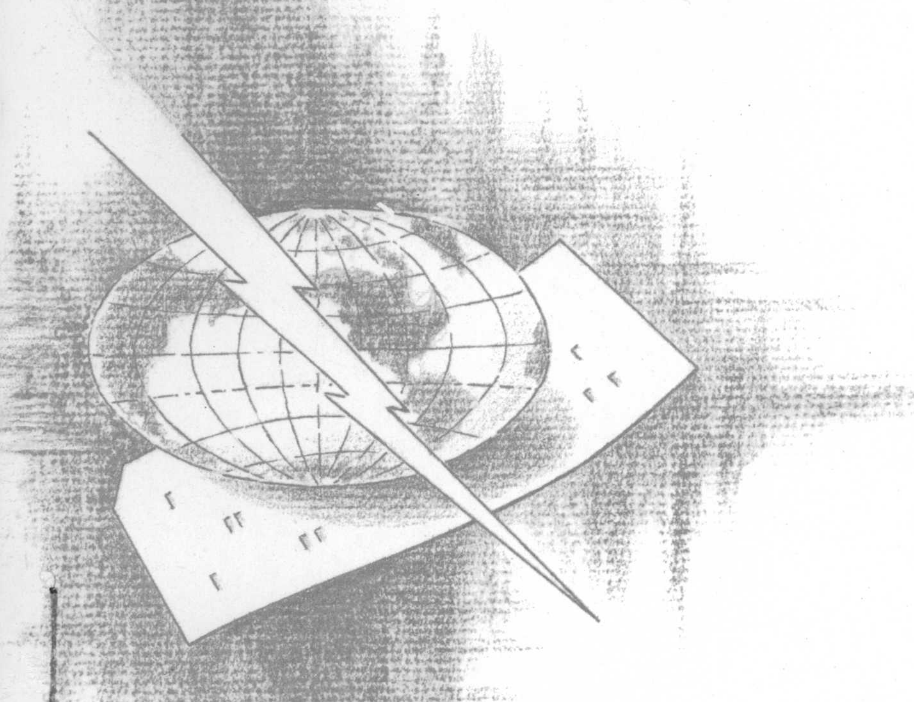
GEOLOGICAL SURVEY
COMPUTER CONTRIBUTION



Programs Avail

GEOLOGICAL SURVEY
COMPUTER CONTRIBUTION

USGS-CCD-73-001

Number 22

Generalized Gradient and Contour Program

by

Marshall Strong Hellmann

Computer Center Division
Washington, D. C.
1972

Program Number: E874
Operating System: IBM System 360 - Release 20
Language: FORTRAN IV (H-Level)
Equipment: IBM 360/65
CalComp 7900 Flatbed Plotting System
CalComp 763 Drum Plotter and 770 Tape Drive

COMPUTER CONTRIBUTIONS

Computer Contributions are available ($0.95 for microfiche copy, $3.00 for paper copy) from the National Technical Information Service, Springfield, Virginia 22151. When ordering, refer to the title and the PB number.

1. Weighted Triangulation Adjustment, by Walter L. Anderson, 1969 (PB-203 580)
2. Perspective Center Determination, by John D. McLaurin, 1969 (PB-195 999)
3. Nonconstant Variance Regression Analysis Program, by Marshall Strong Hellmann, 1970 (PB-203 568)
4. Hot Pipe, by Patrick C. Doherty, 1970 (PB-206 417)
5. Proceedings of the Symposium on Map and Chart Digitizing, 1970 (PB-190 628)
6. Atlantic Region Semianalytical Aerotriangulation Program, by Leonard R. Mannello, Morris L. McKenzie and Robert Montgomery, Jr., 1970 (PB-194 570)
7. Mineral Distribution Program - Mod II (for solving Petrologic Mixing Problems), by Patrick C. Doherty and Thomas L. Wright, 1971 (PB-203 581)
8. Generation and Testing of Random Numbers, by Nicholas C. Matalas and Marshall Strong Hellmann, 1971 (PB-203 140)
9. X-ray Diffractometer Data Reduction Program, by Marshall Strong Hellmann, 1971 (PB-209 838)
10. Iterative Three-Dimensional Solution of Gravity Anomaly Data, by Lindrith Cordell, 1970 (PB-196 979)
11. The STATPAC System, by Fred B. Sower, Ralph N. Eicher and Gary I. Selner, 1971 (PB-203 787)
12. Transient Flow, by Marshall Strong Hellmann, 1971 (PB-209 839)
13. Graphic Normative Analysis Program, by Roger W. Bowen, 1971 (PB-206 736)
14. Calculation of Pseudo-gravity Anomaly from Total Magnetic Intensity Data, by Lindrith Cordell, 1971 (PB-203 726)

15. Solution of Combined Gravity and Magnetic Data for Physical Properties, by Lindrith Cordell and Patrick T. Taylor, 1971 (PB-203 725)

16. Calculation of Amphibole Structural Formulae, by Fraser E. Goff and Gerald K. Czamanske, 1972 (PB-207 710)

17. Program PRØFL3, A Generalized Profile Plotting Program, by A. Graig McHendrie, 1972 (PB-207 523)

18. A FORTRAN IV Computer Program for Calculating Borehole Gravity Terrain Corrections, by L. A. Beyer and C. E. Corbato, 1972 (PB-208 679)

