NASA TECHNICAL NOTE

NASA TN D-6022

c.1

NASA TN D-6022

A GENERALIZED DIGITAL CONTOURING PROGRAM

by Ruben L. Jones

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1971

1. Report No. NASA TN D-6022	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A GENERALIZED DIGITAL CONTOURING PROGRAM		5. Report Date January 1971
		6. Performing Organization Code
7. Author(s) Ruben L. Jones		8. Performing Organization Report No. L-7273
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		10. Work Unit No. 185-42-12-01
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Technical Note
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program which generates contour charts from depth matrices has been written. The maximum dimensions of the depth matrix are 60 by 60. Each element of the matrix is computed from a weighted series of gradients, each of which is computed from neighboring control points. A maximum of 1000 control points can be processed at one time.

17. Key Words (Suggested by Author(s)) Contour Charts	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 78	22. Price* $3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

A GENERALIZED DIGITAL CONTOURING PROGRAM

By Ruben L. Jones
Langley Research Center

SUMMARY

A generalized digital contouring program is presented and discussed. The contouring program was developed by combining desirable characteristics from several existing contouring programs and can be easily adapted to many different research requirements. The overlaid structure of the program permits desired modifications to be made with ease.

The contouring program performs both the task of generating a depth matrix from either randomly or regularly spaced surface heights and the task of contouring the data. Each element of the depth matrix is computed as a weighted mean of heights predicted at an element by planes tangent to the surface at neighboring control points. Each contour line is determined by its intercepts with the sides of geometrical figures formed by connecting the various elements of the depth matrix with straight lines.

Both input and output of the digital program are described, as well as the critical program variables and tests. The program variables, subroutines, overlaid programs, and the listing of each are described and cross-referenced. A sample problem composed of 552 data points is furnished. The resulting computer output listing, as well as the contour chart, is shown.

INTRODUCTION

Contour charts are usually thought of as being two-dimensional pictorial representations of topographic formations of land masses relative to a smooth mathematical reference surface (datum surface). Contour charts can also be useful in portraying data which are obtained during the course of research in various scientific disciplines and which would ordinarily be tabulated. Any set of data, such as barometric pressure relative to sea level, whether computed or observed, which can be referenced to a two-dimensional coordinate system can in principle be represented graphically in the form of a contour chart. Further, scalar variations in three-dimensional space are representable by a series of contour charts, each of which represents the observations within a section through the field. Mathematical formulations of complex problems can also be studied with the aid of contour charts. In fact, the utility of contour charts is limited only by the user's ingenuity in adapting the data to a form suitable for graphic representation (contouring).

Since the usual technique for drawing contour charts requires that each scalar magnitude be plotted within a rectangular grid before contour lines are drawn, a digital contouring program is desirable from the standpoints of efficiency and manpower requirements. Several existing contouring programs (for example, ref. 1) were carefully studied for accuracy, efficiency, and so forth, and each was found to have desirable characteristics. A digital contouring program incorporating the best characteristics of each as well as certain improvements in data processing was considered desirable. The objective was to develop a simplified, generalized digital contouring program, which could be readily adapted to many different research needs. This paper describes that program and the analytical concepts upon which it is based, as well as its operation. The program is written for the Control Data 6000 series computer systems and relies on library functions and routines found in references 2 and 3.

SYMBOLS

A,B,C,A',B',C',D'	coefficients of the equation of a plane
F	function
H	elevation (scalar variation) above reference plane or line
i,j	variables
$\bar{i},\bar{j},\bar{k}$	x-, y-, and z-component, respectively, of a vector
k,k'	last variable in a series
m,l	dimensions of depth matrix
N	magnitude of a vector normal to surface or curve
$\bar{N}$	vector normal to surface or curve
P	control point
$\bar{Q}$	vector connecting two control points

R	ratio between two radial distances, r_j/r_k
r	radial distance between two points
U	discrepancy in a variable
W	computed weight
X,Y	rectangular coordinate axes with arbitrary origin
x,y	coordinates of a scalar variation in X-Y coordinate system
Z	function representing elevation (height) of a plane at an arbitrary point in that plane
z	height of plane at arbitrary point
δ	incremental change or discrepancy

Subscripts:

i,j,n	arbitrary variables
k,k'	last variable in a series
l	particular variable or reference
x,y	x- and y-components of a vector

DESCRIPTION OF PROBLEM AND METHOD OF SOLUTION

The science of graphic representation of scalar variables as a function of two independent variables has been the result of two advances in computer technology – first, the development of supporting equipment which can interpret computer instructions to a mechanical plotter and second, the ability to process large masses of data at one time.

Digital contouring programs, as a rule, perform two basic functions. First, a grid of equally spaced surface heights (terrain, for example) is generated from randomly spaced surface elevations. Since each coordinate of the grid point is equally distant from its neighbors, a grid for surface heights can be represented by a matrix of elevations (the "depth matrix"). Thus, the value at each element (mesh point) of the depth matrix represents an elevation above or below a given reference, and the location of the element in the array (matrix) corresponds to its location within the grid relative to the grid origin (the (1,1) element of the matrix). The second function of a contouring program is plotting the contours of the surface represented by the depth matrix.

Depth Matrix

A segment of a surface profile within a section through a three-dimensional surface is shown in figure 1. Within the figure the control points P_j and P_{j+n} are near neighbors to the ith grid point, and the planes Z_j and Z_{j+n}, which are tangent to the surface at P_j and P_{j+n}, will intersect the plane section (plane of the page) as shown. As can be seen, the heights of the planes at i are z_j and z_{j+n}, and each is an approximation of the height H_i of the profile at the grid point. Further, an improved approximation of H_i is obtained from the average of the z values, and when each Z_j is assigned a weight W_j, which is a normalized monotonic function of the distance r_j of j from the grid point i, a better estimate to the height of the profile at i is given by the expression

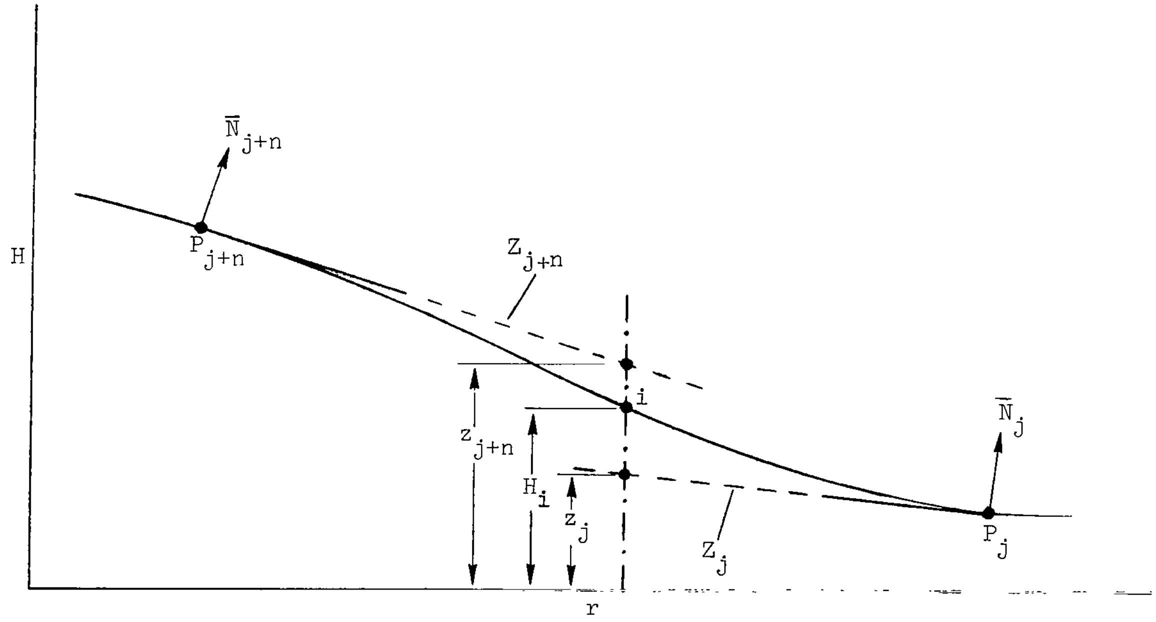

Figure 1.- Segment of a surface profile passing through a grid point and two control points.

$$H_i = \sum_{j=1}^{k'} W_j Z_j \tag{1}$$

where k' represents both the total number of control points entering into the computation of H_i and the control point farthest from i. If

$$R_l = \frac{r_l}{r_{k'}}$$

$$R_j = \frac{r_j}{r_{k'}}$$

when

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2} \qquad (2)$$

then, since an expression for a normalized weight must satisfy the condition

$$\sum_{j=1}^{k'} W_j = 1$$

an expression for W_j, which is a monotonic function of the radial distance of j from the ith grid point, can be written as

$$W_j = \frac{(1 - R_j)^2 R_j^{-2}}{\sum_{l=1}^{k'} (1 - R_l)^2 R_l^{-2}} \qquad (3)$$

As can be shown, W_j diminishes from a maximum of 1 when j = i to a minimum of zero when j = k'.

From the foregoing calculations, it is apparent that the height H_i of a surface above a reference plane can be estimated as a weighted series of heights, each of which is the height predicted at the grid point by a plane tangent to the surface at each control point. The accuracy of each estimated height is a function of the number k' of neighboring control points used in the computation and of the distribution of the control points about the grid point, as well as their proximity to the grid point. Further, conditions were placed on the spacing of the control points relative to each other, and each height H_i is independent of grid spacing and orientation.

As stated, the elevation at each grid point is estimated from a weighted series of heights, each of which is the predicted height of a tangent plane above a reference plane (datum plane). If F_i is a function in x, y, and z which represents the equation of a plane tangent to the surface at the ith control point, then a general equation for the tangent plane is

$$F_i\left(x_i,y_i,z_i\right) = A_i' + B_i'x_j + C_i'y_j + D_i'z_j = 0$$

where z_j is the height of the tangent plane above the datum plane at any $\left(x_j,y_j\right)$. Further, since the surface gradient is proportional to a unit vector $\overline{N}_i$ normal to the surface at i,

$$\text{Gradient} = \bar{\nabla}F_i = B_i'\,i + C_i'\,j + D_i'\,k \sim \overline{N}_i$$

where $\bar{\nabla}$ is a mathematical operator having the form

$$\frac{\partial}{\partial x}\,\bar{i} + \frac{\partial}{\partial y}\,\bar{j} + \frac{\partial}{\partial z}\,\bar{k}$$

Thus,

$$N_{x,i} \sim B_i'$$

$$N_{y,i} \sim C_i'$$

$$N_{z,i} \sim D_i'$$

are the components of a vector normal to the surface at the ith control point.

The particular equation of the tangent plane is determined at the ith control point by setting

$$A_i = \frac{A_i'}{D_i'} = -B_i x_i - C_i y_i - z_i \qquad (4)$$

where

$$\left.\begin{aligned} N_{x,i} &= B_i = \frac{B_i'}{D_i'} \\ N_{y,i} &= C_i = \frac{C_i'}{D_i'} \end{aligned}\right\} \qquad (5)$$

and at any x_j, y_j

$$Z_j \equiv Z_j(x_j, y_j) \equiv -A_i - B_i x_j - C_i y_j = z_j \tag{6}$$

Each gradient for the ith control point is computed from a least-squares minimization process. If $\bar{Q}_j$ is a vector joining a neighboring jth control point (a total of k) with the ith control point (the reference control point), the unit normal $\bar{N}_i$ which minimizes the weighted sum of the squares of the projections of each $\bar{Q}_j$ onto $\bar{N}_i$ is found. Analytically,

$$\sum_{j=1}^{k} W_j \left(\bar{N}_i \cdot \bar{Q}_j \right)^2 = \text{Minimum} \tag{7}$$

where W_j is the weight for each control-point pair as defined by equation (3) when $k' = k$. From equation (7), the $\bar{i}$, $\bar{j}$, and $\bar{k}$ components of $\bar{N}_i$ are determined.

The limiting indices k and k' imply that the number of control points utilized for the gradient computation and the grid-point computation may differ. In practice, a value $k = k' = 10$ is usually satisfactory for all the computations in the expressions discussed above. It is, however, required that the distribution of control points about the ith point be reasonably uniform. For instance, if all the points should lie along a straight line or in a narrow band, the solution will be indeterminate. Thus, the digital contouring program searches the input data in an attempt to select only those neighboring points which will permit a good solution to each $\bar{N}_i$. To search the data, neighboring points are selected by incrementing the inner and outer radii of a ring by one-tenth of the distance between grid points (the difference between the two radii). Further, to avoid the possibility of the control points being on a line with i, the sine of the angle included by each pair of $\bar{Q}_i$ vectors is not allowed to be less than 0.17. The inner and outer radii are incremented until either k control points have been selected or the outer radius has reached a value equal to 10 times the grid separation. If the distribution of points is poor, the gradient at the ith control point is rejected, and processing is continued until each of the remaining gradients have been either evaluated or rejected.

From the foregoing discussion, it is important to note that each of the expressions is independent of all assumptions pertaining to the depth-matrix orientation, origin, and so forth. Thus, the origin or spacing of the equally spaced surface grid can be changed at will without affecting the computed gradients of the represented surface.

At each control point the digital computer program evaluates first the gradient as described and then each coefficient in equation (5). The tangential-plane coefficients at each control point are then utilized in equation (1) to evaluate the scalar variation at each mesh (grid) point in the depth matrix. All computations are performed in the units of the x- and y-coordinates for the control points.

Numerical Techniques

A basic assumption of the contouring routines in the program is that each element in the matrix is both positive and nonzero. Since a particular element evaluated as described can be legitimately either negative or zero, a positive bias is added to each element after evaluation. The bias is determined by the program as an integral multiple of 10 000. The integer multiplier is determined as the least value which when multiplied by 10 000 and added to either the minimum anticipated scalar variation or the actual minimum data point (a program option) will result in a positive nonzero number. Unfilled elements in the depth matrix (a result of insufficient control-point density) can then be detected and filled by a subsequent routine.

The depth matrix is subjected to numerous tests (for example, smoothness) and is corrected where desirable. If the relative difference in elevation between two consecutive elements of the depth matrix is found to exceed a maximum slope assigned by the user, the elements are adjusted relative to each other. The net effect of this operation is to smooth the depth matrix. Further, if elements within the depth matrix are found to be unfilled due to inadequate control-point distribution, the element is filled by use of either interpolation or extrapolation. Finally, dimensions of the matrix cannot exceed 60 by 60 or be less than 3 by 3.

One test is performed to aid the user in evaluating the quality of the matrix fit to the actual input data. The actual position of each control point is used to obtain an estimate from the depth matrix of its predicted scalar variation by interpolating between elements of the matrix. When the predicted and observed values are compared, a slight discrepancy δH_i should result, since the matrix is at best an approximation to the true surface. Each δH_i can then be categorized on the basis of whether it falls within the limits

$$U_j + \frac{\delta U}{2} \geqq \delta H_i \geqq U_j - \frac{\delta U}{2} \tag{8}$$

where

$$U_j = U_{j-1} + \delta U \tag{9}$$

Thus, the distribution of the discrepancies is determined and plotted as a frequency distribution curve. The program allows for all values of U_j where

$$-2000 \leqq U_j \leqq 2000$$

and

$$\delta U = 20$$

Contour Plots

The program contours one 3 by 3 submatrix of the m by l depth matrix at a time. Basically, by assuming no interpolation between mesh points, each 3 by 3 submatrix can be subdivided into a series of adjacent triangles by connecting each set of three elements with a straight line as shown in figure 2. Contour intercepts are then computed along each side of each triangle beginning with triangle I and progressing clockwise around the squares. In the figure, the order in which each side of a triangle is considered is also shown. Intercepts are determined by assuming that each side is sufficiently small to permit the elevation difference to be linearly related to the side length. (See ref. 1.)

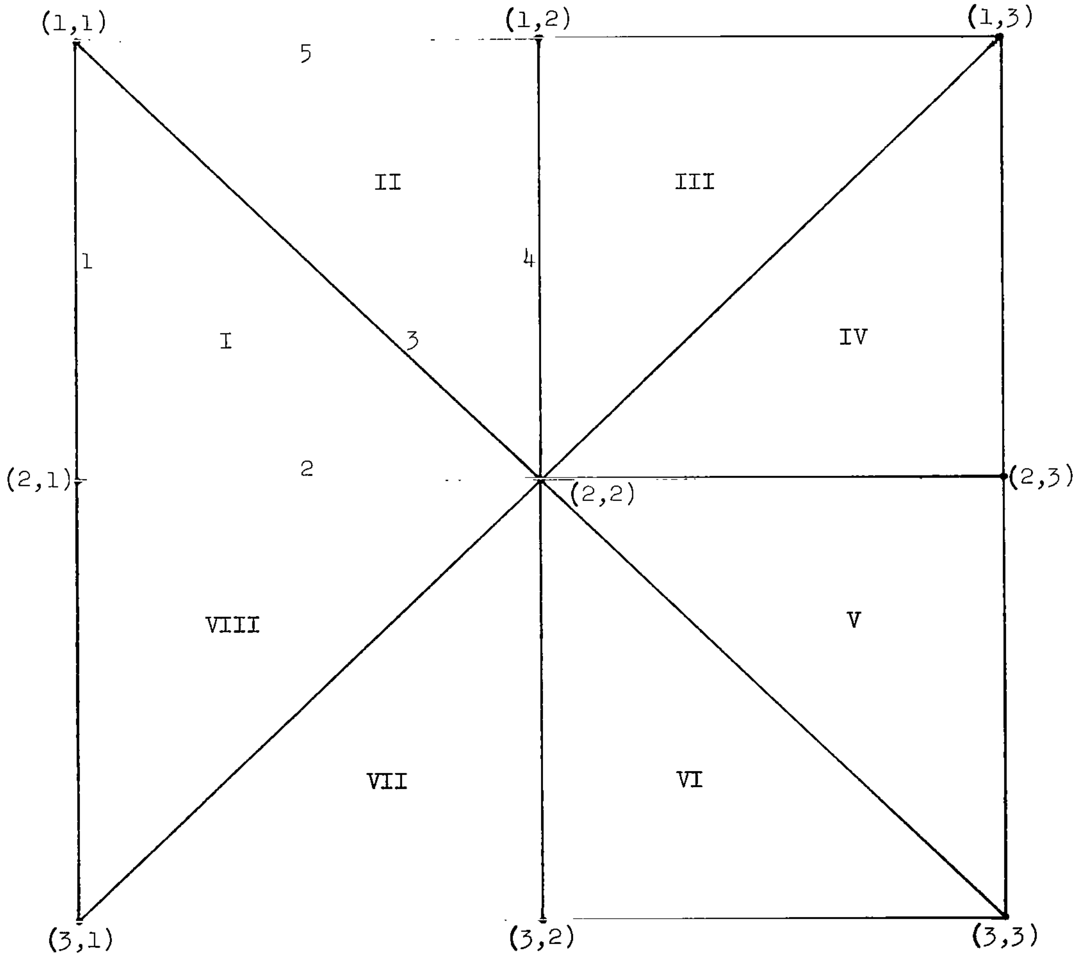

Figure 2.- The geometry of a 3 by 3 submatrix when subdivided by connecting groups of three elements with straight lines.

The procedure is always the same regardless of the number of elements in the depth matrix or the order of interpolation required between grid points. Interpolation between grid points is a linear expansion of each 3 by 3 submatrix into a larger submatrix. This process is repeated within the expanded submatrix until each element of the original 3 by 3 matrix has been considered. At this point a new 3 by 3 submatrix is processed in a similar manner, the third column of the first matrix becoming the first column of the second matrix.

The depth matrix is processed three rows at a time. If the number of elements in any one direction of the depth matrix is even, the resulting contour chart is terminated prematurely in that direction. Thus, it is conceivable that the depth matrix may not be plotted in its entirety.

PROGRAM ORGANIZATION AND DESCRIPTION

The digital contouring program is written in FORTRAN IV for the Control Data 6000 series computer systems, is overlaid, and requires approximately 65 000 octal words of storage. The overlay structure is shown in figure 3.

