(XMIN)
      IF(XMIN.LT.1.) LOG=LOG-1
      XMANTIS=XMIN/(10.**LOG)
      IM=XMANTIS*1000
      XM=IM
      XMIN=(XM/1000)*(10.**LOG)
      LOG=LOG10(YMIN)
      IF(YMIN.LT.1.) LOG=LOG-1
      YMANTIS=YMIN/(10.**LOG)
      IM=YMANTIS*1000
      YM=IM
      YMIN=(YM/1000)*(10.**LOG)
      SUBX=2168000
      SUBY= 185200
      DO 551 I=1,J
      X(I)=X(I)-SUBX
  551 Y(I)=Y(I)-SUBY
      ENDIF
      ZMULT=1.
  920 PRINT*,'  '
      PRINT 8888,MIN(ZMIN,ZMAX),MAX(ZMIN,ZMAX),J
 8888 FORMAT(6H ZMIN=,F15.5,6H ZMAX=,F15.5,16H # DATA POINTS= ,I4)
      PRINT '(A\)'
     +' (1.15a) ENTER MULTIPLIER FOR Z (RETURN IF HAPPY) ) '
      READ '(A20)',ZCHAR
      IF(ZCHAR.NE.'                    ') THEN
      READ(ZCHAR,'(F20.0)') ZMULT
      DO 923 I=1,J
  923 Z(I)=Z(I)*ZMULT
      ZMIN=ZMIN*ZMULT
      ZMAX=ZMAX*ZMULT
      PRINT*,'  '
      GO TO 920
      ENDIF
      FACX=1.
      FACY=1.
  820 PRINT*,'  '
      PRINT 8889,MIN(XMIN,XMAX),MAX(XMIN,XMAX),MAX(XMIN.XMAX)
     + -MIN(XMIN,XMAX),MIN(YMAX,YMIN),MAX(YMAX,YMIN),
     + MAX(YMAX,YMIN)-MIN(YMAX,YMIN)
 8889 FORMAT(6H XMIN=,F15.5,6H XMAX=,F15.5,6H DIFF=,F15.5,/,
     +       ,6H YMIN=,F15.5,6H YMAX=,F15.5,6H DIFF=,F15.5)
      PRINT '(A\)'
     +' (1.15b) ENTER MULTIPLIERS FOR X AND Y (RETURN IF HAPPY) ) '
      READ '(A20)',ZCHAR
      IF(ZCHAR.NE.'                    ') THEN
      OPEN(43,FILE='M3445R')
      WRITE(43,'(A20)') ZCHAR
      REWIND 43
      READ(43,*,ERR=821) XMULT,YMULT
      GO TO 822
  821 PRINT*,'ENTER TWO NUMBERS SEPARATED BY A COMMA.'
      CLOSE(43)
      GO TO 820
  822 CONTINUE
      CLOSE(43,STATUS='DELETE')
      FACX=FACX*XMULT
      FACY=FACY*YMULT
      DO 824 I=1,J
```

```
      X(I)=X(I)*XMULT
824   Y(I)=Y(I)*YMULT
      XMIN=XMIN*XMULT
      YMIN=YMIN*YMULT
      XMAX=XMAX*XMULT
      YMAX=YMAX*YMULT
      GO TO 820
      ENDIF
      XMIN=99.E30
      YMIN=XMIN
      XMAX=-XMIN
      YMAX=XMAX
      DO 553 I=1,J
      IF(X(I).LT.XMIN) XMIN=X(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      IF(X(I).GT.XMAX) XMAX=X(I)
      IF(Y(I).GT.YMAX) YMAX=Y(I)
553   CONTINUE
      PRINT*,'.....WAIT'
      WRITE(8,*) ZMULT
      WRITE(8,'(2A1)') HONOR,SLASH
      WRITE(8,'(2A1)') ICBOR,SLASH
      PQ=0
      IF(TYP.EQ.'TRI\') PQ=0.
      WRITE(8,*) PQ
      WRITE(8,'(A4)') TYP
      WRITE(8,*) FACX,FACY
      WRITE(8,*) SUBX,SUBY
      CLOSE(8)
      WRITE(11,'(14A1)') (FF(I),I=1,14)
      WRITE(11,*) FACX,FACY
      WRITE(11,*) SUBX,SUBY
      IF(ASIS.EQ.'Y'.OR.ASIS.EQ.'y') THEN
      DO 87 I=1,J
      LLBL=CD(I)
      IF(JDEN.EQ.1) LLBL='          '
 87   WRITE(11,*) X(I),Y(I),Z(I),I+1000,' ',LLBL
      GO TO 158
      ENDIF
      IASK='N'
      IF(ISG.EQ.'Y'.OR.ISG.EQ.'y') THEN
      CC=999-J-1
      IF(HONOR.NE.'Y') CC=998
      IGG=(-1.5+SQRT(2.25+4.*CC))/2.