19. The SYSLAB System for Data Analysis of Historical Water-quality Records (basic programs), by Timothy Doak Steele, 1972

20. Job 9214: Indexing and Least-squares Refinement of Powder Diffraction Data, by Daniel E. Appleman, Howard T. Evans, Jr., and David S. Handwerker, 1972

21. Rosenbrock Technique for Determining Greatest or Least Value of a Function, by P. H. Carrigan, 1972

22. Generalized Gradient and Contour Program, by Marshall Strong Hellmann, 1972

CONTENTS

TABLES

ATTACHMENTS

GENERALIZED GRADIENT AND CONTOUR PROGRAM

Marshall Strong Hellmann

ABSTRACT

This program computes estimates of gradients, prepares contour maps, and plots various sets of data provided by the user on the CalComp plotters. The gradients represent the maximum rates of change of a real variable Z=f(X,Y) with respect to the two-dimensional rectangle on which the function is defined. The contours are lines of equal Z values. The program also plots special line data sets provided by the user.

INTRODUCTION

The purpose of this program is to compute estimates of the rate of change of a real variable Z=f(X,Y) defined on a two-dimensional surface, and to generate contours of this variable. The variable Z could typically be the elevation of the ground above sea level. The user must describe the two-dimensional surface as a rectangle whose sides are parallel to the X and Y axes by providing the limiting X and Y coordinates of the rectangle.

If the option indicator is ≠ 0-4, then only the special line data will be processed. (See Table 1, p. 3-5.) Otherwise the program requires that a lattice of grid lines be superimposed on the two-dimensional surface and that the value of Z be given at each of the lattice points defined by the intersection of the grid lines making up the lattice. Two basic sets of data are used by this program to produce gradient arrow maps, contour maps, or maps containing both arrows and contours as well as special lines and symbols.

The first set of data consists of the matrix of Z values at each of the lattice points. This set of data is used to compute both the gradients and the contours. The gradients are plotted on the map in the form of arrows pointing in the direction of these vectors. The contours are plotted as a series of connected line segments.

The second set of data consists of a collection of special line data sets. Each line data set consists of an ordered set of triplets (X,Y,Z), where Z is treated as the elevation variable and X and Y denote the horizontal and vertical coordinates of the point on the line, respectively. (See "Line Data Cards," p. 7 for details.) This second set of data can be used to plot special lines with different meanings.

Optionally, the program will generate contour lines of equal Z values on the map. The arrows plotted by the program will be approximately orthogonal to the contours generated by the automatic contouring feature of this program if the contour intervals are chosen appropriately.

INPUTS

Program Option Cards

These cards are used to specify the various options available to the user to make this program process his particular set of data. Two program control cards and three title cards are required as defined in the following two tables.

Table 1. -- Program Control Card Formats

Card No.	Columns	Format	Name/Meaning
1	1-12	E12.5	XMIN - X coordinate of left side of the map.
	13-24	E12.5	XMAX - X coordinate of right side of the map.
	25-36	E12.5	YMIN - Y coordinate of bottom of the map.
	37-48	E12.5	YMAX - Y coordinate of top of the map.
	49-54	F6.3	WIDTH - Width of the map in inches.
	55-60	F6.3	HEIGHT - Height of the map in inches.
	61-62	I2	Number of horizontal tick marks on the vertical axes. must be ≥2.
	63-64	I2	Number of vertical tick marks on the horizontal axes. must be ≥2.
	65-70	F6.3	FACTOR - Reduction factor for the size of the map.
	71	I1	GR LINE IND - (1-tick marks and lines, 2-tick marks only, 3-lines only, otherwise no graph lines or tick marks will be plotted.) Graph lines are drawn between tick marks if requested.

Card No./Columns		Format	Name/Meaning
1	72-75	F4.3	Z HT - Height of the numbers printed on the map on the contour line labels in inches.[1]
	76-77	I2	Z INCR - Frequency of points on contour lines to be plotted on the map.[1]
	78-80	I3	LINE TYPE Number of points on contour line between contour line labels.[1]
2	1	I1	Number of decimal places to right of decimal point in contour labels. (Used in Z FORMAT.)[1]
	2	I1	Number of decimal places to left of decimal point in contour labels. (Used in Z FORMAT.)[1]
	3	I1	Number of decimal places to the right of the decimal point in all X and Y coordinate labels. (Used in XY FORMAT and XY FORMAT 2.)
	4	I1	Number of decimal places to the left of the decimal point in the X and Y coordinate labels at the corners of the map. (Used in XY FORMAT.)
	5	I1	Number of decimal places to the left of the decimal point in the X and Y coordinate labels at interior tick marks on the map. (Used in XY FORMAT 2.)
	6-8	I3	SPEC CHAR CODE (Special character Code). - See (CalComp, 1970). Corresponding symbol is plotted on contour line by contour labels.[1]