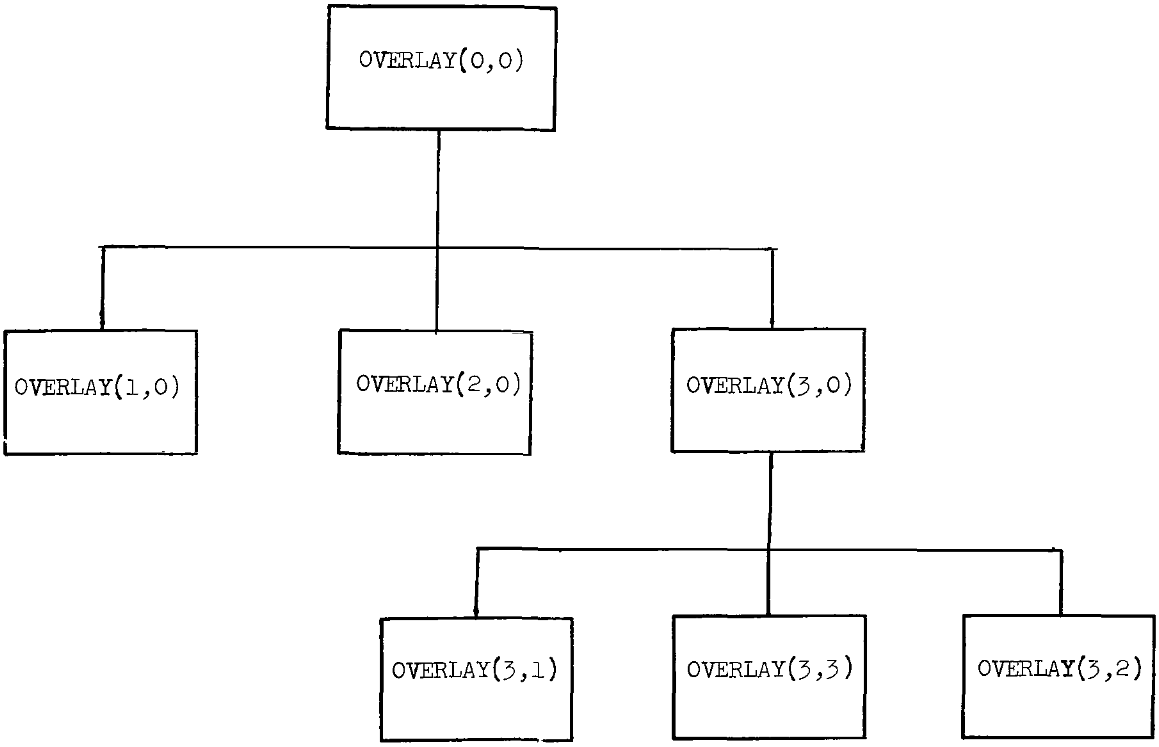

Figure 3.- Overlay structure.

Each overlaid program and the subroutines associated with it are listed in table I. For each program or subroutine, the required storage (in octal words) and the function are given. Further, the page numbers serve as an index to the listing and the more detailed discussion of each nonsystem program and subroutine. System subroutines are not discussed in detail.

Figure 4 is a flow chart showing the interrelation of the various overlaid programs and subroutines. The arrowheads indicate the direction of flow, and the circled numbers (1, 2, 3, etc.), with the exceptions of 8 and 9, indicate the order in which each overlay and subroutine is executed. The connectors 8 and 9 refer to disk files which were generated in one overlay as input to another routine. The letters a, b, and c indicate the order of flow from and to common program connectors.

The flow chart is rather detailed in its portrayal of the major functions of each program overlay and of each major subroutine. The various program options are indicated, as well as some of the more critical decisions. Emphasis has been placed on the program decisions which would terminate the execution of a routine in a normal fashion. It is necessary, however, to point out that certain program diagnostics which will abort the program are not shown but will be discussed subsequently. Additional flow charts are considered to be redundant and are not presented.

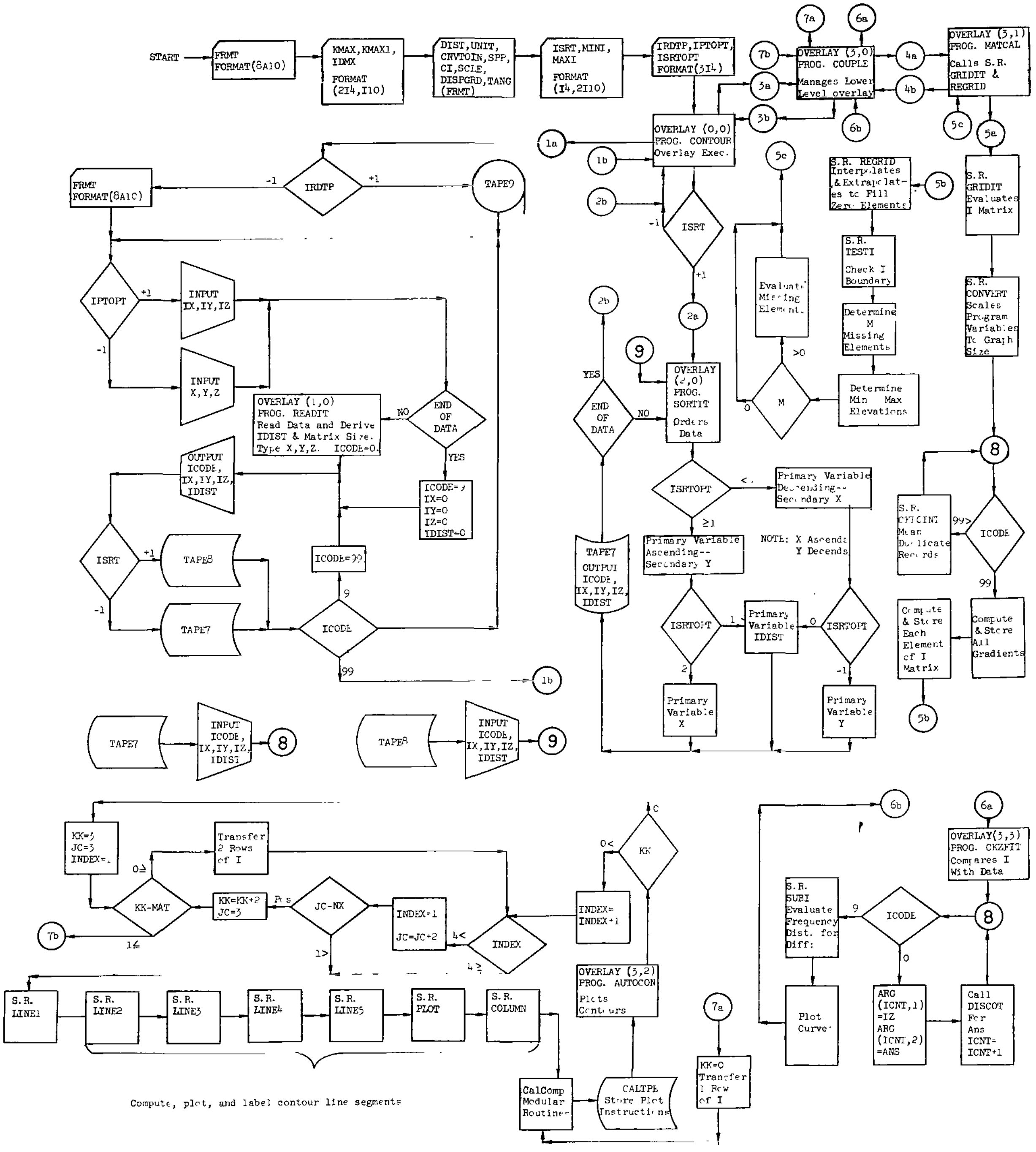
START
FRMT FORMAT(8A10)
KMAX,KMAX1, IDMX FORMAT (2I4,I10)
DIST,UNIT, CRVTOIN,SPP, CI,SCLE, DISPGRD,TANG (FRMT)
ISRT,MINI, MAXI FORMAT (I4,2I10)
IRDTP,IPTOPT, ISRTOPT FORMAT(3I4)
7a
6a
7b
OVERLAY (3,0) PROG. COUPLE Manages Lower Level overlay
4a
4b
3a
3b
6b
OVERLAY (3,1) PROG. MATCAL Calls S.R. GRIDIT & REGRID
5c
5a
OVERLAY (0,0) PROG. CONTOUR Overlay Exec.
1a
1b
2b
5c
S.R. REGRID Interpolates & Extrapolates to Fill Zero Elements
5b
S.R. GRIDIT Evaluates I Matrix
FRMT FORMAT(8A1C)
-1
IRDTP
+1
TAPE9
ISRT
-1
+1
2a
S.R. TESTI Check I Boundary
Determine M Missing Elements
S.R. CONVERT Scales Program Variables To Graph Size
Evaluate Missing Elements
>0
M
0
Determine Min Max Elevations
IPTOPT
+1
-1
INPUT IX,IY,IZ
INPUT X,Y,Z
2b
9
YES
END OF DATA
NO
OVERLAY (2,0) PROG. SORTIT Orders Data
OVERLAY (1,0) PROG. READIT Read Data and Derive IDIST & Matrix Size. Type X,Y,Z. ICODE=0.
NO
END OF DATA
YES
ICODE=9 IX=0 IY=0 IZ=C IDIST=C
OUTPUT ICODE, IX,IY,IZ, IDIST
ISRTOPT
<
Primary Variable Descending-- Secondary X
≥1
Primary Variable Ascending-- Secondary Y
NOTE: X Ascends Y Decends
8
S.R. CFICINT Mean Duplicate Records
99>
ICODE
99
Compute & Store Each Element of I Matrix
Compute & Store All Gradients
5b
ISRT
+1
-1
TAPE8
TAPE7
ICODE=99
9
ICODE
99
1b
TAPE7 OUTPUT ICODE, IX,IY,IZ, IDIST
ISRTOPT
1
Primary Variable IDIST
0
ISRTOPT
2
-1
Primary Variable X
Primary Variable Y
TAPE7
INPUT ICODE, IX,IY,IZ, IDIST
8
TAPE8
INPUT ICODE, IX,IY,IZ, IDIST
9
KK=3 JC=3 INDEX=1
Transfer 2 Rows of I
0≥
KK-MAT
KK=KK+2 JC=3
Pos
JC-NX
INDEX+1 JC=JC+2
4<
INDEX
1>
4≥
7b
1≤
C
0<
KK
INDEX= INDEX+1
6b
6a
OVERLAY(3,3) PROG. CKZFIT Compares I With Data
S.R. SUBI Evaluate Frequency Dist. for Diff:
9
ICODE
8
0
S.R. LINE1
S.R. LINE2
S.R. LINE3
S.R. LINE4
S.R. LINE5
S.R. PLOT
S.R. COLUMN
OVERLAY (3,2) PROG. AUTOCON Plots Contours
7a
Plot Curve
ARG (ICNT,1) =IZ ARG (ICNT,2) =ANS
Call DISCOT For Ans ICNT= ICNT+1
Compute, plot, and label contour line segments
CalComp Modular Routines
CALTPE Store Plot Instructions
KK=0 Transfer 1 Row of I

Figure 4.- Generalized digital contouring program.

Program CONTOUR

CONTOUR, the program executive, has three main functions. First, it reads the program options and constants. Second, it supervises the execution of each of the three lower level overlays. Third, storage for each array and program constant, which is generally common to each overlay, is set aside.

```
  OVERLAY (LINK,0,0)
  PROGRAM CONTOUR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE8,
 1TAPE9)
  INTEGER FRMT
  DIMENSION FRMT(8)
  DIMENSION             KORE(4),KORN(4)
  REAL            MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
  COMMON /BLK2/ KORE,KORN,IFYP,SCLE
  COMMON /BLK3/ KM,L1
  COMMON /BLK4/ DIST,TANG,M
  COMMON /BLK5/ MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
  COMMON /BLK6/ SP,SPP,LAST,DISPGRD, LINEUNT
  COMMON /BLK7/ N,NN,PX1,PY1,XMAX,YMAX,XMAY,YMAY,CI
  COMMON/BLK14/ UNIT,CNVTOIN
  COMMON /BLK15/ IROTP,IPTOPT,ISRTOPT
  COMMON /BLK30/ ISRT
  COMMON /BLK40/ KMAX,KMAX1,IDMX
  COMMON /LIMITS/ MINI,MAXI,IADDI
  COMMON /LINKOC/ LINK,RECALL
  RECALL=6LRECALL
  LINK=4LLINK
3 FORMAT(8A10)
  READ (5,3) FRMT
  READ(5,2) KMAX,KMAX1,IDMX
2 FORMAT (2I4,I10)
  READ (5,FRMT) DIST,UNIT,CNVTOIN ,SPP,CI,SCLE,DISPGRD
 1,TANG
  READ (5,4) ISRT,MINI,MAXI
4 FORMAT(I4,2I10)
  LINEUNT=DISPGRD
  IFYP=1$LAST=2
  READ (5,1) IROTP,IPTOPT,ISRTOPT
1 FORMAT(3I4)
  CALL OVERLAY(LINK,1,0,RECALL)
  IF (ISRT.EQ.1) CALL OVERLAY(LINK,2,0,RECALL)
  CALL OVERLAY(LINK,3,0,RECALL)
  STOP
  END
```

Program READIT

READIT is the first overlay which is subordinate to CONTOUR. READIT is responsible for reading the X- and Y-position and scalar variation (Z-component) for each control point. As pointed out, the data may be read from tape or cards and may be either fixed or floating point. READIT applies the appropriate conversion to the input data. Further, the NW, NE, SW, and SE coordinates of the rectangle which will include all the control points, as well as IDIST (the distance of each control point from an arbitrary origin), are determined by READIT. Each control point IX, IY, and IZ is then written on the appropriate file for use later along with IDIST and a code ICODE which serves to warn subsequent routines when the data have almost expired.

```
      OVERLAY (LINK,1,0)
      PROGRAM READIT
      INTEGER FRMT
      DIMENSION FRMT(8)
      DIMENSION            KORE(4),KORN(4),WRD(3,4)
      REAL MAXWRD2,MINWRD2,MAXWRD3,MINWRD3
      COMMON /BLK2/ KORE,KORN,IFYP,SCLE
      COMMON /BLK4/ DIST
      COMMON /BLK5/ MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
      COMMON /BLK15/ IRDTP,IPTOPT,ISRTOPT
      COMMON /BLK30/ ISRT
      COMMON /LIMITS/ MINI,MAXI,IADDI
      COMMON /LINKDC/ LINK,RECALL
    1 FORMAT (8A10)
      REWIND 7
      ALPHA=(-11.8/180.0)*(3.141592654)
      ICNT=0
      ICODE=0
      IF(IRDTP.NE.1) READ(5,1) FRMT
  100 CONTINUE
      IF (IRDTP.EQ.1) GO TO 1000
      IF (IPTOPT.GE.1) READ(5,FRMT) IX,IY,IZ
      IF (IPTOPT.LT.1) READ(5,FRMT) X,Y,Z
      IF (EOF,5) 1002,1001
 1000 IF (IPTOPT.GE.1) CALL RECIN (9,1,KK,IX,IY,IZ)
      IF (IPTOPT.LT.1) CALL RECIN (9,1,KK,X,Y,Z)
      IF (EOF,9) 1002,1001
 1001 CONTINUE
      IF (MINI.EQ.0.AND.MAXI.EQ.0) 104,105
  105 CONTINUE
      IF ( Z.LT.MINI.OR. Z.GT.MAXI) 100,104
  104 CONTINUE
      IF (IPTOPT.LT.1) GO TO 1003
      X=FLOAT(IX)
      Y=FLOAT(IY)
      Z=FLOAT(IZ)
 1003 CONTINUE
      ICNT=ICNT+1
      IF (ICNT.EQ.1) 300,400
  300 CONTINUE
       MINWRD2=MAXWRD2=Y
       MINWRD3=MAXWRD3=X
  400 CONTINUE
      IF (MINWRD2.GT.Y) MINWRD2=Y
      IF (MINWRD3.GT.X) MINWRD3=X
      IF (MAXWRD2.LT.Y) MAXWRD2=Y
      IF (MAXWRD3.LT.X) MAXWRD3=X
      IX=IFIX(X)$IY=IFIX(Y)$IZ=IFIX(Z)
      X=X-1.0E20
      Y=Y-1.0E20
      IF (ISRTOPT.LT.1) IDIST=IFIX(Y/DIST)
      IF (ISRTOPT.GE.1) IDIST=IFIX(X/DIST)
      CALL RECOUT (8,1,0,ICODE,IX,IY,IZ,IDIST)
      IF (ISRT.LT.1) CALL RECOUT (7,1,0,ICODE,IX,IY,IZ,IDIST)
      GO TO 100
 1002 CONTINUE
      KORE(1)=KORE(4)=MINWRD3
      KORE(2)=KORE(3)=MAXWRD3
      KORN(1)=KORN(2)=MAXWRD2
      KORN(3)=KORN(4)=MINWRD2
      ICODE=9
      IDIST=0
      IX=IY=IZ=0
      IF (ISRT.LT.1) CALL RECOUT (7,1,0,ICODE,IX,IY,IZ,IDIST)
      CALL RECOUT (8,1,0,ICODE,IX,IY,IZ,IDIST)
      ICODE=99
      IF (ISRT.LT.1) CALL RECOUT (7,1,0,ICODE,IX,IY,IZ,IDIST)
      CALL RECOUT (8,1,0,ICODE,IX,IY,IZ,IDIST)
      REWIND 8
      REWIND 7
      RETURN
      END
```

Program SORTIT

SORTIT is an optional overlay which is responsible for setting up the appropriate input to SORT2, which in turn calls the SORT/MERGE routines of the computer system. These routines are very flexible and will utilize all the storage allotted to them.

Regardless of which sorting option is chosen (fig. 4), ICODE is the primary variable. In this manner, the codes 9 and 99 are retained at the end of the data file. This fact in no way changes the indicated flow in the flow chart.

```
      OVERLAY (LINK,2,0)
      PROGRAM SORTIT
      COMMON /BLK15/ IRDTP,IPTOPT,ISRTOPT
      COMMON /LINKDC/ LINK,RECALL
      DIMENSION ISM(5),IFN(2),KEY(16)
      REWIND 7
      REWIND 8
      ISM(1)=1
      ISM(2)=4
      ISM(3)=60
      ISM(4)=1HF
      ISM(5)=1HB
      IFN(1)=5LTAPE7
      IFN(2)=5LTAPE8
      KEY(1)=1HA
      KEY(2)=1HX
      KEY(3)=2
      KEY(6)=1HX
      KEY(10)=1HX
      KEY(13)=1HA
      KEY(14)=1HX
      KEY(15)=5
      IF (ISRTOPT.GE.1) GO TO 1000
      KEY(5)=1HD
      KEY(7)=6
      IF (ISRTOPT.EQ.-1) KEY(7)=4
      KEY(9)=1HA
      KEY(11)=3
      GO TO 1001
1000  CONTINUE
      KEY(5)=1HA
      KEY(7)=6
      IF (IS RTOPT.EQ.2) KEY(7)=3
      KEY(9)=1HD
      KEY(11)=4
1001  CONTINUE
      CALL SORT2(ISM,IFN,KEY)
      END FILE 7
      RETURN
      END
```

Program COUPLE

COUPLE is the last of the three major overlays subordinate to CONTOUR. COUPLE is assigned the function of coupling the three lower level overlays. Further, storage for the depth matrix is set aside.

```
      OVERLAY (LINK,3,0)
      PROGRAM COUPLE
      DIMENSION MAT(60,60)
      COMMON /BLK1/ MAT
      COMMON /LINKDC/ LINK,RECALL
      DO 1 I=1,60
      DO 1 J=1,60
    1 MAT(I,J)=0
      CALL OVERLAY(LINK,3,1,RECALL)
      CALL OVERLAY (LINK,3,3,RECALL)
      CALL OVERLAY(LINK,3,2,RECALL)
      RETURN
      END
```

Program MATCAL

MATCAL furnishes a link between the subroutines GRIDIT and REGRID.

```
      OVERLAY (LINK,3,1)
      PROGRAM MATCAL
      COMMON /LINKDC/ LINK,RECALL
      REWIND 7
      CALL GRIDIT
      CALL REGRID
      RETURN
      END
```

Subroutine GRIDIT

GRIDIT is the first of several subroutines which have the responsibility for generating an adequate depth matrix. The indicated flow between connectors 5a and 5b (fig. 4) is descriptive of its functions.

This routine is the largest consumer of computer time. The block in which the gradients are computed first searches the data for control points which are adequately distributed about the control point for which the gradient is desired. In the process of searching the data, no control point which has a relative difference in elevation sufficiently large to yield a slope greater than TANG (an input variable) is accepted. Further, IDMX is the maximum scalar difference allowed for the product of the distance R between the control points and TANG. Thus, it is possible for all the control points to be considered several times. To reduce this undesirable consumption of time, DIST can be increased or KMAX can be decreased. If the time problem is not solved by these actions, the data should be examined carefully.