      IF(IBOR.EQ.2) IGG=IGG-2
      PRINT*,'(1,16) WANT NORMAL(N) OR FINE(F) INVERSE POWER'
      PRINT,'(A1)',' DISTANCE GRIDDING (DEF=N) ? (F OR N) ) '
      READ '(A1)',IASK
      PRINT*,'.....WAIT.....'
      IF(IASK.EQ.'F') IGRID=IGG
      ENDIF
      XGRD=IGRID
      DMAX=((XMAX-XMIN)**2+(YMAX-YMIN)**2)**0.5
      MLAB=1
      DO 30 I2=1,IGRID+1+IBOR
      K=(-1)**I2
      IF(K.LT.0) K=0
      YB=YMIN+(I2-1-IBOR/2)*((YMAX-YMIN))/XGRD
      DO 30 I1=1,IGRID+1+IBOR-1*K
      XB=(K*0.5*((XMAX-XMIN)/XGRD))+XMIN+(I1-1-IBOR/2)*((XMAX-XMIN))
     +   /XGRD
      ST=0.
      SB=0.
      DO 24 I=1,J
      D=((X(I)-XB)/FACX)**2+((Y(I)-YB)/FACY)**2))**.5
      IF(ISG.NE.'Y'.AND.ISG.NE.'y') THEN
      IF(D.LT.0.0075*DMAX) GO TO 30
      ENDIF
      IF(HONOR.EQ.'Y'.OR.HONOR.EQ.'y') THEN
      IF(D.LT.0.0075*DMAX) GO TO 30
      ENDIF
      ST=ST+(Z(I)/((D+1.)**Q))
      SB=SB+(1./((D+1.)**Q))
24    CONTINUE
25    CONTINUE
      IF(ISG.NE.'Y'.AND.ISG.NE.'y') THEN
      IF(I2.EQ.1) THEN
      WRITE(11,*) XB,YB,ST/SB,MLAB,'          '
      GO TO 28
      ENDIF
      IF(I1.EQ.1.AND.K.EQ.0) THEN
      WRITE(11,*) XB,YB,ST/SB,MLAB,'          '
      GO TO 28
```

```
      ENDIF
      IF(I2.EQ.IGRID+1+IBOR) THEN
      WRITE(11,*) XB,YB,ST/SB,MLAB,'
      GO TO 28
      ENDIF
      IF(I1.EQ.IGRID+1+IBOR-1*K.AND.K.EQ.0) THEN
      WRITE(11,*) XB,YB,ST/SB,MLAB,'
      GO TO 28
      ENDIF
   28 CONTINUE
      ENDIF
      IF(ISG.NE.'Y'.AND.ISG.NE.'y'),GO TO 31   ,
      WRITE(11,*) XB,YB,ST/SB,MLAB,'
      MLAB=MLAB+1
   31 CONTINUE
   30 CONTINUE
      IF(HONOR.EQ.'Y'.OR.HONOR.EQ.'y') THEN
      DO 55 I=1,J
      LLBL=CO(I)
      IF(JDEN.EQ.1) LLBL='          '
      WRITE(11,*) X(I),Y(I),Z(I),MLAB+1000,' ',LLBL
      MLAB=MLAB+1
   55 CONTINUE
      ENDIF
  158 ENDFILE 11
      REWIND 11
      OPEN(22,FILE='MCPICK.DAT')
      READ(11,'(A1)') FF(1)
      READ(11,'(A1)') FF(1)
      READ(11,'(A1)') FF(1)
  155 READ(11,160,END=157) XB,YB,ST,DUMM
      XB=(XB/FACX)+SUBX
      YB=(YB/FACY)+SUBY
      WRITE(22,160) XB,YB,ST,DUMM
      GO TO 155
  157 CONTINUE
  160 FORMAT(2F16.0,F16.2,A24)
      CLOSE(22)
      CLOSE(11)
      PRINT*,' FILE ''MCPICK.DAT'' NOW CONTAINS A COPY OF ALL '
      PRINT*,' DATA POINTS YOU HAVE SELECTED PLUS ANY GENERATED'
      PRINT*,' (GRIDDED) POINTS.'
      PRINT*,''
      STOP '....NOW GOING TO CONTOURING ACTIVITY....'
      END
C
      SUBROUTINE DEFALL(D,N,M)
      CHARACTER D*5,DD(5)*1
      N=M
      IF(D.EQ.'     ') RETURN
      READ(D,'(5A1)') (DD(I),I=1,5)
      DO 100 I=1,5
      IF(DD(I).EQ.' ') GO TO 100
      IF(DD(I).EQ.'-'.OR.DD(I).EQ.'+') GO TO 100
      IF(DD(I).LT.'0'.OR.DD(I).GT.'9')  RETURN
  100 CONTINUE
      READ(D,'(I5)') N
      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.
1. Contours (Cartography) -- Computer programs. 2. Military sketching -- Data processing. 3. Cartography -- Computer programs.
I. United States. Army. Corps of Engineers. II. U.S. Army Engineer Waterways Experiment Station. III. Computer Applications in Geotechnical Engineering (CAGE) IV. Title. V. Title: McCON -- a general contouring 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