Card No./Columns		Format	Name/Meaning
2	9-11	I3	Number of horizontal grid lines in lattice, (n).
	12-14	I3	Number of vertical grid lines in lattice, (m).
	15-26	E12.5	GRID DISTANCE (Horizontal) - Distance between lattice points in horizontal direction.
	27-38	E12.5	GRID DISTANCE (Vertical) - Distance between lattice points in vertical direction.
	39-50	E12.5	GRADIENT MAP SCALING FACTOR - This scaling factor should be chosen so that gradient values are in the range 0 to 9.999.
	51-54	F4.3	NUMBER SIZE FACTOR - This factor is used to compute the height of the numbers plotted above the arrows on the map. Appropriate values are in the range 0. to .3 depending on desired results. A value of 0 means no number will be plotted. See attachment A.
	55	I1	OPTIONS INDICATOR - (See p. 11) 0-generate contours. 1-compute gradients on T_1. 2-compute gradients on T_1 and T_2. 3-compute gradients on T_1 and generate contours. 4-compute gradients on T_1 and T_2 and generate contours. Special line data sets can be included for all options.
	56-58	I3	CONTOUR LABELING FREQUENCY - Frequency of contours to be labeled.[1]
	59-70	E12.5	CONTOUR INTERVAL - (See "Automatic Contour Option," p. 15.)

[1] See "PLOTTING OPTIONS," p. 31, of this paper.

Table 2.--Title Cards Format

Card No.	Columns	Format	Meaning
3	1- 2	I2	Number of characters in map title.
	3-74	18A4	Map title.
4	1- 2	I2	Number of characters in X axis title.
	3-74	18A4	X axis title.
5	1- 2	I2	Number of characters in Y axis title.
	3-74	18A4	Y axis title.

Z Variable Input Matrix

This input matrix must contain the values of the Z variable at each of the lattice points of the lattice superimposed on the map. In order to allow the user flexibility in preparing the input data the format in which the data is stored is read into the program from a single card. This card is read in the format 20A4. It should contain the format in which the records containing the Z variable matrix are stored enclosed by parentheses. The Z variable matrix is read row by row. Each row must have m values where m is the number of vertical grid lines in the lattice punched in columns 12-14 of Program Control Card Number 2. The rows are numbered from top to bottom and the columns from left to right on the map or rectangular domain. The Z variable matrix must be stored in this manner with each row starting on a new card or record, and successive values for that row stored as dictated by the variable format card. There must be n rows of values, where n is the number of horizontal grid lines in the lattice punched in columns 9-11 of Program Control Card Number 2. The Z variable input matrix is read from the data set with the DDNAME of FT10F001.

Line Data Cards

This set of data cards contains the special line data. The ends of individual line sets are indicated by end-of-line cards containing a non-zero value in column 37. The data in columns 37-55 determine the way the line will be plotted. The sets of line data cards are placed one after another followed by a /* card. Columns 1-36 are read on the end-of-line cards but the values are not used. All of these cards have the format shown in Table 3.

Table 3.--Line Data Format

Columns	Format	Meaning
1-12	E12.5	Z coordinate of this line point.
13-24	E12.5	X coordinate of this line point.
25-36	E12.5	Y coordinate of this line point.
37	I1	End-of-line card and pen number.[1] See (CalComp, 1970).
38-40	I3	Line Type-number of points on this line between labels.[1]
41	I1	Number of decimal places to the right of the decimal point in the line point labels.[1]
42	I1	Number of digits to the left of the decimal point in the line point labels.[1]
43-45	I3	Special Character Code[1] See (CalComp, 1970).
46-49	F4.3	Height of symbols to be plotted.[1]
50-53	F4.3	Height of numbers to be plotted.[1]
54-55	I2	Frequency of points on special line data set to be plotted on map.[1]

[1] See "PLOTTING OPTIONS," p. 31, of this paper.

OUTPUTS

This program generates the following sets of output depending on the options chosen by the user.

A printout of the input parameters titled "GENERALIZED GRADIENT AND CONTOUR PROGRAM INPUT PARAMETERS". This printout contains a listing identifying the input parameters contained on the two program control cards.

A printout titled "INPUT DATA MATRIX". This printout contains a listing of the input data for each of the grid points in the rectangular area defined on the program control cards. The listing is printed with the values in each row starting at XMIN and ending at XMAX. The rows are printed starting at YMAX and ending at YMIN and are separated by blank lines. This matrix is printed in the format 10E13.5.

Two printouts titled "GRADIENT MATRIX 1" and "GRADIENT MATRIX 2" may be printed by the program. The program prints the first gradient matrix only if the program options indicator on Program Control Card Number 2 contains a 0-4. The second matrix is printed only if the program options indicator is a 2 or 4. The values in "GRADIENT MATRIX 1" correspond to the magnitudes of the gradients on the set of triangles T_1 and those in "GRADIENT MATRIX 2" to T_2.

Two printouts titled "GRADIENT ANGLE MATRIX 1" and "GRADIENT ANGLE MATRIX 2" may also be printed. Once again the value of the program options indicator determines if these matricies may be printed. The values in the first of these two matricies correspond to the gradients on the triangles T_1 and the second to those in T_2.

A printout titled "AUTOMATIC CONTOUR INTERVAL GENERATION PARAMETERS" is printed if the program generates contours automatically. This printout shows the values on the Automatic Contour Interval Generation Parameters (ACIGP) card. (See p. 15.)

A printout titled "ADJUSTED INPUT DATA MATRIX" may be printed if the elevation adjustment factor is not equal to 0 on the ACIGP card. (See p. 19.)