Each evaluated element of the depth matrix is made positive by adding a multiple of 10 000 to it. This constant is denoted as IADDI and is computed from the minimum scalar variation MINI.

```
      SUBROUTINE GRIDIT
      DIMENSION RESULT (2)
      DIMENSION IK(100)
      DIMENSION RS(100)
      DIMENSION AA(3,3),BB(3,1),A(1000),B(1000),C(1000),D(3,1000),BSAVE(
     13,100),DSAVE(3),IPIVOT(3),INDEX(3,2)
      DIMENSION E(3)
      DIMENSION            MAT(60,60),KORN(4),KORE(4)
      DIMENSION GRAD(3),GRADCO(3,3)
      INTEGER S
      REAL MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
      COMMON /BLK1/ MAT
      COMMON /BLK2/ KORE,KORN,IFYP,SCLE
      COMMON /BLK3/ JJ,KK
      COMMON /BLK4/ DIST,TANG
      COMMON /BLK5/ MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
      COMMON /BLK6/ SP,SPP,LAST,DISPGRD
      COMMON /BLK40/ KMAX,KMAX1,IDMX
      COMMON /LIMITS/ MINI,MAXI,IADDI
 2000 FORMAT (7X*CORNER COORDINATES OF AREA TO BE CONTOURED*/)
 2002 FORMAT (* NORTH WEST CORNER--X=*E16.8*  Y=* E15.8 )
 2003 FORMAT (* SOUTH EAST CORNER--X=*E16.8*  Y=* E16.8)
 2004 FORMAT (1X, *Y DIMENSION OF DEPTH MATRIX IS*,F15.5,*INCHES*)
 2005 FORMAT (1X, *X DIMENSION OF DEPTH MATRIX IS*,F15.5,*INCHES*/)
 2006 FORMAT (1X,I4,*ELEMENTS IN Y DIRECTION OF DEPTH MATRIX*)
 2007 FORMAT (1X,I4,*ELEMENTS IN X DIRECTION OF DEPTH MATRIX*/)
      ICNT=0
      RSAVE=DIST
      IDEPSUM=0
      PRINT 243
  240 FORMAT(* DEPTH MATRIX SIZE*/)
  241 FORMAT(/)
  242 FORMAT(* THUS,*)
      LABL=0
      CSLOPE1=1.0/SQRT(1.0+TANG**2)
      CSLOPE=0.9
      IF (CSLOPE.GT.CSLOPE1) CSLOPE=CSLOPE1
  243 FORMAT(1H1,* THE FOLLOWING IS INFORMATION PERTAINING TO THE DEPTH
     1MATRIX--SIZE,DIMENSIONS,ETC.*///)
      MINMAX=0
      IF (MINI.NE.0.OR.MAXI.NE.0) MINMAX=1
      IF (MINMAX.EQ.0) 200,201
  200 MINI=200000
      MAXI=-200000
  201 CONTINUE
      SP=DIST
      IPTOK=0
      CALL CONVERT(SP)
      PRINT 2000
      PRINT 2002 ,MINWRD3,MAXWRD2
      PRINT 2003,MAXWRD3,MINWRD2
      CALL CONVERT (DISPGRD)
    8 FORMAT (F5.2,F10.4,2I5,I5)
   10 FORMAT (2I10)
      IF (IFYP) 4,3,4
    3 IY=KORN(3)-KORN(1)
      GO TO 2
    4 IY=KORN(1)-KORN(3)
    2 YMAX=FLOAT(IY)
      CALL CONVERT(YMAX)
      IX=KORE(3)-KORE(1)
      XMAX=FLOAT(IX)
      CALL CONVERT(XMAX)
      JJ=IFIX(YMAX/SP)+1
      KK=IFIX(XMAX/SP)+1
      IF (XMAX-FLOAT(KK-1)*SP.GT.0.0) KK=KK+1
      IF (YMAX-FLOAT(JJ-1)*SP.GT.0.0) JJ=JJ+1
      IF (KK.LT.3.OR.JJ.LT.3) 301,302
  302 CONTINUE
      IF (KK.GT.60.OR.JJ.GT.60) 230,231
  301 PRINT 300
      GO TO 304
  230 PRINT 232
  304 CONTINUE
      PRINT 2008
      PRINT 2004,YMAX
      PRINT 2005,XMAX
      PRINT 240
      PRINT 2006,JJ
      PRINT 2007,KK
  232 FORMAT(* SINCE  DIST IS TOO SMALL THE MAXIMUM DIMENSIONS OF THE DE
     1PTH MATRIX ARE EXCEEDED--I(60,60)*)
  300 FORMAT(* SINCE  DIST IS TOO LARGE THE MINIMUM DIMENSIONS OF THE DE
     1PTH MATRIX VIOLATED--I(3,3)*)
      GO TO 215
  231 CONTINUE
 2008 FORMAT(//* DIMENSIONS OF REQUIRED PLOTTING SURFACE EXCLUSIVE OF GR
     1ID*/)
      PRINT 2008
      DO 180 J=1,JJ
      DO 180 K=1,KK
      MAT(J,K)=0
  180 CONTINUE
      K=1
      PRINT 2004,YMAX
      PRINT 2005,XMAX
      PRINT 2006,JJ
      PRINT 2007,KK
      TEST=1.0E+20
      S=-2
      DO 740 I=1,1000
      DO 740 J=1,3
  740 D(J,I)=TEST
  104 CALL RECIN(7,1,KKK,ICODE,IX,IY,IZ,IDIST)
      IF (IFYP) 40,30,40
   30 IY=(-1)*IY
   40 CONTINUE
      IF (ICODE.EQ.99) GO TO 221
      IF (ICNT.EQ.1000.AND.ICODE.LT.9) 202,203
  202 PRINT 211
      GO TO 215
  211 FORMAT(* YOU HAVE OVER 1000 CONTROL POINTS.*/* ADDITIONAL CONTROL
     1POINTS CAN NOT BE ACCEPTED.*/* STORED DATA MAY NOT ADEQUATELY REPR
     2ESENT AREA TO BE CONTOURED.*)
  203 CONTINUE
      IF (LABL.EQ.0) 204,205
  204 IF (MINMAX.EQ.0) 212,213
  212 PRINT 206
      LABL=1
      PRINT 242
  206 FORMAT(* YOU HAVE ELECTED TO REJECT NO CONTROL POINTS*)
  207 FORMAT(* YOU HAVE ELECTED TO REJECT ALL CONTROL POINTS WITH SCALAR
     1 VARIATIONS LESS THAN*I6* AND GREATER THAN*I6)
  208 FORMAT(* YOU HAVE A MINIMUM SCALAR VARIATION OF*I10)
  209 FORMAT(* YOU HAVE A MAXIMUM SCALAR VARIATION OF*I10)
      GO TO 210
  213 PRINT 207,MINI,MAXI
      LABL=1
      PRINT 242
      IMINI=200000$IMAXI=-200000
  210 CONTINUE
  214 FORMAT(* MUST TERMINATE PROGRAM EXECUTION WITH A MODE 4 FATAL ERRO
     1R*)
      GO TO 216
  215 CONTINUE
      PRINT 214
      AZERO=0.0
      AX=BX/AZERO
      AY=AX*AX
  216 CONTINUE
  205 CONTINUE
      IF (ICODE.EQ.9) 219,220
  219 IF (MINMAX.NE.0) PRINT 208,IMINI
      IF (MINMAX.NE.0) PRINT 209, IMAXI
      IF (MINMAX.EQ.0) PRINT 208,MINI
      IF (MINMAX.EQ.0) PRINT 209,MAXI
      IADDI=(-1)*(10000)*(MINI/10000)+10000
      MINI=MINI+IADDI$MAXI=MAXI+IADDI
      GO TO 221
  220 IF (MINMAX.EQ.0) 217,218
  217 IF (IZ.LT.MINI) MINI=IZ
      IF (IZ.GT.MAXI) MAXI=IZ
      GO TO 221
  218 CONTINUE
      IF (IZ.LT.IMINI) IMINI=IZ
      IF (IZ.GT.IMAXI) IMAXI=IZ
  221 CONTINUE
      B(1)=FLOAT(IX)
      B(2)=FLOAT(IY)
      IJUMP=-1
      IF (ICODE-9) 100,1133,105
  100 CONTINUE
 1133 CALL CKPOINT (B(1),B(2),IZ,IPTOK,ICODE)
      IF (IPTOK.EQ.-1) GO TO 104
  103 CONTINUE
      ICNT=ICNT+1
 1103 D(1,ICNT)=B(1)
      D(2,ICNT)=B(2)
      B(3)=FLOAT(IZ)
      D(3,ICNT)=B(3)
      IF (ICODE.LT.9) GO TO 104
  105 CONTINUE
      DO 700 J=1,ICNT
      IF (D(1,J).EQ.TEST) GO TO 700
      DO 750 II=1,3
      GRAD(II)=0.0
      BB(II,1)=0.0
      DO 750 IJ=1,3
      AA(II,IJ)=0.0
  750 CONTINUE
      DXMX=DYMX=-1000.0
      DXMN=DYMN=+1000.0
      DO 705 N2=1,3
      GRADCO(N2,1)=D(N2,J)
  705 DSAVE(N2)=D(N2,J)
      RSAVE=0.0
      RSUM=0.0$RMAX=0.0
      INUM=0
      JNUM=0
      DO 771 ID=1,100
      RLAST=RSAVE
      RSAVE=RSAVE+DIST/(10.0)
      DO 701 I=1,ICNT
      R=0.0
      IF (INUM.EQ.100) GO TO 701
      IF (D(1,I).EQ.TEST.OR.I.EQ.J) GO TO 701
      DO 703 N1=1,2
      IF(N1.EQ.1)DX=DSAVE(N1)-D(N1,I)
      IF(N1.EQ.2)DY=DSAVE(N1)-D(N1,I)
  703 R=R+(DSAVE(N1)-D(N1,I))**2
      R=SQRT(R)
      IF (R.GT.RSAVE) GO TO 701
      IF (R.LE.RLAST) GO TO 701
      IF (ABS((DSAVE(3)-D(3,I))          ) /R.GT.TANG.OR.ABS(DSAVE(3)-D(3
     1,I)).GT.IDMX) GO TO 701
      KADD=-1
      IF (INUM.GE.KMAX1) 773,7774
  773 DO 775 I10=1,INUM
      IF (RMAX.EQ.RS(I10)) 774,775
  774 CONTINUE
      KADD=KADD+1
      IF (KADD.GT.0) GO TO 7774
  775 CONTINUE
      GO TO 772
 7774 CONTINUE
      IF (R.GT.RMAX) RMAX=R
      INUM=INUM+1
      RS(INUM)=R
      DO 707 N4=1,3
      DXMX=AMAX1(DXMX,DX)
      DYMX=AMAX1(DYMX,DY)
      DXMN=AMIN1(DXMN,DX)
      DYMN=AMIN1(DYMN,DY)
  707 BSAVE(N4,INUM)=D(N4,I)
  701 CONTINUE
  771 CONTINUE
  772 CONTINUE
      IF (INUM.LT.KMAX1) GO TO 751
      DO 712 I2=1,INUM
  712 RSUM=RSUM+(1.0-RS(I2   )/RMAX)**2*(RS(I2) /RMAX)**S
      RWSUM=RSUM
      INUM1=INUM-1
      DO 710 I1=1,INUM1,1
      NNUM=I1
      JNUM=I1+1$SUMSQ1=0.0$SUMSQ2=0.0
      DO  710 I6=JNUM,INUM,1
      DO 770 J6=1,3
      GRADCO(J6,2)=BSAVE(J6,I1)
      GRADCO(J6,3)=BSAVE(J6,I6)
      IF (J6.EQ.3) GO TO 770
      SUMSQ1=(GRADCO(J6,2)-GRADCO(J6,1))**2
      SUMSQ2=(GRADCO(J6,3)-GRADCO(J6,1))**2
  770 CONTINUE
      IF (SUMSQ1.EQ.0.0) GO TO 710
      IF (SUMSQ2.EQ.0.0) GO TO 710
      RTSMSQ1=SQRT(SUMSQ1)
      RTSMSQ2=SQRT(SUMSQ2)
      SALPHA1=ABS(GRADCO(2,2)/RTSMSQ1)
      SALPHA2=ABS(GRADCO(2,3)/RTSMSQ2)
      CALPHA1=ABS(GRADCO(1,2)/RTSMSQ1)
      CALPHA2=ABS(GRADCO(1,3)/RTSMSQ2)
      SDIFALP=SALPHA2*CALPHA1-CALPHA2*SALPHA1
      IF (ABS(SDIFALP   )-00.17365.LT.0.0) GO TO 710
      DG=GRADCO(1,1)*(GRADCO(2,2)*GRADCO(3,3)-GRADCO(3,2)*GRADCO(2,3))
     1-GRADCO(2,1)*(GRADCO(1,2)*GRADCO(3,3)-GRADCO(3,2)*GRADCO(1,3))
     2+GRADCO(3,1)*(GRADCO(1,2)*GRADCO(2,3)-GRADCO(2,2)*GRADCO(1,3))
      IF(DG.EQ.0.0)GO TO 710
      GRAD(1)=(GRADCO(2,2)*GRADCO(3,3)-GRADCO(3,2)*GRADCO(2,3)-GRADCO(2,
     11)*(GRADCO(3,3)-GRADCO(3,2))+GRADCO(3,1)*(GRADCO(2,3)-GRADCO(2,2))
     2)/DG
      GRAD(2)=(GRADCO(1,1)*(GRADCO(3,3)-GRADCO(3,2))-(GRADCO(3,3)*GRADCO
     1(1,2)-GRADCO(3,2)*GRADCO(1,3))+(GRADCO(3,1)*(GRADCO(1,2)-GRADCO(1,
     23))))/DG
      GRAD(3)=(GRADCO(1,1)*(GRADCO(2,2)-GRADCO(2,3))-GRADCO(2,1)*(GRADCO
     1(1,2)-GRADCO(1,3))+GRADCO(1,2)*GRADCO(2,3)-GRADCO(2,2)*GRADCO(1,3)
     2)/DG
      RW=SQRT((DSAVE(1)-BSAVE(1,I1))**2+(DSAVE(2)-BSAVE(2,I1))**2)/RMAX
      W=SQRT((1.0-RW)**2*RW**S/RWSUM)
      E(1)=W*(BSAVE(1,I1)-DSAVE(1))
      E(2)=W*(BSAVE(2,I1)-DSAVE(2))
      E(3)=W*(BSAVE(3,I1)-DSAVE(3))
      DO 711 J1=1,3
      BB(J1,1)=BB(J1,1)-E(J1)*(E(1)*GRAD(1)+E(2)*GRAD(2)+E(3)*GRAD(3))
     1*(-1.0)
      DO 711 J2=1,3
      AA(J1,J2)=AA(J1,J2)+E(J1)*E(J2)
  711 CONTINUE
  710 CONTINUE
      IF (JNUM.EQ.1) GO TO 780
      CALL MATINV(AA,3,BB,1,DETERM,IPIVOT,INDEX,3,ISCALE)
      DO 1700 IA=1,3
      IV=LEGVAR(BB(IA,1))
      IF (IV) 751,1700,751
 1700 CONTINUE
      IF (DETERM.EQ.0.0) GO TO 751
      DC=BB(3,1)
      A(J)=BB(1,1)
      B(J)=BB(2,1)
      IF((DC.EQ.0.0).AND.(A(J).EQ.0.0).AND.(B(J).EQ.0.0)) GO TO 751
      BB(3,1)=DC/SQRT(DC**2+A(J)**2+B(J)**2)
      GO TO 781
  780 CONTINUE
      A(J)=GRAD(1)
      B(J)=GRAD(2)
      DC=GRAD(3)
      GO TO 751
  781 CONTINUE
      TXY=(DYMX-DYMN)/(DXMX-DXMN)
      BB(3,1)=ABS(BB(3,1))
      IF (BB(3,1).LT.CSLOPE) GO TO 751
      A(J)=-A(J)/DC
      B(J)=-B(J)/DC
      C(J)=-(A(J)*DSAVE(1)+B(J)*DSAVE(2)-DSAVE(3))
      GO TO 700
  751 A(J)=TEST
      B(J)=TEST
      C(J)=TEST
  700 CONTINUE
      DO 720 I=1,KK
      X=KORE(1)+DIST*FLOAT(I-1)
      DO 720 J=1,JJ
      Y=KORN(1)-DIST*FLOAT(J-1)
      RMAX=0.0
      RSAVE=0.0
      INUM=0
      DO 742 L=1,100
      RLAST=RSAVE
      RSAVE=DIST*FLOAT(L)/(10.0)
      DO 741 II=1,ICNT
      KADD=-1
      IF (INUM.GE.KMAX) 776,7777
  776 DO 778 I3=1,INUM
      IF (SQRT((D(1,II)-X)**2+(D(2,II)-Y)**2).EQ.RMAX) 777,778
  777 CONTINUE
      KADD=KADD+1
      IF (KADD.GT.0) GO TO 7777
  778 CONTINUE
      GO TO 744
 7777 CONTINUE
      IF (A(II).EQ.TEST.AND.B(II).EQ.TEST) GO TO 741
      DR=SQRT((D(1,II)-X)**2+(D(2,II)-Y)**2)
      IF (DR.GT.RSAVE) GO TO 741
      IF (DR.LE.RLAST) GO TO 741
      IF (DR.GT.RMAX) RMAX=DR
      INUM=INUM+1
      IK(INUM)=II
  741 CONTINUE
  742 CONTINUE
  744 CONTINUE
      IF(INUM.EQ.0)GO TO 720
      RD=0.0
      DO 730 I1=1,INUM
      I4=IK(I1)
      R=SQRT((X-D(1,I4))**2+(Y-D(2,I4))**2)/RMAX
  730 RD=RD+(1.0-R)**2*R**S
      IF(RD.EQ.0.0)GO TO 720
      DO 731 I2=1,INUM
      I4=IK(I2)
      RS=SQRT((X-D(1,I4))**2+(Y-D(2,I4))**2)
      MAT(J,I)=MAT(J,I)+(1.0-RS/RMAX)**2*(RS/RMAX)**S/RD*(C(I4)+A(I4)*X+
     1B(I4)*Y)
  731 CONTINUE
      MAT(J,I)=MAT(J,I)+IADDI
  720 CONTINUE
 3333 RETURN
      END
```

Subroutine REGRID

REGRID first scans the depth matrix generated by GRIDIT to evaluate its maximum and minimum values (IMAX and IMIN) as well as the number of zero (unfilled) elements therein. If a border column or row contains all zeros, the matrix dimensions are reduced by TESTI.

Each zero element is filled and checked by REGRID. Further, the slope between each pair of adjacent elements in the matrix is checked. If the slope exceeds either TANG or IMAX or is less than IMIN, the element is first rejected and then evaluated again. It is possible for this routine to reject the entire depth matrix and terminate the job.

```
      SUBROUTINE REGRID
      INTEGER AGAIN
      DIMENSION I(60,60)
      COMMON /BLK1/ I
      COMMON /BLK3/ KM,LM
      COMMON /BLK4/ DIST,TANG,M
      COMMON /LIMITS/ MINI,MAXI,IADDI
  600 FORMAT (* MINIMUM ELEVATION PRIOR TO SMOOTHING = *I10)
  601 FORMAT (* MAXIMUM ELEVATION PRIOR TO SMOOTHING = *I10,/)
  602 FORMAT(/,* MINIMUM ELEVATION AFTER INTERPOLATION ETC = *,I10,/)
  603  FORMAT (* MAXIMUM EXPECTED SLOPE BETWEEN TWO CONSECUTIVE GRID POI
     1NTS  =  *,F15.5)
  604 FORMAT (20I6)
  605 FORMAT (10I5)
  607 FORMAT (* UNSMOOTHED DEPTH MATRIX* //)
  606 FORMAT (* SMOOTHED DEPTH MATRIX*//)
   25 FORMAT (I10,I10)
    6 FORMAT (I3,F10.0,F10.5)
    5 FORMAT (20I4)
      IF (       MINI.LT.1000) 300,302
  300 PRINT 301
  301 FORMAT(* YOU ARE IN SUBROUTINE REGRID WITH A MINIMUM SCALAR VARIAT
     1ION WHICH IS TOO SMALL*/* CHECK  LAST MINIMUM SCALAR VARIATION IF
     2PROGRAM FAILS TO EXECUTE PROPERLY*)
  302 CONTINUE
  310 FORMAT(1H1,* THE FOLLOWING INFORMATION REGARDING YOUR UNSMOOTHED D
     1EPTH MATRIX IS FURNISHED*///)
  311 FORMAT(/)
  312 FORMAT(1H1,* THE FOLLOWING INFORMATION REGARDING THE SMOOTHED DEPT
     1H MATRIX IS FURNISHED*///)
      PRINT 310
      CALL TESTI
      PRINT 311
      M=1
      MCNT=0
      PRINT 607
      DO 711 K=1,KM
      PRINT 604,(I(K,J),J=1,LM)
      DO 711 L=1,LM
      IF (I(K,L).LT.MINI.OR.I(K,L).GT.MAXI) I(K,L)=0
      IF (I(K,L).EQ.0) MCNT=MCNT+1
      IF (I(K,L).NE.0) 800,801
  800 IF (MCNT.GT.M) M=MCNT
      MCNT=0
  801 CONTINUE
  711 CONTINUE
      PRINT 312
      IF (M.EQ.0) PRINT 313
  313 FORMAT(* IT HAS BEEN DETERMINED THAT EACH ELEMENT OF YOUR UNSMOOTH
     1ED DEPTH MATRIX IS FILLED*/* NO ELEMENT WAS FOUND TO EXCEED THE LI
     2MITS ESTABLISHED BY YOU*/)
      IF (M.NE.0) PRINT 314
  314 FORMAT(* IT HAS BEEN DETERMINED THAT EACH ELEMENT OF YOUR UNSMOOTH
     1ED DEPTH MATRIX IS INADEQUATELY FILLED*/)
      CALL TESTI
      PRINT 311
      JCNT=0
      ICHEAT=0
C     THIS SECTION FINDS MIN AND MAX DEPTHS
      IMIN=20000
      IMAX=-10000
      DO 710 K=1,KM
      DO 710 L=1,LM
      IF (I(K,L).GT.IMAX) 701,702
  701 IMAX=I(K,L)
  702 IF (I(K,L).LT.IMIN.AND.I(K,L).GT.0) 703,710
  703 IMIN=I(K,L)
  710 CONTINUE
      MAXL=IMAX-IADDI
      MINL=IMIN-IADDI
      PRINT 315
  315 FORMAT(* REGRID WILL NOT PERMIT THE INTERNAL ELEMENTS OF I TO EXCE
     1ED THE FOLLOWING LIMITS*/)
      PRINT 600,MINL
      PRINT 601,MAXL
  700 ICNT=0
      AGAIN=500.0
      K1=KM-1
      L1=LM-1
      IF (M.EQ.0) 1001,1
    1 CONTINUE
C     THIS SECTION LINEARLY INTERPOLATES TO FILL MISSING INTERIOR GRID P
      DO 20 K=2,K1
      DO 20 L=2,L1
      IF (I(K,L).EQ.0) 10,11
   11 I(K,L)=I(K,L)
      GO TO 20
   10 IF (I(K-1,L).EQ.0.OR.I(K+1,L).EQ.0) 12,13
   13 I(K,L)=(I(K-1,L)+I(K+1,L))/2
      GO TO 20
   12  IF (I(K,L-1).EQ.0.OR.I(K,L+1).EQ.0) 14,15
   15 I(K,L)=(I(K,L-1)+I(K,L+1))/2
      GO TO 20
   14 IF (I(K+1,L).EQ.0.OR.I(K,L+1).EQ.0)  8,17
   17 I(K,L)=(I(K+1,L)+I(K,L+1))/2
      GO TO 20
    8 IF (I(K-1,L).EQ.0.OR.I(K,L-1).EQ.0) 16,7
    7 I(K,L)=(I(K-1,L)+I(K,L-1))/2
      GO TO 20
   16 IF (I(K,L+1).EQ.0.OR.I(K-1,L).EQ.0) 18,19
   19 I(K,L) =(I(K,L+1)+I(K-1,L))/2
      GO TO 20
   18 IF(I(K+1,L).EQ.0.OR.I(K,L-1).EQ.0)20,21
   21 I(K,L)=(I(K+1,L)+I(K,L-1))/2
   20 CONTINUE
      ICNT=ICNT+1
      IF (ICNT.EQ.M) 23,1
   23 CONTINUE
C     FILL BLANKS I(1,L) BY LINEAR INTERPOLATION
      DO 90 L=2,L1
      DO 90 K=1,1
      IF (I(K,L).EQ.0) 91,90
   91 IF (I(K,L+1).EQ.0.OR.I(K,L-1).EQ.0) 90,92
   92 I(K,L)=(I(K,L+1)+I(K,L-1))/2
   90 CONTINUE
C     FILL BLANKS I(1,L) BY EXTRAPOLATION
      DO 50 L=1,LM
      DO 50 K=1,1
      IF (I(K,L).EQ.0.AND.(I(K+1,L).NE.0.AND.I(K+2,L).NE.0)) 51,50
   51 J=I(K+1,L)-I(K+2,L)
      I (K,L)=I(K+1,L)+J
   50 CONTINUE
C     FILL BLANKS I(K,1) BY LINEAR INTERPOLATION
      DO 30 K=2,K1
      DO 30 L=1,1
      IF (I(K,L).EQ.0) 31,30
   31 IF (I(K+1,L).EQ.0.OR.I(K-1,L).EQ.0) 30,32
   32 I(K,L)=(I(K+1,L)+I(K-1,L))/2
   30 CONTINUE
C     FILL BLANKS I(K,1) BY EXTRAPOLATION
      DO 55K=1,KM
      DO 55 L=1,1
      IF (I(K,L).EQ.0.AND.(I(K,L+1).NE.0.AND.I(K,L+2).NE.0)) 56,55
   56 J=I(K,L+1)-I(K,L+2)
      I(K,L)=I(K,L+1)+J
   55 CONTINUE
C     FILL BLANKS I(KM,L) BY LINEAR INTERPOLATION
      DO 35 K=KM,KM
      DO 35 L=2,L1
      IF (I(K,L).EQ.0) 36,35
   36 IF (I(K,L+1).EQ.0.OR.I(K,L-1).EQ.0) 35,37
   37 I(K,L)=(I(K,L+1)+I(K,L-1))/2
   35 CONTINUE
C     FILL BLANKS I(KM,L) BY EXTRAPOLATION
      DO 60 K=KM,KM
      DO 60 L=1,LM
      IF (I(K,L).EQ.0.AND.(I(K-1,L).NE.0.AND.I(K-2,L).NE.0)) 61,60
   61 J=I(K-1,L)-I(K-2,L)
      I(K,L)=I(K-1,L)+J
   60 CONTINUE
C     FILL BLANKS I(K,LM) BY LINEAR INTERPOLATION
      DO 670 K=2,K1
      DO 670 L=LM,LM
      IF (I(K,L).EQ.0) 671,670
  671 IF (I(K+1,L).EQ.0.OR.I(K-1,L).EQ.0) 670,672
  672 I(K,L)=(I(K+1,L)+I(K-1,L))/2
  670 CONTINUE
C     FILL BLANKS I(K,LM) BY EXTRAPOLATION
      DO 65 K=1,KM
      DO 65 L=LM,LM
      IF (I(K,L).EQ.0.AND.(I(K,L-1).NE.0.AND.I(K,L-2).NE.0)) 66,65
   66 J=I(K,L-1)-I(K,L-2)
      I(K,L)=I(K,L-1)+J
   65 CONTINUE
C     AT THIS POINT ALL BLANKS ARE FILLED
 1001 CONTINUE
      ICNTI=0
      IF (JCNT.GT.0) 999,998
  998 CONTINUE
      DO 997 K=1,KM
      DO 997 L=1,LM
      IF (I(K,L).GT.IMAX.OR.I(K,L).LT.IMIN) 996,997
  996 I(K,L)=0
  350 FORMAT(* BE CAUTIOUS OF ELEMENT I(*I2*,*I2*)*)
      PRINT 350,K,L
      ICNTI=ICNTI+1
  997 CONTINUE
      M=20$JCNT=1
      IF (ICNTI.EQ.0) GO TO 999
      GO TO 700
  999 CONTINUE
    4 FORMAT (2X,20I5)
      IF (I(1,1).GT.I(1,2))76,77
   77 MIN=I(1,1)
      GO TO 78
   76 MIN=I(1,2)
   78 CONTINUE
      DO 75 K=2,K1
      DO 75 L=2,L1
      IF (I(K,L).GT.MIN) 75,80
   80 CONTINUE
      MIN=I(K,L)
   75 CONTINUE
      MINL=MIN-IADDI
      PRINT 602,MINL
      IF (JCNT.EQ.3) GO TO 305
      IF (MIN.GE.1000) 81,303
  303 IF (JCNT.EQ.1) 304,305
  304 PRINT 306
  306 FORMAT(* MINIMUM SCALAR VARIATION IN INTERNAL ELEMENTS OF I MATRIX
     1 IS TOO SMALL*/* WILL TRY TO SMOOTH AND THEN EVALUATE AGAIN*/)
      GO TO 81
  305 PRINT 307
  307 FORMAT(* HAVE TRIED TO EVALUATE DEPTH MATRIX*/* WILL PRINT I MATRI
     1X BEFORE TERMINATING YOUR JOB*/* PLEASE CHECK AND RESUBMIT*//)
      DO 320 K=1,KM
      PRINT 604,(I(K,L),L=1,LM)
  320 CONTINUE
      PRINT 308
  308 FORMAT(* POSSIBLE REASONS FOR FAILURE*//* ABSOLUTE VALUE OF MINIMU
     1M AND MAXIMUM SCALAR VARIATIONS MAY BE TOO LARGE*/* DISTANCE BETWE
     2EN GRID POINTS NEEDS ADJUSTING*/* INSUFFICIENT CONTROL POINT DENSI
     3TY*/*  MAXIMUM SLOPE BETWEEN DATA AND/OR GRID POINTS NEEDS ADJUSTI
     4NG*)
      PRINT 309
  309 FORMAT(* PROGRAM EXECUTION TERMINATED WITH A MODE 4 FATAL ERROR*)
      AZERO=0.0
      AX=BX/AZERO
      AY=AX*AX
   82 CONTINUE
      PRINT 606
      DO 9912 K=1,KM
      PRINT 604,(I(K,L),L=1,LM)
 9912 CONTINUE
      REWIND 8
      RETURN
   81 TAM=TANG
      PRINT 603,TAM
      PRINT 311
      K2=KM-2
      L2=LM-2
      DO 95 K=1,K2
      DO 95 L=1,L2
      KP1=K+1
      KP2=K+2
      LP1=L+1
      LP2=L+2
      T1=ABS((       I(K,L)-I(KP1,L))/DIST)
      IF (TAM-T1) 100,95,95
  100 T2=       (I(KP1,L)-I(KP2,L))/DIST
      T3=ABS(T2)
      IF (TAM-T3) 102,103,103
  103 I(K,L)=I(KP1,L)+(I(KP1,L)-I(KP2,L))
      GO TO 95
  102 T4=ABS(      (I(K,LP1)-I(K,LP2))/DIST)
      IF (TAM-T4) 105,106,106
  106 I(K,L)=I(K,LP1)+(I(K,LP1)-I(K,LP2))
      GO TO 95
  105 I(K,L)=0
   95 CONTINUE
      DO 110 K=1,K2
      DO 110 L=1,L2
      LP2=L+2
      KP2=K+2
      LP1=L+1
      KP1=K+1
      T10=ABS(      (I(K,L)-I(K,LP1))/DIST)
      IF (TAM-T10) 200,110,110
  200 T20=(      I(K,LP1)-I(K,LP2))/DIST
      T30=ABS(T20)
      IF (TAM-T30) 202,203,203
  203 I(K,L)=I(K,LP1)+(I(K,LP1)-I(K,LP2))
      GO TO 110
  202 I(K,L)=0
  110 CONTINUE
      DO 650 K=1,KM
      DO 650 L=1,LM
      IF (I(K,L).GT.IMAX.OR.I(K,L).LT.IMIN) 651,650
  651 I(K,L)=0
  650 CONTINUE
      DO 500 K=1,KM
      DO 500 L=1,LM
      IF (I(K,L).EQ.0) 501,500
  501 AGAIN=0
  500 CONTINUE
      JCNT=JCNT+1
      IF (AGAIN.EQ.0) 700,82
      END
```

Subroutine CKPOINT

CKPOINT is responsible for averaging the scalar variation of any control point which has been observed more than once.

```
      SUBROUTINE CKPOINT (XIN1,YIN1,IDEP1,IPTOK,ICODE)
      IF (ICODE.EQ.9) GO TO 1124
      IF (IPTOK.EQ.0) 103,105
 1123 CONTINUE
      IDEP2=IDEPSUM/IDENOM
      IDEPSUM=0
      IDENOM=1
      GO TO 1133
 1113 IDENOM=IDENOM+1
      IDEPSUM=IDEPSUM+IDEP1
      IPTOK=-1
      GO TO 104
  103 IPTOK=-1
      IDENOM=1
      XIN2=XIN1
      YIN2=YIN1
      IDEP2=IDEP1
      IDEPSUM=IDEP2
      GO TO 104
  105 CONTINUE
      IF (YIN1.EQ.YIN2.AND.XIN1.EQ.XIN2) GO TO 1113
 1124 IF (IDENOM.GT.1) GO TO 1123
 1133 CONTINUE
      TEMP=XIN2
      XIN2=XIN1
      XIN1=TEMP
      TEMP=YIN2
      YIN2=YIN1
      YIN1=TEMP
      TEMP=IDEP2
      IDEP2=IDEP1
      IDEP1=TEMP
      IDEPSUM=IDEP2
      IPTOK=1
      IF(ICODE.EQ.9)IPTOK=0
  104 RETURN
      END
```

Subroutine TESTI

TESTI scans the depth matrix for border rows and columns for which the sum is zero. If this condition occurs, the appropriate dimensions of the matrix are reduced.

```
      SUBROUTINE TESTI
      REAL MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
      DIMENSION IROW(2),ICOL(2)
      COMMON /BLK1/ I(60,60)
      COMMON /BLK3/ KM,LM
      COMMON /BLK5/ MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
      COMMON /BLK4/ DIST
  101 CONTINUE
      DO 100 J=1,2
      IROW(J)=0
      DO 100 K=1,LM
      IF (I(J,K).GT.0) IROW(J)=IROW(J)+1
  100 CONTINUE
      IF (IROW(1)+IROW(2).GT.0) GO TO 302
      DO 300 J=1,KM
      DO 300 L=1,LM
  300 I(J,L)=I(J+1,L)
      KM=KM-1
      MAXWRD2=MAXWRD2-DIST
      GO TO 101
  302 J=KM
      IR=0
      DO 301 L=1,LM
      IF (I(J,L).GT.0) IR=IR+1
  301 CONTINUE
      IF (IR.GT.0) 201,304
  304 KM=KM-1
      MINWRD2=MINWRD2+DIST
      GO TO 302
  201 CONTINUE
      DO 200 K=1,2
      ICOL(K)=0
      DO 200 J=1,KM
      IF (I(J,K).GT.0) ICOL(K)=ICOL(K)+1
  200 CONTINUE
      IF (ICOL(1)+ICOL(2).GT.0) GO TO 402
      DO 400 L=1,LM
      DO 400 J=1,KM
  400 I(J,L)=I(J,L+1)
      LM=LM-1
      MINWRD3=MINWRD3+DIST
      GO TO 201
  402 J=LM
      IR=0
      DO 401 L=1,KM
      IF (I(L,J).GT.0) IR=IR+1
  401 CONTINUE
      IF (IR.GT.0) 501,404
  404 LM=LM-1
      MAXWRD3=MAXWRD3-DIST
      GO TO 402
```

```
501 CONTINUE
    PRINT 1,MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
  1 FORMAT(/,                                                        *
   1ADJUSTED BOUNDARIES OF DEPTH MATRIX*//* MINIMUM Y*E16.8* MAXIMUM Y
   1*E16.8,/,* MINIMUM X*E16.8* MAXIMUM X*E16.8)
    RETURN
    END
```

Subroutine CONVERT

Program variables are scaled to plot dimensions and converted to inches by CONVERT.

```
SUBROUTINE CONVERT(X)
COMMON /BLK2/ KORE(4),KORN(4),IFYP,SCLE
COMMON/BLK14/ UNIT,CNVTOIN
X=X*UNIT/(SCLE*CNVTOIN)
RETURN
END
```

Program CKZFIT

CKZFIT is the second of three overlays which are subordinate to COUPLE. CKZFIT establishes the X- and Y-coordinates of each element within the depth matrix and stores each within the appropriate array table and reorganizes the I matrix into a one-dimensional array. Each of the array tables and each control-point X- and Y-coordinate are part of the argument listed for the CALL statement to DISCOT, a system routine. DISCOT uses linear interpolation to find the scalar variation predicted by the depth matrix at each X and Y control-point position. The real and predicted control-point scalar variations are stored in turn by CKZFIT. Subsequent to reading all the control points, subroutine SUB1 is called.

```
      OVERLAY (LINK,3,3)
      PROGRAM CKZFIT
      DIMENSION XTAB(60),ZTAB(60),YTAB(3600),I(60,60),ARG(1000,2)
      DIMENSION KORE(4),KORN(4)
      DIMENSION YYTAB(60,60)
      EQUIVALENCE (YYTAB,YTAB)
      COMMON /BLK1/ I
      COMMON /BLK2/ KORE,KORN
      COMMON /BLK3/ KM,LM
      COMMON /BLK4/ DIST
      COMMON /LIMITS/ MINI,MAXI,IADDI
      COMMON /LINKDC/ LINK,RECALL
      PRINT 1
    1 FORMAT(1H1,* YOUR SMOOTHED DEPTH MATRIX, I, IS PRINTED ABOVE.*/* A
     1T THIS POINT THE SCALAR VARIATION  PREDICTED BY I AT THE LOCATION
     2OF EACH CONTROL POINT*/* IS COMPARED WITH THE SCALAR MAGNITUDE OF
     3THE CONTROL POINT.*/* A PLOT FOR FREQUENCY OF OCCURRENCE VS DISCRE
     4PANCY FOLLOWS AS PART OF THIS LISTING*//////)
      REWIND 7
      YMAX=FLOAT(KORN(1))-FLOAT(KM-1)*DIST
      XMIN=FLOAT(KORE(1))
      ICNT=0
      I1=0
      DO 100 L=1,LM
      XTAB(L)=XMIN+DIST*(FLOAT(L-1))
      I1=(60)*(L-1)
      I2=I1+KM+1
      DO 100 J=1,KM
      IF (L.EQ.1) ZTAB(J)=YMAX+DIST*FLOAT(J-1)
      I1=I2-J
      YTAB(I1)=I(J,L)
  100 CONTINUE
      LM1=LM+1
      KM1=KM+1
      DO 101 N=KM1,60
      ZTAB(N)=ZTAB(KM)+DIST*(N-KM)
      DO 101 J=1,LM
      K=(J-1)*(60)+KM
  101 YYTAB(N,J)=YTAB(K)
      DO 102 N=1,KM
      K=(LM-1)*(60)+N
      DO 102 J=LM1,60
      XTAB(J)=XTAB(LM)+DIST*(J-LM)
  102 YYTAB(N,J)=YTAB(K)
  200 CONTINUE
      CALL RECIN(7,1,KKK,ICODE,IX,IY,IZ,IDIST)
      IF (EOF,7) 201,202
  202 IF(ICODE.GT.0) GO TO 201
      XA=IX
      ZA=IY
      CALL DISCOT(ZA,XA,ZTAB,YTAB,XTAB,111,3600,60,ANS)
      ANS=ANS-FLOAT(IADDI)
      ICNT=ICNT+1
      ARG(ICNT,1)=IZ/(1000.0)
      ARG(ICNT,2)=ANS/(1000.0)
      GO TO 200
  201 CONTINUE
      CALL SUB1(ARG,ICNT)
      RETURN
      END
```

Subroutine SUB1

SUB1 computes the discrepancy between the predicted and true scalar variations. Each discrepancy is then categorized according to its magnitude as described under the heading "Numerical Techniques" and is counted along with those discrepancies which fall within the same category and which precede it in storage. SUB1, after processing each stored variable, plots a frequency distribution curve of the discrepancies as part of the listing. The plot is scaled to page size and begins with the first category for which the ordinate is not zero and continues until all the nonzero ordinates have been plotted. The total number of discrepancies which are found to fall within each category is printed to the right of each ordinate in the event the user may be interested in its magnitude. Further, the standard deviation and the mean value of all the discrepancies are determined.

```
      SUBROUTINE SUB1(ARG, NOPT)
      DIMENSION ARG(1000,2),CLASS2(201),DIFF(1000)
      DIMENSION CLASS(201),COUNT(201),PERCENT(201),IN(2)
      DIMENSION IORD(11),IABSC(201),LORD(201),ABSC(201),   XM(2),YM1(3)
      ITAPE=5LTAPE7
      CNTMX=-200.0
      REWIND 7
      SUM=SUMSQ=B1=B2=0.0
      ISUM=0
      DO 20 I=1,200
   20 CLASS(I)=(I-101)*(0.02)+0.01
      DO 40 I=1,201
   40 COUNT(I)=0
      DO 130 J=1,NOPT
      IF (ARG(J,1).EQ.1000000.0.OR.ARG(J,2).EQ.1000000.0) GO TO130
      DIFFER=ARG(J,2)-ARG(J,1)
      SUM=SUM+DIFFER
      ISUM=ISUM+1
      SUMSQ=SUMSQ+DIFFER**2
      N=J+1
      IF(J.EQ.NOPT)GO TO 45
      DO 30 I=N,NOPT
      IF (ARG(I,1).EQ.1000000.0.OR.ARG(I,2).EQ.1000000.0) GO TO 30
      DARG1=ARG(I,1)-ARG(J,1)
      DARG2=ARG(I,2)-ARG(J,2)
      DELARG=DARG2-DARG1
   30 CONTINUE
  130 CONTINUE
   45 CONTINUE
      AV=SUM/FLOAT(ISUM)
      SIGMASQ=SUMSQ/FLOAT(ISUM)-AV**2
      DO 75 I=1,200
      COUNT(I)=0
   75 CLASS2(I)=(I-101)*(0.02)+0.01
      DO 131 J=1,NOPT
      IF (ARG(J,1).EQ.1000000.0.OR.ARG(J,2).EQ.1000000.0) GO TO131
      DIFFER=ARG(J,2)-ARG(J,1)
      B1=B1+(DIFFER-AV)**3
      B2=B2+(DIFFER-AV)**4
      DIFF(J)=DIFFER-AV
      DO 77 I=1,199
      IF(DIFF(J).GE.CLASS2(I).AND.DIFF(J).LE.CLASS2(I+1))GO TO 78
   77 CONTINUE
      GO TO 79
   78 COUNT(I+1)=COUNT(I+1)+1.0
   79 CONTINUE
  131 CONTINUE
      DO 80 I=1,200
      IF (COUNT(I).GT.CNTMX) CNTMX=COUNT(I)
   80 CLASS2(I)=(CLASS2(I)+0.01)*1000.0
      ISCALE=IFIX(CNTMX)/100+1
      JLAST=0$JMAX=0
      DO 70 I=1,200
      IABSC(I)=CLASS2(I)
      LORD(I)=IFIX(COUNT(I)/FLOAT(ISCALE))
      IF (LORD(I).GT.0) JMAX=JMAX+1
      IF (I.LE.11) IORD(I)=I-1
      IF (I.LT.96.OR.I.GT.106) ABSC(I)=2H
   70 CONTINUE
      ABSC( 96)=2H D
      ABSC( 97)=2H I
      ABSC(98 )=2H S
      ABSC(99 )=2H C
      ABSC(100)=2H R
      ABSC(101)=2H E
      ABSC(102)=2H P
      ABSC(103)=2H A
      ABSC(104)=2H N
      ABSC(105)=2H C
      ABSC(106)=2H Y
      ISTART=10H
      IEND=ISCALE*10
      PRINT 301,(IORD(I),I=1,11),IEND
  301 FORMAT(52X,*FREQUENCY OF OCCURRENCE*/,3X,11I10,*X*I4*   NUMBER*)
      ISYMB=1H*
      ISYMB1=1H
      DO 200 I=1,200
      J=LORD(I)
      JO=IFIX(COUNT(I))
      IF (J.GT.0.AND.JLAST.EQ.0) JLAST=1
      IF (J.EQ.0) GO TO 303
      JMAX=JMAX-1
      JJ=J+1
      PRINT300,ABSC(I),IABSC(I),(ISYMB,K=1 ,J),(ISYMB1,JK=JJ,100)
  300 FORMAT(1H+,A2,I10,112A1)
      GO TO 304
  303 CONTINUE
      IF (JLAST.EQ.0.OR.JMAX.LE.0) GO TO 200
      PRINT 300,ABSC(I),IABSC(I)
  304 CONTINUE
      PRINT 302,JO
  302 FORMAT(115X,I10)
  200 CONTINUE
      PRINT 2,AV,SIGMASQ
```

```
2 FORMAT(* ALL CONTROL POINTS HAVE BEEN COMPARED AND THE RESULTS ARE
 1 SHOWN*/* IT WAS DETERMINED FROM THE COMPARISONS THAT THERE WAS A*
 2/* MEAN DIFFERENCE =*E16.8,/* AND A*/* ST DEV =*E16.8,  /,* THE DE
 3PTH MATRIX WILL NOW BE PLOTTED*/* A TWO DIMENSIONAL CONTOUR CHART
 4WILL BE PREPARED WITH A SUITABLE GRID*/* EACH GRID LINE REPRESENTS
 5 AN INTEGRAL MULTIPLE OF DISPGRD(SEE INPUT INSTRUCTIONS) */* AND T
 6HE CONTOURS ARE POSITIONED RELATIVE TO THIS GRID*/* PLOTTING BEGIN
 7S WITH THE UPPER LEFT HAND CORNER OF THE DEPTH MATRIX*/)
  RETURN
  END
```