A printout titled "HORIZONTAL GRID COORDINATES" and "VERTICAL GRID COORDINATES" may be printed. If so, it contains horizontal and vertical grid coordinates for each of the lattice points in the grid used by the program. (See p. 14.)

The plotting output is written on a tape for later use on the CalComp plotter. This tape can be prepared for either the CalComp drum or flatbed plotters. (See SYSNØTES, 1972.)

The user has the option of recording the values of the gradient magnitudes and angles on a data set for subsequent use. If this option is chosen these values are written using the formats (5E16.8) and (8F10.5) for the magnitudes and angles, respectively. (See attachment E., p. 86.)

In addition to the above printouts, error messages may be printed in the output if certain conditions arise. (See "DIAGNOSTIC MESSAGES," p. 25.)

TECHNICAL DESCRIPTION

The gradients can be computed by the following technique. Let the map be divided into (m-1)x(n-1) rectangles by superimposing a lattice of m vertical lines and n horizontal lines equally spaced in the horizontal and vertical directions, respectively. The program allows for the horizontal and vertical scales to be different. The actual distances between grid lines in the horizontal and vertical directions are denoted by S_x and S_y, respectively. (See Table 1, p. 3-5.)

If the Z variable is not known at the grid points in this lattice, then it will be necessary to use a method such as two-dimensional linear interpolation,between some of the known values of Z to determine the values of the Z variable at the grid points.[1]

The horizontal and vertical grid coordinates of the grid points P_{ij} are given by, X_i=XMIN+(XMAX-XMIN)(i-1)/(m-1) and by, Y_j=YMAX-(YMAX-YMIN)(j-1)/(n-1) for i=1,2,...m and j=1,2,...n, respectively.

The function f(X,Y)=Z is assumed to be a bounded, single-valued, and continuous function on the rectangular domain of the map. The surface defined by the function f(X,Y) can be approximated by a surface consisting of triangular sections of planes in the manner described below. This approximation

---

[1] The United States Geological Survey (U.S.G.S.) program E867 (Linear Grid Generation Program) is also available for determining the Z values at the grid points, given an arbitrary set of (X,Y,Z) data values. The mesh or distance between these arbitrary data points must be sufficiently small to insure that the generated values are as accurate as required.

surface can be made to approximate the true surface within any positive tolerance specified by decreasing the grid size sufficiently. Let the given values of $f(X,Y)$ at each of the grid points P_{ij} be denoted by H_{ij} and on the surface of $f(X,Y)$ as $\{(X_i, Y_j, H_{ij})$, for $i=1,2,\ldots m$ and $j=1,2,\ldots n.\}$

The domain of the function $f(X,Y)$ is divided into two sets of triangles, T_1 and T_2. Set $T_k=\{T_{kij}:T_{kij}=[(X,Y):(X,Y)$ is contained in the triangle whose verticies are $(X_{(i+k-1)}, Y_{(j+k-1)})$, $(X_{(i+2-k)}, Y_{(j+k-1)})$, and $(X_{(i+k-1)}, Y_{(j+2-k)})]$ for $i=1,2,\ldots m-1$ and $j=1,2,\ldots n-1\}$ for $k=1,2$.

The approximation surface is defined by the following two sets, S_1 and S_2 of triangular planar sections. Set $S_k=\{S_{kij}:S_{kij}=[(X,Y,H):(X,Y,H)$ is contained in the triangular section of the plane whose verticies are $(X_{(i+k-1)}, Y_{(j+k-1)}, H_{(i+k-1)(j+k-1)})$, $(X_{(i+2-k)}, Y_{(j+k-1)}, H_{(i+2-k)(j+k-1)})$, and $(X_{(i+k-1)}, Y_{(j+2-k)}, H_{(i+k-1)(j+2-k)})]$ for $i=1,2,\ldots m-1$ and $j=1,2,\ldots n-1\}$ for $k=1,2$.

Define the function $g(X,Y)$ on T_{kij} such that if (X,Y) is contained in T_{kij}, then $g(X,Y)=H$, where (X,Y,H) is contained in S_{kij} for each T_{kij} contained in T_1 or T_2. The function $g(X,Y)$ is identical to $f(X,Y)$ at each of the grid points (X_i,Y_j) and varies linearly between any two points of a triangle S_{kij} on its surface for $i=1,2,\ldots m-1$ and $j=1,2,\ldots n-1$ and $k=1,2$. A direction tangent at any point P on the surface of $g(X,Y)$ is defined to be the tangent to that surface in the given direction if it exists. Since none of the S_{kij} are perpendicular to the (X,Y) plane, this tangent is finite, in both the X and Y directions.

The function $g(X,Y)$ on T_{kij} is given by

$$g(X,Y)=H^X_{kij}X+H^Y_{kij}Y+C_{kij}, \tag{1}$$

for each T_{kij} contained in T_1 or T_2 and

where: H^X_{kij} is the direction tangent of the plane S_{kij} in the X direction and is given by equation (2).

H^Y_{kij} is the direction tangent of the plane S_{kij} in the Y direction and is given by equation (3).

C_{kij} is a constant which determines the displacement of the plane S_{kij} from the origin in the Z direction.