Program AUTOCON

AUTOCON is a very complex program which contains the necessary decision-making logic for managing the indexing, the computation, and the plotting of the contour line segments which cross each pair of adjacent triangles obtained from each 3 by 3 submatrix contained in the depth matrix. The basic concept is described in the section "Contour Plots." Further, AUTOCON establishes first a grid of equally spaced lines, each of which is separated by the scaled equivalent of DISPGRD (an input variable), and then it positions the plotted contours within the grid relative to the upper left-hand corner of the grid. Each grid line represents an integral multiple of DISPGRD. Thus, X-Y coordinates of the upper left-hand corner of the grid are the truncated values of

$$X = \text{Minimum } X/\text{DISPGRD}$$

and

$$Y = \text{Maximum } Y/\text{DISPGRD}$$

AUTOCON plots first the grid and establishes the origin of the plot, after which the first two adjacent triangles in figure 2 are processed and then plotted. The program indices are then reset for the processing and plotting required for the next two triangles in figure 2. This process is repeated until the 3 by 3 submatrix is completely plotted, after which the indices are reinitiated and a new 3 by 3 submatrix is considered.

AUTOCON possesses one option which permits the depth matrix to be plotted at a finer grid spacing. If SPP (an input variable) is greater than one, the 3 by 3 submatrix is subdivided so that a 2SPP + 1 by 2SPP + 1 submatrix is processed. The indexing is automatic and SPP cannot exceed 8.

```
OVERLAY (LINK,3,2)
PROGRAM AUTG CON
REAL       MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
DIMENSION IDEPH(160,3)
DIMENSION DEPTH(17,17)
DIMENSION IEL(60,60)
COMMON /BLK1/ IEL
COMMON /BLK3/ MAT,MIT
COMMON /BLK5/ MINWRD2,MAXWRD2,MINWRD3,MAXWRD3
COMMON /BLK6/ SP,SPP,LAST,AP      , LINEUNT
COMMON /BLK7/ N,NN,PX1,PY1,XMAX,YMAX,XMAY,YMAY,CI
COMMON /BLK8/ SQX,SQY
COMMON /BLK9/ PCS1X(20),POS1Y(20),CCNT (20)
COMMON /BLK10/POS2X(20),POS2Y(20),CONTA(20)
COMMON /BLK11/POS3X(20),POS3Y(20),CONTB(20)
COMMON /BLK12/POS4X(20),POS4Y(20),CONTC(20)
COMMON /BLK13/PCS5X(20),POS5Y(20),CONTD(20)
COMMON /LINKDC/ LINK,RECALL
CALL CALCOMP
```

```
      CALL LEROY
      MINWRD2=MINWRD2/FLOAT(LINEUNT)
      MINWRD3=MINWRD3/FLOAT(LINEUNT)
      MAXWRD2=MAXWRD2/FLOAT(LINEUNT)
      MAXWRD3=MAXWRD3/FLOAT(LINEUNT)
      DO 9913 I=1,3
      DO 9913 J=1,160
 9913 IDEPH(J,I)=0
      DO 9914 I=1,17
      DO 9914 J=1,17
 9914 DEPTH(I,J)=0.0
 1000 CONTINUE
      Y20=(MAXWRD2-FLOAT(IFIX(MAXWRD2)))
      X20=(MINWRD3-FLOAT(IFIX(MINWRD3)))
      IF (Y20.GT.0.0) Y20=1.0-Y20
      IF (Y20.LT.0.0) Y20=ABS(Y20)
      IF (X20.LT.0.0) X20=X20+1
      Y20=Y20*AP
      X20=X20*AP
      ILINE=IFIX(MAXWRD3)-IFIX(MINWRD3)
      JLINE=IFIX(MAXWRD2)-IFIX(MINWRD2)
      LINE CNT=0
  258 IF(Y20+( MAXWRD2-MINWRD2-FLOAT(JLINE))*AP+0.5.GE.0.0) JLINE=JLINE+
     11
      IF (Y20+(MAXWRD3-MINWRD3-FLOAT(ILINE))*AP+0.5.GE.0.0) ILINE=ILINE+
     11
      LINECNT=LINECNT+1
      IF (LINECNT.LT.2) GO TO 258
      X10=0.0
      IF (X20-0.4.LE.0.0) X10=X10-AP
      IF (X20-0.4.LE.0.0) ILINE=ILINE+1
      Y10=-FLOAT(JLINE)*AP
      CALL GRID (X10,Y10,AP,AP,ILINE,JLINE)
      DV=1.0/AP
      ORIGIN=FLOAT(IFIX(MINWRD3))
      IF (X20-0.4.LE.0.0) ORIGIN=ORIGIN-DV*AP
      XDIST=ILINE*AP
      YDIST=JLINE*AP
      CALL AXES(X10,Y10,0.0,XDIST,ORIGIN,DV,AP,DV,1H ,0.1,-1)
      ORIGIN=FLOAT(IFIX(MAXWRD2))-FLOAT(JLINE)
      CALL AXES(X10,Y10,90.,YDIST,ORIGIN,DV,AP,DV,1H ,0.1,+1)
      Y20=Y20*(-1.0)
      CALL CALPLT (X20,Y20,-3)
 9942 CONTINUE
      XMAT=FLOAT((MII-1)/2)*SP*2
      XMAX=XMAT
      XMAY=-XMAX
      YMAX=FLOAT((MAI-1)/2)*SP*(2.0 )
      YMAX=YMAX+0.05
      YMAY=-YMAX
      PY1=0.0
      PX1=0.0
      SP=SP/SPP
      SQX=SP
      SQY=-SP
      SPX=SP
      SPY=-SP
      IF (MIT.LT.20) NOC=1
      IF (MIT.EQ.(MIT/20)*20) NOC=MIT/20
      IF (MIT.GT.(MIT/20)*20) NOC=MIT/20+1
   21 K=0
      KK=0
      NX=NOC*20
      IS=SPP
  850 K=K+1
      IF (KK) 8850,8851,8850
 8850 JJ=K-1
      JK=K+1
      DO 8852 I=1,MIT
      IDEPH(I,JJ)=IDEPH(I,JK)
 8852 CONTINUE
 8851 CONTINUE
  779 CONTINUE
      IF (KK-MAT) 9779,812,812
 9779 CONTINUE
      KKK=KK+1
      DO 9933 J=1,MIT
      IDEPH(J,K)=IEL(KKK,J)
 9933 CONTINUE
  760 KK=KK+1
  782 IF (K-3)9850,784,784
 9850 K=K+1
      GO TO 8851
  784 J=1
      K=1
  698 IX=1
      IY=1
      L=1
      LL=L+IS
      LLL=LL+IS
      M=1
      MM=M+IS
      MMM=MM+IS
      K=1
  699 DEPTH(L,M)=IDEPH(J,K)
  700 IF (IDEPH(J+1,K))3,4,3
    3 IF (IDEPH(J,K))5,4,5
    4 IF (J.LE.MIT.AND.IDEPH(J,K).NE.0) GO TO 5
      SQX=SQX+((2.0)*SPP*SP)
      IF (J-1)6,6,7
    7 JCK1=J/2
      JCK2=JCK1*2
      IF (J-JCK2)8,9,8
    8 J=J+2
      GO TO 10
    9 J=J+1
      GO TO 10
    6 J=3
   10 JC=JC+2
      IF (JC-NX+1) 698,1851,1851
 1851 SQY=SQY-((2.0)*SPP*SP)
      K=1
      SQX=SPX
      GO TO 1852
    5 XINC1=IDEPH(J+1,K)-IDEPH(J,K)
      XINC1=XINC1/SPP
      XINC=ABS(XINC1)
  703 L=L+1
      IF (L-LL) 704,705,706
  704 IF (XINC1) 701,701,702
  701 DEPTH(L,M)=DEPTH(L-1,M)-XINC
      GO TO 703
  702 DEPTH(L,M)=DEPTH(L-1,M)+XINC
      GO TO 703
  705 J=J+1
      LL=LLL
      IF (L-LLL) 699,699,706
  706 JC=J
      J=J-2
      L=1
      K=K+1
      M=M+IS
      IF (M-MMM) 707,707,708
  707 LL=L+IS
      GO TO 699
  708 KC=K-1
      L=1
      M=1
      LL=L+IS
  798 YINC1=DEPTH(L,MM)-DEPTH(L,M)
      YINC1=YINC1/SPP
      YINC=ABS(YINC1)
 1803 M=M+1
      IF (M-MM) 1804,1805,1806
 1804 IF (YINC1) 1801,1801,1802
 1801 DEPTH(L,M)=DEPTH(L,M-1)-YINC
      GO TO 1803
 1802 DEPTH(L,M)=DEPTH(L,M-1)+YINC
      GO TO 1803
 1805 L=L+1
      M=M-IS
      IF (L-LLL) 798,798,799
  799 M=MM
      MM=MMM
      L=1
      GO TO 798
 1806 J=JC
      INDEX=1
      M2=2
      L2=2
      L1=2
      K1=2
      J1=1
      J2=1
      K2=1
      M1=1
  100 CONTINUE
      DO 9912 I=1,20
      CONT(I)=0.0
      CONTA(I)=0.0
      CONTB(I)=0.0
      CONTC(I)=0.0
      CONTD(I)=0.0
      POS1X(I)=0.0
      POS2X(I)=0.0
      POS3X(I)=0.0
      POS4X(I)=0.0
      POS5X(I)=0.0
      POS1Y(I)=0.0
      POS2Y(I)=0.0
      POS3Y(I)=0.0
      POS4Y(I)=0.0
      POS5Y(I)=0.0
 9912 CONTINUE
      I=1
      II=1
      III=1
      I4=1
      I5=1
      CALL LINE1 (DEPTH(L1,L2),DEPTH(M1,M2),CI,INDEX,I   ,D1)
      CALL LINE2 (DEPTH(M1,M2),DEPTH(J1,J2),CI,INDEX,II  ,D2)
      CALL LINE3 (DEPTH(L1,L2),DEPTH(J1,J2),CI,INDEX,III,D3)
      CALL LINE4 (DEPTH(J1,J2),DEPTH(K1,K2),CI,INDEX,I4  ,D4)
      CALL LINE5 (DEPTH(L1,L2),DEPTH(K1,K2),CI,INDEX,I5  ,D5)
      IF (D1) 101,103,103
  101 IF (D2) 133,114,114
  114 N=1
      NN=1
  134 IF (CONT(N)-CONTA(NN)) 131,132,133
  131 N=N+1
      IF (CONT(N))134,133,134
  132 IF (CONT(N))1132,133,1132
 1132 CALL PLOT (POS2X(NN),POS2Y(NN),POS1X(N),POS1Y(N),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),+1)
      IF (N) 133,133,1134
 1134 IF (CONTA(NN))134,133,134
  133 IF (D3) 338,140,140
  140 N=1
      NN=1
  153 IF (CONTB(N)-CONTA(NN))149,150,138
  149 IF (CONTB(N)) 152,138,152
  152 NN=NN+1
      IF (CONTA(NN)) 153,138,153
  150 IF (CONTB(N)) 1150,138,1150
 1150 CALL PLOT (POS3X(N),POS3Y(N),POS2X(NN),POS2Y(NN),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),-1)
      GO TO 153
  338 N=1
      NN=1
  350 IF (CONTB(N)-CONTA(NN)) 349,348,349
  349 N=N+1
      IF (CONTB(N)) 350,538,350
  348 IF (CONTB(N)) 1348,538,1348
 1348 CALL PLOT (POS2X(NN),POS2Y(NN),POS3X(N ),POS3Y(N ),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),-1)
      IF (CONTB(N)) 350,538,350
  538 N=1
      NN=1
  542 IF (CONT(N)-CONTB(NN))138,540,138
  540 IF (CONT(N)) 539,138,539
  539 IF (CONTB(NN)) 541,138,541
  541 CALL CALPLT (POS1X(N),POS1Y(N),3)
      CALL CALPLT (POS3X(NN),POS3Y(NN),2)
      N=N+1
      NN=NN+1
      GO TO 542
  103 IF (D2) 162,181,181
  162 N=1
      NN=1
  184 IF (CONT(N)-CONTA(NN)) 181,182,183
  183 N=N+1
      IF (CONT(N)) 184,181,184
  182 IF (CONT(N)) 1182,181,1182
 1182 CALL PLOT (POS1X(N),POS1Y(N),POS2X(NN),POS2Y(NN),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),+1)
      IF (N) 181,181,1184
 1184 IF (CONTA(NN))184,181,184
  181 IF (D3) 187,438,438
  187 N=1
      NN=1
  200 IF (CONTB(N)-CONTA(NN))138,198,199
  199 NN=NN+1
      IF (CONTA(NN)) 200,138,200
  198 IF (CONTB(N)) 1198,138,1198
 1198 CALL PLOT (POS3X(N),POS3Y(N),POS2X(NN),POS2Y(NN),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),-1)
      IF (CONTB(N)) 200,138,200
  438 N=1
      NN=1
  450 IF (CONTB(N)-CONTA(NN)) 538,448,449
  449 N=N+1
      IF (CONTB(N)) 450,538,450
  448 IF (CONTB(N)) 1448,538,1448
 1448 CALL PLOT (POS2X(NN),POS2Y(NN),POS3X(N ),POS3Y(N ),POS2X(NN),
     1 POS2Y(NN),CONTA(NN),-1)
      IF (CONTB(N))450,538,450
  138 IF (D3) 201,203,203
  201 IF (D4) 227,206,206
  206 N=1
      NN=1
  226 IF (CONTB(N)-CONTC(NN)) 223,224,227
  223 N=N+1
      IF (CONTB(N)) 226,227,226
  224 IF (CONTB(N)) 1224,227,1224
 1224 CALL PLOT (POS4X(NN),POS4Y(NN),POS3X(N),POS3Y(N),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),+1)
      IF (N) 227,227,2226
  227 IF (D5) 228,230,230
  230 N=1
      NN=1
  242 IF (CONTD(N)-CONTC(NN)) 239,240,241
  239 NN=NN+1
      IF (CONTC(NN)) 242,243,242
  240 IF (CONTD(N)) 1240,243,1240
 1240 CALL PLOT (POS5X(N),POS5Y(N),POS4X(NN),POS4Y(NN),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),-1)
      IF (CONTC(NN)) 242,243,242
  228 N=1
      NN=1
  255 IF (CONTD(N)-CONTC(NN)) 1252,253,241
  252 N=N+1
      IF (CONTD(N)) 255,241,255
  253 IF (CONTD(N)) 1253,241,1253
 1253 CALL PLOT (POS5X(N),POS5Y(N),POS4X(NN),POS4Y(NN),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),-1)
      IF (CONTC(NN)) 255,241,255
 1252 IF (CONTD(N)) 252,241,252
  241 N=1
      NN=1
  543 IF (CONTB(N)-CONTD(NN)) 243,544,243
  544 IF (CONTB(N)) 545,243,545
  545 IF (CONTD(NN)) 546,243,546
  546 CALL CALPLT (POS3X(N),POS3Y(N),3)
      CALL CALPLT (POS5X(NN),POS5Y(NN),2)
      N=N+1
      NN=NN+1
      GO TO 543
  203 IF (D4) 256,275,275
  256 N=1
      NN=1
 1274 IF (CONTB(N)-CONTC(NN)) 275,276,277
  277 N=N+1
      IF (CONTB(N)) 1274,275,1274
  276 IF (CONTB(N))1276,275,1276
 1276 CALL PLOT (POS4X(NN),POS4Y(NN),POS3X(N),POS3Y(N),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),+1)
      IF (N)  275,275,1274
  275 CONTINUE
      IF (D5) 278,280,280
  278 N=1
      NN=1
  291 IF (CONTD(N)-CONTC(NN)) 243,289,290
  290 NN=NN+1
      IF (CONTC(NN)) 291,243,291
  289 IF (CONTD(N)) 1289,243,1289
 1289 CALL PLOT (POS5X(N),POS5Y(N),POS4X(NN),POS4Y(NN),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),-1)
      IF (CONTC(NN)) 291,243,291
  280 N=1
      NN=1
  302 IF (CONTD(N)-CONTC(NN)) 241,300,301
  301 N=N+1
      IF (CONTD(N)) 302,241,302
  300 IF (CONTD(N)) 1300,241,1300
 1300 CALL PLOT (POS5X(N),POS5Y(N),POS4X(NN),POS4Y(NN),POS4X(NN),
     1 POS4Y(NN),CONTC(NN),-1)
      IF (CONTD(NN)) 302,241,302
  243 INDEX=INDEX+1
      IF (INDEX-4) 800,800,801
  800 GO TO (804,805,806,807) ,INDEX
  804 J1=J1+2
      J2=J2-2
      K1=K1+3
      K2=K2-1
      L1=L1+2
      M1=M1+1
      M2=M2-1
      GO TO 808
  805 J1=J1+2
      K1=K1+1
      K2=K2+1
      M1=M1+1
      M2=M2-1
      GO TO 100
  806 J2=J2+2
      K1=K1-1
      K2=K2+1
      M1=M1+1
      M2=M2+1
      GO TO 100
  807 J1=J1-2
      K1=K1-1
      K2=K2-1
      M1=M1-1
      M2=M2+1
      GO TO 100
  801 IX=IX+1
      IF (IX-IS) 809,809,810
  809 INDEX=1
      GO TO 804
  808 SQX=SQX+(2.0*SPX)
      GO TO 100
  810 IY=IY+1
      IF (IY-IS) 2811,2811,2812
 2811 SQX=SQX-(2.0*(SPP-1.0)*SPX)
      SQY=SQY+(2.0*SPY)
      J1=1
      K1=2
      K2=K2+1
      L1=2
      L2=L2+2
      M1=1
      M2=M2+1
      IX=1
  811 INDEX=1
      GO TO 100
 2812 SQX=SQX+(2.0*SPX)
      SQY=SQY-(2.0*(SPP-1.0)*SPY)
      IF (JC-NX+1) 698,851,851
  851 K=1
      SQX=SPX
      SQY=SQY+(2.0*SPP*SPY)
 1852 CONTINUE
 1008 FORMAT (2F10.5,I5)
      JC=1
      GO TO 850
 2226 IF (CONTC(NN)) 226,227,226
  812 IF (LAST) 1001,1003,1001
 1003 CONTINUE
 9941 CALL CALPLT(24.0,0.0,-3)
      GO TO 9942
 1001 WRITE (6,1002)
 1002 FORMAT (5H END  )
      CALL CALPLT (0.0,0.0,999)
      RETURN
      END
```

Subroutine LINE1

LINE1 computes and stores all contour intercepts with side 1 in figure 2 for each pair of adjacent triangles in the 3 by 3 submatrix in turn.

```
      SUBROUTINE LINE1 (DEPTHL,DEPTHM,CI,INDEX,I   ,D1)
      COMMON /BLK6/ SP
      COMMON /BLK8/ SQX,SQY
      COMMON /BLK9/ POS1X(20),POS1Y(20),CONT (20)
      D1=DEPTHL-DEPTHM
      KX=DEPTHL/CI
      SX=KX
      IF (D1.LT.0) CONT (I   )=SX*CI+CI
      IF (D1.GE.0) CONT (I   )=SX*CI
  104 CONTINUE
      IF (D1) 1,2,2
    1 CONTINUE
      IF (CONT(I)-DEPTHM.LE.0.0.AND.D1.LT.0.0) 105,106
    2 CONTINUE
      IF (CONT(I)-DEPTHM.LT.0.0.AND.D1.GE.0.0) 106,105
  105 IF (D1.LT.0) DC=CONT (I   )-DEPTHL
      IF (D1.GE.0) DC=DEPTHL-CONT (I   )
      IF (D1.EQ.0.0) D1=0.00000001
      GO TO (107,108,109,110),INDEX
  107 POS1X(I)=SQX-((DC/ABS(D1))*SP)
      POS1Y(I)=SQY
  111 I=I+1
      IF(I   .GT.20) PRINT 500
      IF (I.GT.20) B=(FOUR/0.0)**2
  500 FORMAT(1H1* CONTOUR IS TOO SMALL*/)
      IF (D1.LT.0.0) CONT (I   )=CONT (I   -1)+CI
      IF (D1.GE.0.0) CONT (I   )=CONT (I   -1)-CI
      GO TO 104
  108 POS1Y(I)=SQY+((DC/ABS(D1))*SP)
      POS1X(I)=SQX
      GO TO 111
  109 POS1X(I)=SQX+((DC/ABS(D1))*SP)
      POS1Y(I)=SQY
      GO TO 111
  110 POS1X(I)=SQX
      POS1Y(I)=SQY-((DC/ABS(D1))*SP)
      GO TO 111
  106 CONT(I)=0.0
      RETURN
      END
```

Subroutine LINE2

LINE2 computes and stores all contour intercepts with side 2 in figure 2 for each pair of adjacent triangles in the 3 by 3 submatrix in turn.

```
      SUBROUTINE LINE2 (DEPTHM,DEPTHJ,CI ,INDEX,II ,D2)
      COMMON /BLK6/ SP
      COMMON /BLK8/ SQX,SQY
      COMMON /BLK10/POS2X(20),POS2Y(20),CONTA( 20)
      D2=DEPTHM-DEPTHJ
      KX=DEPTHM/CI
      SX=KX
      IF (D2.LT.0) CONTA(II )=SX*CI+CI
      IF (D2.GE.0) CONTA(II )=SX*CI
  115 CONTINUE
      IF (D2) 1,2,2
    1 CONTINUE
      IF (CONTA(II)-DEPTHJ.LE.0.AND.D2.LT.0) 116,117
    2 CONTINUE
      IF (CONTA(II)-DEPTHJ.LT.0.AND.D2.GE.0) 117,115
  116 IF (D2.LT.0) DC=CONTA(II )-DEPTHM
      IF (D2.GE.0) DC=DEPTHM-CONTA(II)
      IF (D2.EQ.0.0) D2=0.000000001
      GO TO (118,119,120,121) ,INDEX
  118 POS2X(II)=SQX-SP
      POS2Y(II)=SQY+((DC/ABS(D2))*SP)
  122 II=II+1
      IF(II.GT.20) B=(FOUR/0.0)**2
      IF(II .GT.20) PRINT 500
  500 FORMAT(* CONTOUR INTERVAL IS TOO SMALL*/)
      IF (D2.LT.0.0) CONTA(II )=CONTA(II -1)+CI
      IF (D2.GE.0.0) CONTA(II )=CONTA(II -1)-CI
      GO TO 115
  119 POS2X(II)=SQX+((DC/ABS(D2))*SP)
      POS2Y(II)=SQY+SP
      GO TO 122
  120 POS2X(II)=SQX+SP
      POS2Y(II)=SQY-((DC/ABS(D2))*SP)
      GO TO 122
  121 POS2X(II)=SQX-((DC/ABS(D2))*SP)
      POS2Y(II)=SQY-SP
      GO TO 122
  117 CONTA(II)=0.0
      RETURN
      END
```

Subroutine LINE3

LINE3 computes and stores all contour intercepts with side 3 in figure 2 for each pair of adjacent triangles in the 3 by 3 submatrix in turn.

```
      SUBROUTINE LINE3 (DEPTHL,DEPTHJ,CI,INDEX,III,D3)
      COMMON /BLK6/ SP
      COMMON /BLK8/ SQX,SQY
      COMMON /BLK11/POS3X(20),POS3Y(20),CONTB(20)
      D3=DEPTHL-DEPTHJ
      KX=DEPTHL/CI
      SX=KX
      IF (D3.LT.0) CONTB(III)=SX*CI+CI
      IF (D3.GE.0) CONTB(III)=SX*CI
  341 CONTINUE
      IF (D3) 1,2,2
    1 CONTINUE
      IF (CONTB(III)-DEPTHJ.LE.0.0.AND.D3.LT.0.0)339,340
    2 CONTINUE
      IF (CONTB(III)-DEPTHJ.LT.0.0.AND.D3.GE.0.0)340,339
  339 IF (D3.LT.0) DC=CONTB(III)-DEPTHL
      IF (D3.GE.0) DC=DEPTHL-CONTB(III)
      IF (D3.EQ.0.0) D3=0.000000001
      GO TO (342,343,344,345) ,INDEX
  342 POS3X(III)=SQX-((DC/ABS(D3))*SP)
      POS3Y(III)=SQY+((DC/ABS(D3))*SP)
  346 III=III+1
      IF(III.GT.20) PRINT 500
      IF(III.GT.20) B=(FOUR/0.0)**2
  500 FORMAT(* CONTOUR INTERVAL IS TOO SMALL*/)
      IF (D3.LT.0.0) CONTB(III)=CONTB(III-1)+CI
      IF (D3.GE.0.0) CONTB(III)=CONTB(III-1)-CI
      GO TO 341
  343 POS3X(III)=SQX+((DC/ABS(D3))*SP)
      POS3Y(III)=SQY+((DC/ABS(D3))*SP)
      GO TO 346
  344 POS3X(III)=SQX+(DC/ABS(D3))*SP
      POS3Y(III)=SQY-(DC/ABS(D3))*SP
      GO TO 346
  345 POS3X(III)=SQX-((DC/ABS(D3))*SP)
      POS3Y(III)=SQY-((DC/ABS(D3))*SP)
      GO TO 346
  340 CONTB(III)=0.0
      RETURN
      END
```

Subroutine LINE4

LINE4 computes and stores all contour intercepts with side 4 in figure 2 for each pair of adjacent triangles in the 3 by 3 submatrix in turn.

```
      SUBROUTINE LINE4 (DEPTHJ,DEPTHK,CI,INDEX,I4 ,D4)
      COMMON /BLK6/ SP
      COMMON /BLK8/ SQX,SQY
      COMMON /BLK12/POS4X(20),POS4Y(20),CONTC(20)
      D4=DEPTHJ-DEPTHK
      KX=DEPTHJ/CI
      SX=KX
      IF (D4.LT.0) CONTC(I4 )=SX*CI+CI
      IF (D4.GE.0) CONTC(I4 )=SX*CI
  207 CONTINUE
      IF (D4) 1,2,2
    1 CONTINUE
      IF (CONTC(I4)-DEPTHK.LE.0.0.AND.D4.LT.0.0)208,209
    2 CONTINUE
      IF (CONTC(I4)-DEPTHK.LT.0.0.AND.D4.GE.0.0)209,208
  208 IF (D4.LT.0) DC=CONTC(I4 )-DEPTHJ
      IF (D4.GE.0.0) DC=DEPTHJ-CONTC(I4)
      IF (D4.EQ.0.0) D4=0.00000001
      GO TO (210,211,212,213) ,INDEX
  210 POS4X(I4)=SQX-SP+((DC/ABS(D4))*SP)
      POS4Y(I4)=SQY+SP
  214 I4=I4+1
      IF(I4 .GT.20) PRINT 500
      IF(I4.GT.20) B=(FOUR/0.0)**2
  500 FORMAT(* CONTOUR INTERVAL IS TOO SMALL*/)
      IF (D4.LT.0.0) CONTC(I4 )=CONTC(I4 -1)+CI
      IF (D4.GE.0.0) CONTC(I4 )=CONTC(I4 -1)-CI
      GO TO 207
  211 POS4X(I4)=SQX+SP
      POS4Y(I4)=SQY+SP-((DC/ABS(D4))*SP)
      GO TO 214
  212  POS4X(I4)=SQX+SP-((DC/ABS(D4))*SP)
      POS4Y(I4)=SQY-SP
      GO TO 214
  213 POS4X(I4)=SQX-SP
      POS4Y(I4)=SQY-SP+((DC/ABS(D4))*SP)
      GO TO 214
  209 CONTC(I4)=0.0
      RETURN
      END
```

Subroutine LINE5

LINE5 computes and stores all contour intercepts with side 5 in figure 2 for each pair of adjacent triangles in the 3 by 3 submatrix in turn.

```
      SUBROUTINE LINE5 (DEPTHL,DEPTHK,CI,INDEX,I5 ,D5)
      COMMON /BLK6/ SP
      COMMON /BLK8/ SQX,SQY
      COMMON /BLK13/POS5X(20),POS5Y(20),CONTD(20)
      D5=DEPTHL-DEPTHK
      KX=DEPTHL/CI
      SX=KX
      IF (D5.LT.0) CONTD(I5 )=SX*CI+CI
      IF (D5.GE.0) CONTD(I5 )=SX*CI
  244 CONTINUE
      IF (D5) 1,2,2
    1 CONTINUE
      IF (CONTD(I5)-DEPTHK.LE.0.0.AND.D5.LT.0.0) 245,246
    2 CONTINUE
      IF (CONTD(I5)-DEPTHK.LT.0.0.AND.D5.GE.0.0) 246,245
  245 IF (D5.LT.0) DC=CONTD(I5 )-DEPTHL
      IF (D5.GE.0.0) DC=DEPTHL-CONTD(I5)
      IF (D5.EQ.0.0) D5=0.00000001
      GO TO (247,248,249,250) ,INDEX
  247 POS5X(I5)=SQX
      POS5Y(I5)=SQY+((DC/ABS(D5))*SP)
  251 I5=I5+1
      IF(I5 .GT.20) PRINT 500
      IF(I5.GT.20) B=(FOUR/0.0)**2
  500 FORMAT(* CONTOUR INTERVAL IS TOO SMALL*/)
      IF (D5.LT.0.0) CONTD(I5 )=CONTD(I5 -1)+CI
      IF (D5.GE.0.0) CONTD(I5 )=CONTD(I5 -1)-CI
      GO TO 244
  248 POS5X(I5)=SQX+((DC/ABS(D5))*SP)
      POS5Y(I5)=SQY
      GO TO 251
  249 POS5X(I5)=SQX
      POS5Y(I5)=SQY-((DC/ABS(D5))*SP)
      GO TO 251
  250 POS5X(I5)=SQX-((DC/ABS(D5))*SP)
      POS5Y(I5)=SQY
      GO TO 251
  246 CONTD(I5)=0.0
      RETURN
      END
```

Subroutine PLOT

PLOT generates the necessary plotter instructions for drawing a straight line between two intercepts for a particular contour line. If the contour terminates or begins with a side which connects two boundary elements of the depth matrix, the contour value is plotted.

PLOT is called from within a series of nested logical statements which first match the corresponding stored intercepts and then index sequentially until all contours within the confines of a triangle are plotted.

```
      SUBROUTINE PLOT (POS3X,POS3Y,POS1X,POS1Y,POS2X,POS2Y,CONTA,ISIGN)
      COMMON /BLK7/ N,NN,PX1,PY1,XMAX,YMAX,XMAY,YMAY,CI
      COMMON /LIMITS/ MINI,MAXI,IADDI
      DATA CCT/03770000000000000000/
      CALL CALPLT (POS3X,POS3Y,3)
      CALL CALPLT (POS1X,POS1Y,2)
      DX=POS2Y-PY1
      DY=POS2X-PX1
      S=SQRT((DX*DX)+DY*DY)
      AMAX=XMAX+0.05
      TEMPX=CONTA
      IF (CONTA.LT.100) GO TO 29
      CONTA=CONTA-IADDI
      IF (S-0.07) 29,30,30
   30 IF (POS2X-0.0015) 23,22,22
   23 CALL COLUMN (-0.05,POS2Y,0.07,CONTA,0.0,-1)
      PY1=POS2Y
      PX1=POS2X
      GO T O 29
   22 IF (POS2Y+0.0015) 24,24,25
   25 CALL NUMBER (POS2X,0.05,0.07,CONTA,90.0,-1)
      PY1=POS2Y
      PX1=POS2X
      GO TO 29
   24 IF (POS2Y+YMAX-0.0515) 26,27,27
   26 CALL COLUMN (POS2X,YMAY,0.07,CONTA,90.0,-1)
      PY1=POS2Y
      PX1=POS2X
      GO TO 29
   27 IF (POS2X-XMAX+0.0015) 29,29,28
   28 CALL NUMBER (AMAX,POS2Y,0.07,CONTA,0.0,-1)
   99 CONTINUE
      PY1=POS2Y
      PX1=POS2X
   29 N=N-1*ISIGN
      CONTA=TEMPX
      NN=NN+1
      RETURN
      END
```

Subroutine COLUMN

By George Salley

COLUMN right-justifies a BCD (binary-coded decimal) number.

```
      SUBROUTINE COLUMN (X,Y,S,FPN,TH,N)
C      X AND Y ARE THE COORDINATES OF THE LOWER RIGHTMOST EDGE
C      S    IS THE SIZE OF CHARACTER TO BE USED IN PRINTING
C      FPN IS AN ACTUAL FLOATING POINT NUMBER WHOSE VALUE IS TO BE
C           PRINTED ON THE PLOTTED  OUTPUT
C      TH    IS AN ANGLE (DEGREES) AT WHICH THE NUMBER IS TO APPEAR
C      N  IS AN INTEGER SPECIFYING THE ACCURACY TO WHICH THE NUMBER IS
C          TO BE PRINTED
      SPC = .857143
      SPG =.285714
      NL = 0
      M = 0
      IF (N) 50,20,20
   20 NL = N+1
      M = N
   50 TFPN = ROUND(FPN)
       IFPN = TFPN*10**M
      IF (IFPN) 70,55,75
   55 IF (TFPN) 60,60,100
   60 NL = NL+1
      GO TO 100
   70 NL = NL+1
      TFPN = ABS(TFPN)
   75 M = 0.4343*ALOG(TFPN)+1.0
      IF (M) 100,100,80
   80 NL = NL + M
  100 DLT =(SPC * S *(FLOAT(NL)))-(SPG*S)
      T = TH*0.017453
      YP = Y -(DLT*SIN(T))
      XP = X -(DLT*COS(T))
  120 CALL NUMBER (XP,YP,S,FPN,TH,N)
      RETURN
      END
```

OPERATING INSTRUCTIONS

Input

Basically the input data can be divided into two categories. One category is termed "program data," and the other is referred to as "control-point data." (The maximum number of control points is limited to 1000.) Program data are composed of first a format (user's choice) for reading the floating-point program variables; the fixed-point program variables and the appropriate codes for each option follow. If the control-point data are not on tape, the user must supply a format for reading each control-point data card. The control-point data are composed of individual cards or tape records each of which contains the X-Y coordinate of each scalar variation and its magnitude Z, in that order.

The control-point data can originate from any source. If an algebraic expression is to be utilized to obtain the control-point data, irregularly spaced scalar variations are preferred. However, regularly spaced scalar variations can be processed with the program.

In view of the various options available to the user (to be discussed subsequently), the number of required data cards is not rigidly fixed. Thus, a figure indicating the card on which a particular variable is to be found is not possible. However, in figure 4 the order in which each variable is read and the format utilized for reading it are indicated. In table II program variables are listed and its function in the program is defined.

Some knowledge of the input data is required to adequately assign the program variable values. Caution should be exercised in the selection of DIST (grid separation), since it is primarily responsible for establishing the dimensions of the depth matrix. If DIST is too small, the maximum dimensions of the depth matrix (60 by 60) will be exceeded. DIST also controls the dimensions of the region within which control points can be found. (See the discussion on "Depth Matrix.") All the control-point data are searched for values which fall within a ring with the initial inner radius set equal to zero and the outer radius set equal to the incremental radial difference of DIST/10. The ring has a mean radius of one-half the sum of the two radii. Since the mean radius of the ring is incremented in multiples of the incremental radial difference before each search of the data, DIST probably influences the required computer time more than any of the remaining variables.

The units for DIST, X, Y, and DISPGRD are required to be the same. If X and Y represent different quantities, such as Mach number and height, respectively, an appropriate scale factor must be applied to X to establish a fictitious one-to-one correspondence between the magnitudes of the two variables. Since the objective is to establish roughly the same order of magnitude between the two variables, Mach number should be converted into the units of height per second.

It is assumed that X, Y, and sometimes Z are expressed as integral multiples of the user's basic unit (meters, inches, etc.). Thus, if it is desired to contour data which are expressible in fractional parts of the basic unit, X, Y, and Z must be scaled up in magnitude so that the significant part thereof is plotted. That is, data expressed to 0.0001 inch are to be multiplied by 10 000. If similarly treated, SCLE will reestablish the desired plot dimensions.

The inputs MAXI, MINI, TANG, and IDMX establish the criteria for either accepting or rejecting a given control point. The upper and lower limits for the magnitudes of the control-point X-component are set by MAXI and MINI. If both MAXI and MINI are zero, no data points are rejected. Together TANG and IDMX determine whether the relative height difference between the Z-coordinates of two control points is sufficiently small to permit their combined use in the gradient determinations. Of the two variables TANG is the dominant variable over a radial distance equal to the product of TANG and IDMX. Thereafter, IDMX becomes the dominant variable. The required computer time can be affected by TANG and IDMX.

The units for Z, IDMX, and CI are the same. The input Z may be any scalar variation (height, temperature, etc.). The program assumes IDMX, CI, and Z are integral multiples of a basic unit and not fractional parts thereof.

Options

Control-point data.- The control-point data may be on either punched cards or tape and may be either floating point or fixed point. Program codes specify which of the four options are to be expected. If the punched-card option is selected, a format for reading each data card must be furnished by the user. The subroutine READIT will automatically place the data in the proper form and establish the proper type required for processing.

Sorting of control-point data.- There is no requirement for the control-point data to be in a specific order. The program subroutine GRIDIT will automatically scan all the data furnished until a maximum of KMAX control points are selected for the gradient computation. The scanning process is, however, a very time-consuming activity. In fact, it is by far the largest user of computer time and should be minimized as much as possible. As a result, the program contains an optional sorting routine, SORTIT. SORTIT can be called with a code of +1 for ISRT. The variable ISRTOPT specifies the manner in which the data are to be sorted.

Interpolation.- The digital program will expand each 3 by 3 submatrix into a 2SPP + 1 by 2SPP + 1 submatrix in AUTOCON. Each missing element, if any, is determined by use of linear interpolation. The input SPP permits the size of each triangle over which the line segments, representing contour lines, are drawn to be reduced so that the contours will appear more continuous. The value of SPP can not exceed 8.

Deck Setup

As stated, the digital contouring program is composed of seven overlaid programs and their various subroutines. CONTOUR is the main overlay (0,0 level) and is, as a result, the program identification.

A deck setup for a case in which the control-point data are contained on a deck of punched cards is shown in figure 5, and a deck setup for a case in which a binary tape file is used is shown in figure 6. The binary tape must be one which was generated by using RECOUT. If the control-point data are contained on a deck of punched cards, the control cards for tape file 9 are not needed.

Output

The digital program outputs data in two forms. First, there is a printed listing of information pertaining to the depth matrix (dimensions, origin, etc.) and to the plot size, as well as a frequency distribution plot of all the differences between the actual scalar measurement at each control point and that predicted by the depth matrix. Second, plotter instructions for a graphic display of the scalar variations are written on tape. The tape is then used by the mechanical plotter to produce the contour charts.

Several diagnostics have been programed into the digital contouring program. Each attempts to isolate potential execution difficulties and to inform the user before they occur, in the interest of saving computer time. The following reasons are given for the diagnostics:

Control-point data exceed 1000 points.
Maximum dimensions of depth matrix are exceeded.
Depth-matrix dimensions are too small.
Minimum depth-matrix elevation is too small.
Depth matrix cannot be smoothed.
Contour interval is too small.

A mode 4 stop is anticipated for each diagnostic. The diagnostics themselves are sometimes lengthy and are not given herein.

```
JOB,1,1000,65000.  ----------------------
RUN(S)
NORFL.
SETINDF.
LGO.
REWIND(CALTPE)
REQUEST TAPEXX,HI,X.  CALTP , RIL, -------
REWIND(TAPEXX)
COPYBF(CALTPE,TAPEXX)
UNLOAD(TAPEXX)
EXIT.
UNLOAD(TAPEXX)
7,8,9
  :
  :
Source Deck
  :
  :
7,8,9
Program Data
Control Point Data
6,7,8,9
```

NOTE: TAPEXX may be any legal file name.