The function $g(X,Y)$ has the following properties:

1. $g(X,Y)$ is single-valued and bounded for all (X,Y) contained in the domain of $f(X,Y)$.
2. $g(X,Y)$ is continuous throughout the domain of $f(X,Y)$.
3. $g(X,Y)$ is differentiable on the interior of each of the triangular domains T_{kij} contained in T_1 and T_2.
4. $g(X,Y)$ is not in general differentiable on the boundary of the triangles T_{kij} contained in T_1 and T_2.
5. The partial derivatives of $g(X,Y)$ with respect to X and Y are constant within the interior of T_{kij} for each T_{kij} contained in T_1 and T_2.
6. The gradients of $g(X,Y)$ defined on T_{kij} are identical for each point in the interior of T_{kij}, i.e., the interiors of T_{kij} are isogradient domains.

The horizontal gradient component of g(X,Y) on the interior of T_{kij} is given by

$$H^X_{kij}=\frac{H_{(i+1)(j+k-1)}-H_{i(j+k-1)}}{S_x} \qquad (2)$$

for each T_{kij} contained in T_1 or T_2.

The vertical gradient component of g(X,Y) on the interior of T_{kij} is given by

$$H^Y_{kij}=\frac{H_{(i+k-1)j}-H_{(i+k-1)(j+1)}}{S_y} \qquad (3)$$

for each T_{kij} contained in T_1 or T_2.

The magnitude of the gradient $\vec{G}_{kij}$ on the interior of T_{kij} is given by

$$|\vec{G}_{kij}|=[(H^X_{kij})^2+(H^Y_{kij})^2]^{.5} \qquad (4)$$

for each T_{kij} contained in T_1 or T_2.

The direction of the gradient $\vec{G}_{kij}$ on the interior of T_{kij} is given by

$$\phi_{kij}=\tan^{-1}(H^Y_{kij}/H^X_{kij}). \qquad (5)$$

The downhill or opposite direction of the gradient is given by adding 180° to ϕ_{kij}. The angles used and printed by this program are the opposite direction angles.

If both H^X_{kij} and H^Y_{kij} are zero on T_{kij} then the gradient vector $\vec{G}_{kij}$ degenerates to the null vector. This means that the function g(X,Y) is constant throughout the interior of the domain T_{kij}, i.e., the approximation surface is flat in that domain. This program assigns the values of 0 to $|\vec{G}_{kij}|$ and $-90°$ to ϕ_{kij} whenever: (1) k=1, and [(i=1 or m) or (j=1 or n)]; (2) k=2, and [(i=m-1 or m) or (j=n-1 or n)]; (3) $\vec{G}_{kij}$ is the null vector.

The X and Y coordinates are transformed to an equivalent set of coordinates in the range 0 to 1 for the purpose of generating contours more accurately. This transformation is effected only if the automatic contouring option is selected. The transformed X coordinates are given by X'=(X-XMIN)/(XMAX-XMIN). The transformed Y coordinates are given by Y'=(Y-YMIN)/(YMAX-YMIN). The transformed coordinates of the horizontal and vertical lines in the lattice are printed by the program only if the automatic contouring option is selected. The use of these transformed coordinates also simplifies the computational processes required to check the results manually. The values printed by the program can be transformed back to the original coordinates by using the inverse of these two transformations.

If the automatic contouring option is selected, then this program generates contours of the function g(X,Y) or a slightly modified form of g(X,Y) depending on the value of the ELEVATION ADJUSTMENT FACTOR. (See p. 16.)

AUTOMATIC CONTOUR OPTION

This option is an extension of the gradient program as originally written. It provides for automatic generation of contours of the Z variable input matrix. These contours are in addition to any contours, which may be generated by the line data cards described in "Line Data Cards," p. 7.

This option is effected by specifying either a 0, 3 or 4 in column 55 of Program Control Card Number 2. The contours generated will be the same in either case. If option 3 or 4 is selected, the program first computes the gradients and then generates the contour lines for selected values of the variable Z. The program provides the user with the option of selecting the contour values by several different methods using the Automatic Contour Interval Generation Parameters described below.

Table 4.--Automatic Contour Interval Generation Parameters

Columns	Format	Name/Meaning
1	I1	OPTION IND - Values should be 1-5. (See below.)
2- 4	I3	MAX NO OF CONTOURS - Must be in the range 1 to 100 unless the option indicator is not equal to 1-5. (See p. 16-18.)
5- 7	I3	CONTOUR INTERVAL FACTOR (Lower) - This value can range from -99 to 999 depending on the values of the Z variable input matrix.
8-10	I3	CONTOUR INTERVAL FACTOR (Upper) - This value can range from -99 to 999 but should be greater or equal to the lower contour interval factor.

11-22	E12.5	TARGET CONTOUR ELEVATION - If the exponential contouring options (3 or 4) are selected, then this elevation is used as the base for generating the contour elevations.
23-34	E12.5	ELEVATION ADJUSTMENT FACTOR - This factor is used to adjust the input elevation matrix values if necessary. (See p. 19.)

Option Indicator for Contours

Five different options are available to the user for selecting the contour intervals. If the option indicator is specified as 0 or 6-9, then Option 5 is assumed and the maximum number of contour intervals generated will be 100.

Option 5 - Constant Contour Intervals (Automatic) This option uses the contour interval, CI, specified in columns 59-70 of Program Control Card Number 2 to generate up to 100 contour elevations starting with the lowest integral multiple of CI, which is greater than or equal to the minimum value in the input elevation matrix. For example, if CI=20, the minimum value in the Z variable input matrix is -50, and the maximum value is 60, then the following set of contour elevations would be generated, (-40,-20,0,20,40,60) unless the maximum number of contours requested was less than 6 and Option 5 was specified. In this latter case only the maximum number specified would be generated, starting with -40. Note the actual number of contours generated is determined by the data.

Option 1 - User Specified Contour Intervals This option provides the user the option to specify the contour intervals directly on punched cards. The contour elevations are read using the format (6E12.5). There should be

N values specified, where N is the value contained in columns 2-4 of the ACIGP card. N must be in the range 1 to 100. As many as 6 values are read from each card.

Option 2 - Constant Contour Intervals (User Controlled) This option is similar to Option 5, but the contour intervals start at $L_1 \cdot CI$, where L_1 is the lesser of the two contour interval factors on the ACIGP card. If $L_1 \cdot CI$ is less than the minimum value in the Z variable input matrix, then this option will generate contours starting at the lowest integral multiple of CI greater than the minimum value in the input matrix. The largest contour value generated by this option will be $(L_1+N-1) \cdot CI$ or the greatest integral multiple of CI, which is less than the maximum value in the input matrix, whichever is smaller.

Option 3 - Exponential Contour Intervals for Minimums This option provides the user with the option of generating contour intervals whose spacing varies between subsets of these intervals by powers of 10. The TARGET CONTOUR ELEVATION in columns 11-22 of the ACIGP card is used as the base elevation for generating these contour intervals. For example, if columns 11-22 of the ACIGP card contain the value 1, and the CONTOUR INTERVAL FACTORS are -1 and 0, then the following set of contour intervals may be generated: (-10, -9,-8,-7,-6,-5,-4,-3,-2,-1.9,-1.8,-1.7,-1.6,-1.5,-1.4,-1.3, -1.2,-1.1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2,3,4,5,6,7, 8,9,10). If any of these contour elevations are outside of the range of the input matrix, then those contours are deleted.

Furthermore, no more than N contours will be generated and those farthest from the base elevation will be omitted. This option can be useful in locating relative minimums of a function, whose range spans several orders of magnitude. The contour elevations are given by

$$C_{i_k} = \pm(Z_0 + i \cdot 10^k) \text{ for } i=1,2,\ldots 9, \tag{6}$$

and $k=L_1, L_1+1, \ldots L_2$,

where: C_{i_k} is a generated contour interval.

Z_0 is the TARGET CONTOUR ELEVATION.

L_1 is the lesser of the two CONTOUR INTERVAL FACTORS.

L_2 is the larger of the two CONTOUR INTERVAL FACTORS.

Option 4 - Exponential Contour Intervals for Maximums

This option is similar to Option 3, but a slightly different formula is used to generate the contour intervals. This option is useful in locating relative maximums of a function whose range spans several orders of magnitude. The contour elevations are given by

$$C_{i_k} = \pm(Z_0 - i \cdot 10^k) \text{ for } i=1,2,\ldots 9, \tag{7}$$

and $k=L_1, L_1+1, \ldots L_2$.

Additional options may be added in the future as need or use dictates, however Option 1 allows the user to specify any set of contour elevations desired. These contour elevations are generated by the subroutine CONTIN. If desired the user can program additional options by modifying this subroutine. If this is done, the generated contour elevations have to be sorted in increasing order. This can be achieved by inserting a GO TO 8 statement after storing the values in CI.

Technical Considerations for Generating Contours

Due to the method used to generate the contours by this program, it may sometimes be necessary to adjust the Z variable input matrix values to insure that a contour does not pass through the vertex of a triangle in the set S_1 or S_2. This is accomplished by the program by using the ELEVATION ADJUSTMENT FACTOR contained in columns 23-34 of the ACIGP card. If this value is a 0 or blank, then the program will not adjust the data. However, the program will not operate properly if a contour passes through the vertex of one or more of these triangles. This adjustment factor should be chosen sufficiently small so that none of the contours generated will be displaced appreciably. The value of the adjustment factor depends on the values of the input data and the contour elevations chosen. A typical value for this factor is 10^{-10}, but other values can be used to produce the results desired. An error message is printed if a contour passes through the vertex of a triangle. If this happens, the user is advised to try a different adjustment factor. This adjustment of the data is carried out as described below.

The values of H_{ij} for i=1,2,...m and j=1,2,...n are checked to determine if $|H_{ij}-C_k|<E$, where $E=C_kD$ or if $C_kD=0$, $E=D$. If so, then the value of H_{ij} is set equal to $H_{ij}+E$, otherwise the value of H_{ij} is not changed. This procedure is performed for each elevation C_k with the C_k in increasing

order for k=1,2,...N. These possibly new values of H_{ij} are used in place of the original values of the H_{ij} for all subsequent calculations. Furthermore, the definition of g(X,Y), S_1, and S_2 are also modified by using these possibly new values of the H_{ij}.