Figure 5.- Deck setup when a punched deck of control-point data is used.

```
JOB,1,1000,65000.
RUN(S)
NORFL.
SETINDF.
REQUEST TAPE1,HY.  XXXXXX ,ROL,
REWIND(TAPE1)
REWIND(TAPE9)
COPYBF(TAPE1, TAPE9)
REWIND(TAPE9)
DROPFIL(TAPE1)
LGO.
REWIND(CALTPE)
REQUEST TAPEXX,HI,X.   CALTP , RIL, ---------
REWIND(TAPEXX)
COPYBF(CALTPE,TAPEXX)
UNLOAD(TAPEXX)
EXIT
UNLOAD(TAPEXX)
7,8,9
  .
  .
  .
  .
Source Deck
  .
  .
  .
  .
7,8,9
Program Data
6,7,8,9
```

NOTE: TAPEXX may be any legal file name
XXXXXX may be any legal tape number

Figure 6.- Deck setup when a binary file of control-point data is used.

SAMPLE PROBLEM

A sample problem was set up for a punched-card deck of control-point data. For each program data card, the magnitude of each variable punched on each data card is shown in figure 7, and in table III, the control-point data read by the variable format FRMT ((3I10) in fig. 7) are tabulated. The plotted graphic is shown in figure 8.

Card type	Column number
	12345678901234567890123456789012345678901234567890123456789012345678901234567890
1	(4E18.8/4E18.8)
2	10 10 6000
3	+30000000E+04 +12000000E+02 +10000000E+01 +10000000E+01
4	+10000000E+03 +90000000E+05 +12000000E+05 +10000000E+01
5	-1 -3600 9000
6	-1 +1 +0
7	(3I10)

Figure 7.- Program data for sample problem.

The computer program requires approximately 65 000 octal words of storage, and the execution time varies from job to job. The execution time is primarily dependent on the number of control points being processed, the distribution of the control points, and the magnitude of DIST. On the average, $1\frac{1}{3}$ seconds of central processor time and an equivalent amount of peripheral time is required for each control point being processed. It is possible to reduce the time required to process a particular job by increasing DIST. The sample problem required approximately 739.95 decimal seconds of central processor time and 767.18 decimal seconds of peripheral time. The peripheral time may vary considerably even if the same data are used.

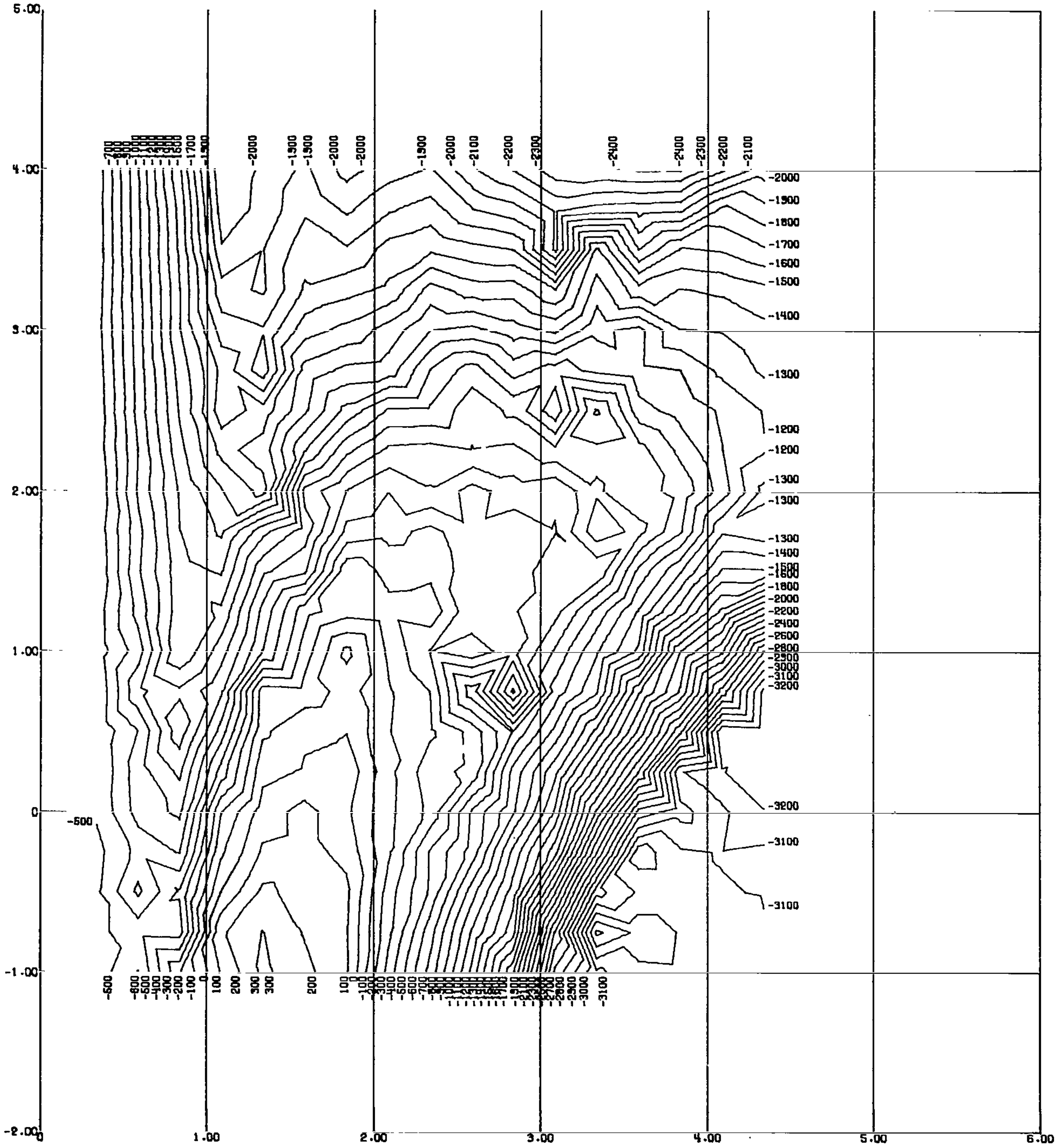

Figure 8.- Contour chart for sample problem.

Listing of Unsmoothed Depth-Matrix Data

```
 THE FOLLOWING IS INFORMATION PERTAINING TO THE DEPTH MATRIX--SIZE,DIMENSIONS,ETC.


      CORNER COORDINATES OF AREA TO BE CONTOURED

NORTH WEST CORNER--X=  4.00000000E+03  Y=  6.00307200E+04
SOUTH EAST CORNER--X=  5.09880000E+04  Y=  0.


DIMENSIONS OF REQUIRED PLOTTING SURFACE EXCLUSIVE OF GRID

Y DIMENSION OF DEPTH MATRIX IS         8.00400INCHES
X DIMENSION OF DEPTH MATRIX IS         6.26493INCHES

  22ELEMENTS IN Y DIRECTION OF DEPTH MATRIX
  17ELEMENTS IN X DIRECTION OF DEPTH MATRIX

YOU HAVE ELECTED TO REJECT ALL CONTROL POINTS WITH SCALAR VARIATIONS LESS THAN -3600 AND GREATER THAN  9000
THUS,
YOU HAVE A MINIMUM SCALAR VARIATION OF      -3554
YOU HAVE A MAXIMUM SCALAR VARIATION OF        327


 THE FOLLOWING INFORMATION REGARDING YOUR UNSMOOTHED DEPTH MATRIX IS FURNISHED



ADJUSTED BOUNDARIES OF DEPTH MATRIX

MINIMUM Y  0.                 MAXIMUM Y  6.00307200E+04
MINIMUM X  4.00000000E+03 MAXIMUM X  5.09880000E+04


UNSMOOTHED DEPTH MATRIX


  9461   9000   8504   7903   8021   8123   7962   8055   8107   7914   7794   7637   7612   7586   7592   7784   7953
  9457   9010   8559   7944   8029   8185   8082   8188   8247   8105   8042   7886   7947   7960   7979   8232   8130
  9463   9015   8575   8009   8098   8257   8209   8339   8383   8333   8299   7873   8622   8201   8409   8391   8311
  9452   9030   8601   8137   8085   8313   8363   8422   8642   8551   8559   8403   8753   8538   8606   8581   8529
  9440   9045   8601   8250   8210   8456   8529   8643   8797   8829   8745   8677   8876   8934   8703   8695   8628
  9458   9052   8656   8370   8135   8639   8726   8745   9004   9155   8956   9051   8970   8929   8900   8774   8681
  9470   9060   8687   8341   8442   8682   8934   9116   9117   9263   9177   8925   9413   9042   8956   8888   8793
  9460   9093   8729   8468   8546   8898   9177   9326   9340   9306   9326   9232   9277   9242   9074   8902   8806
  9492   9077   8785   8608   8465   9235   9404   9499   9411   9542   9448   9490   9313   9231   9174   8950   8670
  9461   9131   8802   8682   8969   9149   9531   9562   9631   9525   9563   9504   9237   9332   9113   8716   8772
  9482   9114   8836   8786   9178   9282   9775   9668   9672   9540   9568   9423   9487   9070   8808   8482   8489
  9450   9163   8839   8918   9332   9662   9756   9731   9770   9512   9507   9437   9172   8891   8556   8171   7841
  9453   9248   8804   9052   9403   9747  10026   9778   9596   9672   9532   9284   9013   8781   8102   7821   7196
  9489   9122   9016   9180   9859   9840   9942   9813   9699   9243   9823   9159   8916   8323   7987   7357   6767
  9478   9297   8938   9341   9881   9985   9996   9805   9720   9566   9291   8968   8568   8076   7520   6710   6215
  9499   9245   9078   9539   9969  10010  10072   9832   9639   9573   9067   8730   8237   7701   6909   6088   5572
  9485   9258   9178   9791  10048  10136  10024   9827   9674   9283   9015   8582   7880   7163   6339   5534   4903
  9529   9371   9290   9879  10115  10137  10070   9817   9491   9261   8776   8341   7507   6738   5753   4953   4217
  9517   9277   9512  10000  10221  10163  10125   9751   9483   9046   8651   7912   7033   6186   5239   4379   3568
  9517   9460   9449  10181  10303  10192  10111   9727   9345   8886   8537   7493   6576   5624   4726   3814   2958
  9556   9398   9782  10107  10322  10231  10113   9658   9263   8769   8214   7271   6071   5068   4139   3224   2338
  9507   9351   9790  10138  10318  10180  10057   9609   9195   8609   7911   7132   5660   4571   3605   2662   1736
```

Listing of Smoothed Depth-Matrix Data

```
 THE FOLLOWING INFORMATION REGARDING THE SMOOTHED DEPTH MATRIX IS FURNISHED


IT HAS BEEN DETERMINED THAT EACH ELEMENT OF YOUR UNSMOOTHED DEPTH MATRIX IS INADEQUATELY FILLED


ADJUSTED BOUNDARIES OF DEPTH MATRIX

MINIMUM Y  0.                 MAXIMUM Y  6.0030720E+04
MINIMUM X  4.0000000E+03 MAXIMUM X  5.0988000E+04


REGRID WILL NOT PERMIT THE INTERNAL ELEMENTS OF I TO EXCEED THE FOLLOWING LIMITS

MINIMUM ELEVATION PRIOR TO SMOOTHING =        -3424
MAXIMUM ELEVATION PRIOR TO SMOOTHING =          322

BE CAUTIOUS OF ELEMENT I(15,17)

MINIMUM ELEVATION AFTER INTERPOLATION ETC =        -3424

MAXIMUM EXPECTED SLOPE BETWEEN TWO CONSECUTIVE GRID POINTS  =            1.00000


SMOOTHED DEPTH MATRIX


 9461   9000   8504   7933   8021   8123   7962   8055   8107   7914   7794   7637   7612   7586   7592   7784   7953
 9457   9010   8559   7944   8129   8185   8082   8188   8247   8105   8042   7886   7947   7960   7979   8232   8130
 9463   9015   8575   8029   8098   8257   8209   8339   8383   8333   8299   7873   8622   8201   8409   8391   8311
 9452   9030   8601   8137   8085   8313   8363   8422   8642   8551   8559   8403   8753   8538   8606   8581   8529
 9447   9045   8601   8251   8210   8456   8529   8643   8797   8829   8745   8677   8876   8934   8703   8695   8628
 9459   9052   8655   8377   9135   8639   8726   8745   9004   9155   8956   9051   8970   8929   8900   8774   8681
 9470   9069   8687   8341   8642   8682   8934   9116   9117   9263   9177   8925   9413   9042   8956   8888   8793
 9460   9093   8729   8458   8546   8908   9177   9326   9340   9306   9326   9232   9277   9242   9074   8902   8806
 9492   9077   8785   8508   8465   9235   9404   9459   9411   9542   9448   9490   9313   9231   9174   8950   8670
 9441   9131   8802   8582   8969   9149   9531   9562   9631   9525   9563   9504   9237   9332   9113   8716   8772
 9482   9114   8836   8736   9178   9282   9775   9668   9672   9540   9568   9423   9487   9070   8808   8482   8489
 9450   9163   8839   8918   9332   9662   9756   9731   9770   9512   9507   9437   9172   8891   8556   8171   7841
 9453   9248   8804   9052   9403   9747  10026   9778   9596   9672   9532   9294   9013   8781   8102   7821   7196
 9489   9122   9016   9187   9859   9840   9942   9813   9699   9243   9823   9159   8916   8323   7987   7357   6767
 9478   9297   8939   9341   9881   9985   9996   9905   9720   9566   9291   8968   8558   8076   7520   6710   6738
 9490   9245   9078   9539   9969  10010  10072   9932   9639   9573   9067   8730   8237   7701   6909   6809   6709
 9485   9259   9178   9731  10048  10136  10024   9827   9674   9283   9015   8582   7880   7163   7036   6922   6808
 9529   9371   9290   9379  10115  10137  10070   9817   9491   9261   8776   8341   7507   6738   6887   6904   6921
 9517   9277   9512  10009  10221  10163  10125   9751   9483   9046   8651   7912   7033   6885   6886   6895   6904
 9517   9460   9449  10191  10303  10192  10111   9727   9345   8886   8537   7493   6576   6730   6838   6851   6894
 9556   9398   9782  10107  10322  10231  10113   9658   9263   8769   8214   7271   6923   6826   6817   6834   6851
 9507   9351   9790  10198  10318  10180  10057   9609   9195   8609   7911   7132   7270   6922   6926   6817   6908

 YOUR SMOOTHED DEPTH MATRIX, I, IS PRINTED ABOVE.
```

Sample Problem Output for Evaluation of Smoothed Depth Matrix

```
AT THIS POINT THE SCALAR VARIATIONS PREDICTED BY I AT THE LOCATION OF EACH CONTROL POINT
IS COMPARED WITH THE SCALAR MAGNITUDE OF THE CONTROL POINT.
A PLOT FOR FREQUENCY OF OCCURRENCE VS DISCREPANCY FOLLOWS AS PART OF THIS LISTING



                                                       FREQUENCY OF OCCURRENCE
            0         1         2          3         4         5          6         7          8         9        10X  30  NUMBER
 D       -70*                                                                                                                   5
 I       -50**                                                                                                                  6
 S       -40********                                                                                                           24
 C       -20*****************                                                                                                  51
 R         0****************************************                                                                          120
 F        20**************************************************************************                                        223
 P        40*****************************                                                                                      88
 A        60***********                                                                                                        33
 N        70*****                                                                                                              16
 C        90*                                                                                                                   4
 Y       110*                                                                                                                   5
         130*                                                                                                                   4
ALL CONTROL POINTS HAVE BEEN COMPARED AND THE RESULTS ARE SHOWN
IT WAS DETERMINED FROM THE COMPARISONS THAT THERE WAS A
MEAN DIFFERENCE =  7.15327593E-03
AND A
ST DEV =  2.11931572E-03
THE DEPTH MATRIX WILL NOW BE PLOTTED
A TWO DIMENSIONAL CONTOUR CHART WILL BE PREPARED WITH A SUITABLE GRID
EACH GRID LINE REPRESENTS AN INTEGRAL MULTIPLE OF DISPGPD(SEE INPUT INSTRUCTIONS)
AND THE CONTOURS ARE POSITIONED RELATIVE TO THIS GRID
PLOTTING BEGINS WITH THE UPPER LEFT HAND CORNER OF THE DEPTH MATRIX

END
```

FORTRAN VARIABLE NAMES

The major FORTRAN variable names and arrays are defined in table IV. Where required, the algebraic equivalent of the variable is shown. Certain variables assumed different names in different programs and subroutines and are so indicated in the table.

CONCLUDING REMARKS

A generalized digital contouring program has been developed and described. The program will accept any type of data regardless of the units or the dimensions of the region over which the data are to be plotted. Further, the program makes no assumptions regarding either the absolute or the relative magnitudes of the scalar variations being plotted.

The digital contouring program is limited in the number of acceptable control points and, to a degree, in the magnitude of the grid separation. A maximum of 1000 control points can be processed at one time. The grid separation should not be so small that it requires more than 60 grid points in any one direction in the depth matrix which represents the region being contoured. Further, the depth matrix cannot be smaller than a 3 by 3 array.

A sample problem with 552 data points was processed. The resulting computer listing of the output is shown, as well as the machine-generated contour chart.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., September 3, 1970.

REFERENCES

1. Osborn, Roger T.: An Automated Procedure for Producing Contour Charts. IM No. 67-4, U.S. Naval Oceanographic Office, Feb. 1967. (Available from DDC as AD 807617.)

2. Anon.: Control Data 6400/6500/6600 Computer Systems FORTRAN Reference Manual. Publ. No. 60174900B, Control Data Corp., Nov. 1967.

3. Anon.: Control Data 6400/6500/6600 Computer Systems SORT/MERGE Reference Manual. Publ. No. 60176900, Control Data Corp., Mar. 1967.

TABLE I.- FUNCTION AND STORAGE OF EACH OVERLAID PROGRAM AND SUBROUTINE

Overlay	Program	Subroutine	Storage	Description	Page
(0, 0)			$16\ 460_8$		13
	CONTOUR		$12\ 327_8$	Reads program options and directs the order of execution of the overlays	
(1, 0)			$5\ 625_8$		14
	READIT		442_8	Reads either fixed-point or floating-point X-, Y-, and Z-coordinates from either card or tape and generates a coded binary tape of data suitable for use by the remaining overlays and subroutines	
		RECIN	302_8	A system routine for reading binary and BCD tapes generated by RECOUT	
		RECOUT	340_8	A system routine for packing binary and/or BCD shorter records into longer logical records of approximately 512_{10} core memory words	
(2, 0)			$27\ 305_8$		16
	SORTIT		164_8	Prepares the input to SORT2	
		SORT2	207_8	A routine for calling SORT/MERGE using RECIN and RECOUT (Data is ordered according to program options.)	
(3, 0)			$13\ 060_8$		17
	COUPLE		$13\ 060_8$	Manages lower level overlays	
(3, 1)			$31\ 712_8$		17
	MATCAL		23_8	Manages the subroutines GRIDIT and REGRID	
		GRIDIT	$17\ 160_8$	Evaluates the gradient at each control point and evaluates the depth matrix	18
		REGRID	$2\ 260_8$	Checks each element of the depth matrix; fills in the missing elements; smooths the matrix	25
		CONVERT	16_8	Converts the units of the input data to the plot equivalent in inches	33
		TESTI	251_8	Examines the boundary elements of the depth matrix and reduces the matrix dimensions when all the elements in a column or row are missing	31
		CKPOINT	107_8	Checks data for control points which have been observed more than once	30
(3, 3)			$27\ 061_8$		34
	CKZFIT		$12\ 634_8$	Compares each scalar variation predicted by the depth matrix at the location of each control point with its control-point counterpart	
		DISCOT	555_8	Interpolates between depth-matrix elements to evaluate the predicted scalar variation at the location of each control point	
		SUB1	$5\ 155_8$	Compares the predicted scalar variation with its control-point counterpart and plots a frequency distribution curve on the listing	36

TABLE I.- FUNCTION AND STORAGE OF EACH OVERLAID PROGRAM AND SUBROUTINE – Concluded

Overlay	Program	Subroutine	Storage	Description	Page
(3, 2)			$25\ 056_8$		39
	AUTOCON		$3\ 351_8$	Determines grid spacing and plots origin, as well as controls the orderly processing of the columns and rows of depth matrix and controls the orderly call to each subroutine	
		GRID	117_8	A CalComp modular routine for plotting a grid of equally spaced lines	
		CALPLT	154_8	Basic CalComp instruction to plotter pen; establishes plot origin and pen position (up or down) before, during, and after changing locations	
		LINE1	223_8	Determines and stores the coordinates of the intercept of each contour line with side 1 of a triangle	47
		LINE2	226_8	Determines and stores the coordinates of the intercept of each contour line with side 2 of a triangle	48
		LINE3	245_8	Determines and stores the coordinates of the intercept of each contour line with the common side (side 3) of adjacent triangles	49
		LINE4	230_8	Determines and stores the coordinates of the intercept of each contour line with side 4 of the adjacent triangle	50
		LINE5	223_8	Determines and stores the coordinates of the intercept of each contour line with side 5 of the adjacent triangle	51
		PLOT	$1\ 366_8$	Plots each line segment connecting corresponding contour intercepts on opposite sides of a triangle and calls NUMBER	52
		NUMBER	246_8	Labels those contour lines which terminate at the boundaries of the plot	
		LEROY	25_8	Slows plotter for Leroy pen	
		CALCOMP	$1\ 056_8$	Selects CalComp plotting system	
		COLUMN	163_8	Right-justifies BCD number	53
		AXES	510_8	Plots and annotates axes	

TABLE II.- LIST OF VARIABLE NAMES AND DEFINITIONS

FORTRAN variable name	Card type	Definition
FRMT	1	Variable format for reading data on card types 3 and 4
KMAX	2	Integer specifying the number of control-point gradients required to compute an elevation at a particular grid point
KMAX1	2	Integer specifying the number of control points required to compute a gradient at a particular control point
IDMX	2	Integer specifying the maximum difference between scalar variations
DIST	3	Floating-point variable specifying the spacing between elements of the depth matrix
UNIT	3	Floating-point variable for converting the units of DIST into the smallest unit for which a conversion factor is available for converting UNIT to inches
CNVTOIN	3	Floating-point conversion factor for converting UNIT to inches
SPP	3	Floating-point interpolation factor for subdividing the submatrices in AUTOCON
CI	4	Contour interval
SCLE	4	Scale of the graphic
DISPGRD	4	Desired separation between grid lines of the graphic
TANG	4	Maximum anticipated slope between control points
ISRT	5	Code specifying whether data are to be sorted +1 sort data -1 do not sort data
MINI	5	Minimum anticipated scalar variation
MAXI	5	Maximum anticipated scalar variation
IRDTP	6	A code specifying whether control-point data are on tape or punched cards +1 data are on tape file 9 -1 data are on punched cards

TABLE II.- LIST OF VARIABLE NAMES AND DEFINITIONS – Concluded

<table>
<tr><th>FORTRAN variable name</th><th>Card type</th><th>Definition</th></tr>
<tr><td>IPTOPT</td><td>6</td><td>A code indicating type of data
+1 data are fixed point
-1 data are floating point</td></tr>
<tr><td>ISRTOPT</td><td>6</td><td>A code indicating primary and secondary variables in SORT/MERGE routines
<table>
<tr><th>Code</th><th>Variable</th><th>Direction</th><th>Order</th></tr>
<tr><td>-1</td><td>X</td><td>Ascending</td><td>Secondary</td></tr>
<tr><td></td><td>Y</td><td>Descending</td><td>Primary</td></tr>
<tr><td>0</td><td>X</td><td>Ascending</td><td>Secondary</td></tr>
<tr><td></td><td>IDIST</td><td>Descending</td><td>Primary</td></tr>
<tr><td>1</td><td>IDIST</td><td>Ascending</td><td>Primary</td></tr>
<tr><td></td><td>Y</td><td>Descending</td><td>Secondary</td></tr>
<tr><td>2</td><td>Y</td><td>Descending</td><td>Secondary</td></tr>
<tr><td></td><td>X</td><td>Ascending</td><td>Primary</td></tr>
</table>
</td></tr>
<tr><td>FRMT</td><td>7</td><td>A variable format for reading X, Y, and Z parameters on control-point data cards</td></tr>
</table>

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM

IX	IY	IZ	IX	IY	IZ
4000	0	-445	6763	56428	-957
4100	1200	-460	6903	57629	-981
4159	2401	-471	6919	58829	-986
4171	3602	-479	6939	60029	-992
4311	4802	-498	7979	0	-508
4327	6002	-504	7987	1200	-540
4347	7203	-511	7999	2402	-571
4387	8404	-520	8079	3603	-597
4395	9605	-526	8239	4803	-619
4407	10805	-532	8339	6003	-644
4487	12005	-546	8399	7204	-670
4647	13206	-565	8411	8405	-698
4747	14407	-580	8551	9605	-722
4807	15607	-591	8567	10805	-750
4819	16807	-596	8587	12006	-777
4959	18008	-615	8627	13207	-803
4975	19209	-620	8635	14408	-828
4995	20410	-626	8647	15608	-851
5035	21610	-635	8727	16809	-875
5043	22811	-640	8887	18010	-900
5055	24012	-646	8987	19210	-925
5135	25213	-659	9047	20410	-950
5295	26413	-680	9059	21610	-976
5395	27613	-693	9159	22811	-1006
5455	28814	-701	9215	24013	-1032
5467	30015	-703	9235	25213	-1054
5607	31216	-722	9275	26413	-1069
5623	32416	-725	9283	27614	-1081
5643	33616	-729	9295	28815	-1092
5683	34817	-735	9375	30016	-1100
5691	36018	-737	9535	31216	-1135
5703	37218	-746	9635	32416	-1157
5783	38419	-760	9695	33617	-1175
5943	39619	-787	9707	34818	-1188
6043	40820	-805	9847	36019	-1213
6103	42021	-819	9863	37219	-1232
6115	43221	-825	9883	38419	-1252
6255	44422	-850	9923	39620	-1274
6271	45623	-856	9931	40821	-1290
6291	46823	-863	9943	42021	-1307
6331	48023	-872	10023	43221	-1331
6339	49223	-876	10183	44422	-1365
6351	50424	-881	10283	45623	-1390
6431	51626	-896	10343	46824	-1411
6591	52826	-922	10355	48024	-1424
6691	54027	-940	10495	49225	-1455
6751	55227	-952	10511	50426	-1469
			10531	51626	-1482

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM – Continued

IX	IY	IZ	IX	IY	IZ
10571	52827	-1499	14523	50425	-1871
10579	54027	-1510	14583	51626	-1903
10591	55227	-1522	14595	52826	-1930
10671	56429	-1544	14735	54027	-1967
10831	57630	-1578	14751	55228	-1993
10931	58830	-1603	14771	56428	-2018
10991	60030	-1621	14911	57628	-2045
12003	0	92	14819	58828	-2068
12143	1201	72	14831	60029	-2090
12159	2401	46	15911	0	327
12179	3601	2	16071	1201	319
12219	4801	-43	16171	2402	307
12227	6002	-92	16231	3602	295
12239	7204	-141	16243	4802	274
12319	8404	-188	16383	6003	246
12479	9604	-223	16399	7204	214
12579	10805	-267	16419	8404	184
12639	12006	-324	16459	9604	154
12651	13207	-394	16467	10805	125
12791	14407	-456	16479	12006	96
12807	15607	-531	16559	13207	70
12827	16808	-603	16719	14407	48
12867	18009	-670	16819	15608	19
12875	19210	-739	16879	16809	-20
12887	20410	-804	16891	18009	-65
12957	21611	-859	17031	19209	-102
13127	22811	-905	17047	20409	-148
13227	24012	-957	17067	21610	-216
13287	25212	-1011	17107	22812	-280
13299	26412	-1045	17115	24012	-355
13439	27613	-1110	17127	25212	-442
13455	28814	-1161	17207	26413	-513
13475	30015	-1209	17367	27614	-579
13515	31215	-1254	17467	28815	-656
13523	32416	-1297	17527	30015	-737
13535	33617	-1339	17539	31215	-822
13615	34817	-1378	17679	32416	-881
13775	36017	-1416	17695	33617	-958
13875	37219	-1460	17715	34818	-1030
13935	38419	-1504	17755	36018	-1095
13967	39620	-1546	17763	37218	-1166
14087	40821	-1589	17775	38419	-1234
14103	42021	-1627	17855	39620	-1289
14123	43221	-1666	18015	40820	-1328
14163	44422	-1701	18115	42020	-1373
14171	45623	-1733	18175	43221	-1421
14183	46823	-1764	18187	44422	-1474
14263	48023	-1798	18327	45623	-1506
14423	49224	-1837	18343	46823	-1552

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM – Continued

IX	IY	IZ	IX	IY	IZ
18363	48024	-1595	22255	45622	-1307
19403	49225	-1634	22355	46822	-1382
18411	50426	-1675	22415	48023	-1456
18423	51626	-1714	22427	49224	-1526
18503	52826	-1741	22567	50425	-1598
18663	54027	-1757	22583	51625	-1663
18763	55228	-1779	22603	52826	-1725
18823	56429	-1805	22643	54027	-1786
18835	57629	-1837	22651	55227	-1843
18975	58829	-1849	22663	56427	-1897
18991	60030	-1878	22743	57627	-1956
20011	0	216	22903	58828	-2022
20051	1200	201	23003	60030	-2092
20059	2400	188	24063	0	-220
20071	3600	174	24075	1200	-206
20151	4801	156	24215	2400	-204
20311	6003	140	24231	3601	-192
20411	7203	128	24251	4802	-182
20471	8403	117	24291	6003	-165
20483	9604	102	24299	7203	-152
20623	10805	83	24311	8403	-145
20639	12006	69	24391	9604	-144
20659	13206	54	24551	10805	-150
20699	14406	38	24651	12006	-153
20707	15607	23	24711	13206	-155
20719	16808	3	24723	14406	-156
20799	18009	-20	24863	15607	-162
20959	19209	-46	24879	16808	-161
21059	20409	-68	24899	18008	-162
21119	21610	-92	24939	19208	-171
21131	22811	-120	24947	20409	-181
21271	24011	-154	24959	21610	-192
21287	25211	-189	25039	22811	-205
21307	26412	-221	25199	24011	-221
21347	27613	-260	25299	25212	-235
21355	28814	-306	25359	26413	-256
21367	30014	-350	25371	27613	-281
21447	31215	-396	25511	28813	-308
21607	32416	-446	25527	30013	-333
21707	33616	-497	25547	31214	-360
21767	34816	-554	25587	32616	-400
21779	36017	-618	25595	33617	-440
21919	37217	-686	25607	34817	-478
21935	38419	-786	25687	36017	-536
21955	39620	-885	25847	37218	-596
21995	40820	-976	25947	38419	-658
22003	42020	-1065	26007	39619	-723
22015	43221	-1149	26019	40819	-788
22095	44422	-1230	26159	42020	-850

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM – Continued

IX	IY	IZ	IX	IY	IZ
26175	43221	-925	29927	40819	-765
26195	44422	-997	30087	42020	-930
26235	45622	-1094	30187	43221	-910
26243	46823	-1187	30247	44421	-989
26255	48024	-1276	30259	45621	-1068
26335	49224	-1357	30399	46822	-1153
26495	50424	-1427	30415	48023	-1236
26595	51624	-1493	30435	49224	-1326
26655	52825	-1559	30475	50424	-1414
26667	54027	-1625	30483	51625	-1502
26807	55227	-1677	30495	52825	-1589
26823	56427	-1736	30575	54027	-1676
26943	57628	-1792	30735	55227	-1765
26983	58829	-1844	30935	56427	-1849
26991	60030	-1896	30895	57628	-1933
27003	0	-727	30997	58829	-2006
27983	1200	-700	31047	60031	-2089
28143	2400	-585	32063	0	-1467
28243	3601	-444	32083	1200	-1381
28303	4802	-539	32123	2400	-1303
28315	6003	-509	32131	3601	-1233
28455	7203	-572	32143	4802	-1177
28471	8403	-534	32223	6002	-1117
28491	9604	-409	32383	7202	-1069
28531	10805	-470	32483	8403	-1015
28539	12005	-441	32543	9604	-956
28551	13205	-414	32555	10805	-898
28631	14406	-305	32695	12005	-859
28791	15607	-384	32711	13205	-808
28891	16809	-378	32731	14406	-760
28951	18008	-371	32771	15607	-712
28963	19209	-363	32779	16807	-659
29103	20410	-361	32791	18007	-609
29119	21610	-356	32871	19209	-567
29139	22810	-353	33031	20410	-534
29179	24010	-351	33131	21610	-503
29187	25211	-349	33191	22810	-486
29199	26413	-358	33203	24011	-479
29279	27613	-371	33343	25212	-477
29429	28814	-386	33359	26413	-472
29539	30014	-400	33379	27613	-468
29599	31215	-414	33419	28813	-466
29611	32416	-428	33427	30014	-464
29751	33616	-458	33439	31215	-464
29767	34816	-493	33519	32416	-474
29787	36017	-535	33579	33616	-491
29827	37218	-580	33779	34816	-508
29935	38419	-639	33839	36017	-542
29947	39619	-703	33851	37218	-582

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM – Continued

IX	IY	IZ	IX	IY	IZ
33991	38418	-624	37679	36018	-570
34007	39618	-698	37759	37219	-412
34027	40819	-766	37919	38419	-672
34067	42020	-932	38019	39619	-732
34075	43221	-897	38079	40820	-796
34087	44421	-985	38091	42021	-837
34167	45622	-1067	38231	43221	-891
34227	46823	-1144	38247	44421	-975
34427	48023	-1237	38267	45622	-1054
34497	49223	-1341	38307	46823	-1127
34499	50423	-1440	38315	48024	-1197
34639	51624	-1540	38327	49224	-1310
34655	52826	-1640	38407	50425	-1418
34675	54027	-1739	38567	51626	-1519
34715	55227	-1847	38667	52826	-1633
34723	56427	-1950	38727	54027	-1758
34735	57628	-2049	38739	55227	-1880
34815	58829	-2146	38879	56428	-2010
34975	60029	-2242	38895	57629	-2145
36075	0	-2554	38915	58830	-2272
36135	1200	-2425	38955	60030	-2391
36147	2400	-2289	40055	2401	-3554
36287	3602	-2195	40215	3602	-3413
36303	4803	-2072	40315	4802	-3256
36323	6003	-1937	40375	6002	-3072
36363	7203	-1809	40387	7203	-2877
36371	8404	-1680	40527	8404	-2728
36383	9605	-1559	40543	9603	-2550
36463	10805	-1478	40563	10805	-2382
36623	12005	-1414	40603	12005	-2228
36723	13206	-1344	40611	13206	-2073
36783	14407	-1271	40623	14407	-1928
36795	15608	-1186	40703	15607	-1799
36935	16808	-1115	40853	16807	-1593
36951	18009	-1030	40963	18008	-1580
36971	19210	-950	41023	19209	-1466
37011	20410	-882	41035	20410	-1359
37019	21610	-923	41175	21610	-1291
37031	22810	-769	41191	22811	-1211
37111	24011	-725	41211	24012	-1137
37271	25213	-688	41251	25213	-1067
37371	26413	-643	41259	26413	-987
37431	27613	-597	41271	27613	-912
37443	28814	-580	41351	28814	-847
37583	30015	-570	41511	30015	-791
37599	31216	-563	41611	31216	-744
37619	32416	-557	41671	32416	-697
37659	33616	-553	41683	33616	-685
37667	34817	-549	41823	34817	-679

TABLE III.- TABULATED CONTROL POINT DATA FOR SAMPLE PROBLEM – Concluded

IX	IY	IZ	IX	IY	IZ
41839	36018	-700	46159	45623	-1118
41859	37218	-732	46239	46824	-1190
41899	38418	-766	46399	48024	-1263
41907	39619	-801	46499	49225	-1334
41919	40820	-830	46559	50426	-1402
41909	42021	-986	46571	51626	-1462
42159	43221	-956	46711	52827	-1545
42259	44422	-1025	46727	54027	-1625
42319	45623	-1086	46747	55227	-1703
42331	46823	-1140	46787	56429	-1778
42471	48023	-1204	46795	57630	-1987
42487	49223	-1288	46807	58830	-2188
42507	50424	-1366	46887	60030	-2377
42547	51626	-1437	48867	16808	-3545
42555	52825	-1532	49007	18009	-3294
42567	54027	-1685	49023	19210	-3019
42647	55227	-1830	49043	20410	-2775
42307	56428	-1971	49083	21610	-2582
42907	57629	-2121	49091	22811	-2392
42967	58829	-2260	49103	24012	-2214
42979	60029	-2415	49183	25212	-2082
44615	10805	-3539	49243	26412	-1951
44627	12006	-3313	49443	27613	-1818
44767	13207	-3133	49503	28814	-1685
44783	14408	-2927	49515	30015	-1554
44903	15608	-2715	49655	31215	-1463
44943	16809	-2500	49671	32416	-1365
44951	18010	-2288	49691	33617	-1274
44863	19210	-2085	49731	34817	-1191
44943	20410	-1927	49739	36017	-1152
45103	21610	-1836	49751	37218	-1133
45203	22811	-1736	49931	38418	-1126
45263	24013	-1639	49991	39620	-1130
45275	25213	-1537	50091	40821	-1147
45415	26413	-1429	50151	42021	-1176
45431	27614	-1331	50163	43221	-1206
45451	28815	-1230	50303	44422	-1243
45491	30016	-1136	50319	45623	-1267
45409	31215	-1059	50339	46823	-1284
45511	32416	-987	50379	48023	-1304
45501	33617	-926	50387	49224	-1346
45751	34818	-874	50399	50425	-1434
45851	36019	-858	50479	51626	-1514
45911	37219	-877	50639	52826	-1594
45923	38419	-911	50739	54027	-1672
45963	39620	-954	50799	55228	-1749
45979	40821	-994	50911	56428	-1938
46059	42021	-1012	50951	57628	-1925
46139	43221	-1042	50967	58828	-2009
46147	44422	-1073	50987	60029	-2091

TABLE IV.- DEFINITIONS AND MATHEMATICAL EQUIVALENTS

OF MAJOR FORTRAN VARIABLE NAMES[a]

Primary FORTRAN name	Equivalent FORTRAN names	Mathematical equivalent	Definition
A,B,C		A_i,B_i,C_i	Arrays containing coefficients of gradient vector
ARG		H_i,Z_j	Array containing real and predicted scalar variations
BB		$\bar{N}_i$	Array representing estimated gradient
CLASS2		U_j	Array defining categories of discrepancies
CONT			Array of contour lines intersecting side 1
CONTA			Array of contour lines intersecting side 2
CONTB			Array of contour lines intersecting side 3
CONTC			Array of contour lines intersecting side 4
CONTD			Array of contour lines intersecting side 5
COUNT			Array containing a count of discrepancies (frequency of occurrence)
D			Array for storing in memory all the control-point data
DEPTH			Array within which the 2SPP + 1 by 2SPP + 1 elements of the submatrix are stored
E		$\bar{Q}_j$	Array of vectorial components of a vector approximating gradient
GRAD			Array containing components of a vector approximating the gradient
I	MAT,IEL	H_i	Depth-matrix array

[a] Variables in table II are not listed except where mathematical equivalents are given.

TABLE IV.- DEFINITIONS AND MATHEMATICAL EQUIVALENTS

OF MAJOR FORTRAN VARIABLE NAMES[a] – Continued

Primary FORTRAN name	Equivalent FORTRAN names	Mathematical equivalent	Definition
IADDI			Constant bias added to I
ICODE			Signifies end of data
IDEPH			Array for storing three rows of I matrix
IDIST			Distance of control data point from a common origin
IFN			Array containing location of input and output files for SORT2
ISM			An array containing information regarding input record structure and number of input files and of variables to be ordered
IX	X	x_j	X-coordinate of control-point data
IY	Y	y_j	Y-coordinate of control-point data
IZ	Z	z_j	Z-coordinate of control-point data
KEY			Array containing location and type of variable to be ordered and direction of sort
KORE			Locations of the X-coordinates of each corner of I matrix
KORN			Locations of the Y-coordinates of each corner of I matrix
KM	JJ,MAT		Number of rows in I matrix
KMAX		k'	Number of neighboring control points

[a] Variables in table II are not listed except where mathematical equivalents are given.

TABLE IV.- DEFINITIONS AND MATHEMATICAL EQUIVALENTS OF MAJOR FORTRAN VARIABLE NAMES[a] – Continued

Primary FORTRAN name	Equivalent FORTRAN names	Mathematical equivalent	Definition
KMAX1		k	Number of neighboring control points
LM	KK,MIT		Number of columns in I matrix
M			Number of unfilled I elements
MAXWRD2			Maximum IY
MAXWRD3			Maximum IX
MINWRD2			Minimum IY
MINWRD3			Minimum IX
POS1X			Locations of X-intercepts with side 1
POS1Y			Locations of Y intercepts with side 1
POS2X			Locations of X intercepts with side 2
POS2Y			Locations of Y intercepts with side 2
POS3X			Locations of X intercepts with side 3
POS3Y			Locations of Y intercepts with side 3
POS4X			Locations of X intercepts with side 4
POS4Y			Locations of Y intercepts with side 4
POS5X			Locations of X intercepts with side 5
POS5Y			Locations of Y intercepts with side 5
R		r_l, r_j	Radial distance between two points

[a] Variables in table II are not listed except where mathematical equivalents are given.

TABLE IV.- DEFINITIONS AND MATHEMATICAL EQUIVALENTS OF MAJOR FORTRAN VARIABLE NAMES[a] – Concluded

Primary FORTRAN name	Equivalent FORTRAN names	Mathematical equivalents	Definition
RLAST			Inner radius of ring
RMAX		r_k	Maximum distance of data point from arbitrary origin
RSAVE			Outer radius of ring
SP			Distance between elements of I at plotting scale
SQX			X-coordinate of submatrix center
SQY			Y-coordinate of submatrix center
W		w_j	Weight
XMAX			X-dimension of I
YMAX			Y-dimension of I

[a] Variables in table II are not listed except where mathematical equivalents are given.

NASA-Langley, 1971 —— 8 L-7273

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546