The contour elevations define a set of N planes, which are parallel to the (X,Y) plane. Since the surface defined by S_1 and S_2 consists of the set of triangular planar sections S_{kij} defined by using these possibly new values of H_{ij}, the desired contours are given by the intersection of these two surfaces and consist of a series of connected line segments, which do not pass through the vertex of any triangle in S_1 or S_2. Therefore, the line segments form a set of lines consisting of possibly (1) disjoint polygons and/or (2) disjoint simple closed parts of polygons, which terminate on the boundary of the rectangular domain of g(X,Y). Since the N planes defined by the contour elevations do not pass through the vertex of any triangle in S_1 or S_2, none of these triangles can lie entirely in one of these N planes nor can a contour terminate at an interior point of the rectangular domain. This is the reason why the data may need to be adjusted. If the adjustment factor is chosen as 0, then the above described conditions may exist. If this happens the program will write an error message and stop execution. (See "DIAGNOSTIC MESSAGES," p. 25.)

Due to the width of the pen used for plotting and the accuracy of the CalComp plotting subroutines and CalComp plotters and to a lesser extent the accuracy of the finite number system used in the computer, these lines may appear to intersect when the plottape is plotted. Greater seperation of the contour lines can be achieved in general by increasing the adjustment factor D. However, the problem of contour lines appearing to overlap can also be due to the contour intervals being chosen so small that they come together on the domain. To circumvent this problem and to make the map easier to read, the contours are plotted in four different colors with the color changing from one contour to the next higher contour. This feature is not available on the CalComp drum plotter. To further aid in the interpretation of these contours, a contour color key is drawn on the plot with the values of the contours plotted in the same color as the corresponding contours.

Finally, the user has the option of specifying that every r^{th} point on every s^{th} contour be plotted as a special symbol such as a "+" and the elevation for that contour can be plotted by the symbol. The value of r is given by the LINE TYPE variable in columns 78-80 of Program Control Card Number 1. The value of s is given by the CONTOUR LABELING FREQUENCY variable in columns 56-58 of Program Control Card Number 2. The values of r and s should be positive and for good results should not be chosen too small. For example, if r is chosen too small the program may generate so many contour labels and special symbols as to obliterate the contours. Also the user must choose the values in columns 1 and 2 of Program Control Card Number 2 to specify the format for the contour line labels desired. If the user chooses too small a value in column 1 or 2, then incorrect label values will be produced; however, unwanted leading digits can be deleted.

Limitations

If the number of points on a single contour is greater than 999, then the program stops execution and may write a STOP 999 in the HASP system log. If this should occur, the user is advised to divide his domain in half and run the job for each half separately. The maps produced can then be taped together to form a composite map for the entire domain. If this is necessary, be sure to divide the domain so that it reduces the number of points in the contour, which contains too many points, and be sure to include the last row or column in the first half as the first row or column in the second half.

Maximum size of a contour line label is limited by the integers in columns 1 and 2 of Program Control Card Number 2. If the contour elevations are too large, the program may not print the correct value on the label. The user should check the output for error messages to this effect. Also if the contour elevations are too large, the program may generate invalid numbers and terminate with an overflow condition.

TIMING AND STORAGE REQUIREMENTS

Program Size

This program is designed to process a matrix of input data up to the size 101 columns by 51 rows. Since the user may wish to process larger matrices with this program, several changes must be made by the user. These changes are described below.

Change the dimension variables, MD and LD in the main program, to specify the maximum number of rows and columns in the input data matrix desired. MD and LD must both be less than 1000.

Change the dimension in the main program and in each subroutine of the arrays H, PHI1, H2, PHI2, AM, and AN, unless they are already dimensioned using the dimension variables MD and LD. The arrays H, AM, and AN are two-dimensional arrays. For example, the array H should be dimensioned as H(150,100) if LD is set to 150 and MD to 100. The arrays PHI1, H2, and PHI2 are one dimensional arrays and should be given the dimensional value of the maximum of (LD and MD).

The program required 242K bytes of core storage to execute for the set of data used to produce the graph presented herein. The program was compiled by the FORTRAN H compiler with optimization = 0 under release 20 on the U.S.G.S. IBM 360/65 with HASP. If the variables MD and LD are increased and the corresponding changes described above are made, then the size of the program will of course change. This change in size can be estimated by

$$\text{Change in Size} = \frac{(14(MD \cdot LD-101 \cdot 51)+8(\max(LD,MD)-101)+1023)}{1024} K. \quad (8)$$

Program Timing

The program required approximately 2.9 minutes of execution time to execute on the U.S.G.S. IBM 360/65 using a set of data to produce the graph and gradient arrow map contained in attachment A. The program is operational on the IBM 360/65 and the plot tapes have been plotted on both the CalComp drum plotter and the CalComp flatbed plotter. When using the drum plotter, a dummy subroutine must be included called NEWPEN since this subroutine is not contained in the CalComp Software Package for the drum plotter. Also the flatbed plotter should be used if the user desires to produce maps in color.

The plotting times vary significantly depending on the options chosen on the program control cards and the nature of the input data. Approximately 30 minutes of plot time were used in preparing the arrow map and the graph contained in attachment A.

DIAGNOSTIC MESSAGES

In order to aid the user in making sure that he has provided the correct information to the program and to indicate if certain error conditions have been encountered, this program may print the following diagnostic messages.

ALL GRADIENT VALUES ARE ZERO, CHECK INPUT DATA TO SEE IF THE INPUT DATA MATRIX WAS READ IN CORRECTLY.

Cause: All of the gradient magnitudes generated by the program are zero.

Action: Execution terminates and a STOP 2 may be printed in the HASP system log.

NUMBER OF HORIZONTAL GRID LINES MUST BE IN THE RANGE 3 TO nnn.

Cause: The number of horizontal grid lines must be in the range 3 to MD, where MD is the maximum allowable number of horizontal grid lines.

Action: Processing terminates and a STOP 1 may be written in the HASP system log.

NUMBER OF VERTICAL GRID LINES MUST BE IN THE RANGE 3 TO nnn.

Cause: The number of vertical grid lines must be in the range 3 to LD, where LD is the maximum allowable number of vertical grid lines.

Action: Processing terminates and a STOP 1 may be written in the HASP system log.

CONTOUR DOES NOT CLOSE OR END ON BOUNDARY OF DOMAIN.

Cause: The contour does not terminate on the boundary of the domain or else the distance between the last point and the first point in the contour is not less than 10^{-13}.

Action: Processing continues, but results are not usually good. Check output for additional error messages and examine plot carefully.

TOO MANY LINE SEGMENTS IN A CONTOUR.

Cause: A single contour contained 999 or more line segments.

Action: Execution terminates and a STOP 999 may be written in the HASP system log. See "Limitations" p. 22.

THERE ARE 1000 OR MORE LINE DATA CARDS IN A SINGLE SET OF LINE DATA CARDS.

Cause: 1000 or more line data cards were read before an end-of-line card was read.

Action: The program searches for the next end-of-line data card and if one is found, plots the first 999 line data points. Processing continues.

NO END-OF-LINE DATA CARD FOUND FOR LAST SET OF LINE DATA CARDS.

Cause: End-of-line data card is missing.

Action: Make sure each set of line data cards is followed by an end-of-line data card. Execution continues but the last set of line data cards is not plotted.

INSUFFICIENT INPUT DATA FOUND

Cause: The program encountered an end of file on the data set with the DDNAME FT10F001, while trying to read the Z variable input matrix. Make sure variable format statement and data set contain the correct information.

Action: Execution terminates and a STOP 3 may be printed in the HASP system log.

NUMBER OF CONTOUR LEVELS REQUESTED IS INVALID.

Cause: The number of contour levels requested was either less than one or greater than 100.

Action: Processing terminates and a STOP 10 may be printed in the HASP system log.

CONTOUR PASSES THROUGH 3 POINTS ON BOUNDARY OF A TRIANGLE.

Cause: Probably due to the elevation adjustment factor being equal to zero or too close to zero. Increase the value of the factor and run the job again.

Action: Execution terminates and a STOP 4 may be written in the HASP system log.

BREAK IN CONTOUR DETECTED

X1 Y1 X2 Y2 DIST = d

Cause: Either the horizontal or vertical distance between the endpoint of the previous line segment and the starting point of the next segment is greater than 10^{-13}. The distance between these two points is given by d. This message will be printed out for only the first 10 such breaks. May be caused by a contour passing through the vertex of a triangle or poor input data.

Action: Processing continues but results probably will not be good.

CONTOUR DOES NOT CONNECT EXACTLY

X1 X2 Y1 Y2

Cause: The endpoint of the previous line segment (X1,Y1) is not exactly equal to the starting point (X2,Y2) of the next line segment. The coordinates of these two points are printed in hexadecimal. This message will be printed for only the first 20 such occurrences. If this message is printed then $|X1-X2| < 10^{-13}$ and $|Y1-Y2| < 10^{-13}$.

Action: Processing continues, results are usually good but the user should carefully check the plot and the printed output for additional error messages.

CONTOUR PASSES THROUGH THE VERTEX OF A TRIANGLE.

Cause: Probably due to the elevation adjustment factor not being large enough. See "Technical Considerations for Generating Contours" p. 19.

Action: Processing terminates and a STOP 4 may be written in the HASP system log.

THE FOLLOWING NUMBER CANNOT BE FORMATTED FOR PLOTTING AS REQUESTED

0.xxxxxxxxxD$\pm$nn NDEC=nn ND=nn

Cause: The number requested to be formatted for plotting could not be transformed as requested by the MDIG function.

NDEC is the number of requested decimal places to the right of the decimal point.

ND is the maximum number of requested digits to the left of the decimal point.

Action: Processing continues but the number which could not be formatted correctly may be plotted incorrectly. Usually caused by inappropriate values on Program Control Card Number 2 in columns 1-5 or numbers which are too large for conversion.

RAY DOES NOT INTERSECT GRAPH OR POINT BEING PLOTTED IS NOT ON GRAPH.

X1 Y1 X2 Y2 Z1

Cause: The line joining two successive points (X1,Y1), (X2,Y2) requested for plotting does not intersect the graph. The Z value for the point (X_1,Y_1) is also printed out. If $(X_1,Y_1)=(X_2,Y_2)$, then this point is not on the graph. The coordinates X1, Y1, X2, and Y2 are printed in inches on the plotter relative to the lower left corner of the rectangle.

Action: The PTPLOT subroutine continues to process the remaining points requested for plotting and execution continues as normal. Make sure points are within the graph boundaries.

MAGNITUDE OF GRADIENT=$\pm$0.xxxxxE$\pm$NN
FOR THE (i,j) ELEMENT IN GRADIENT MATRIX k.
MAGNITUDE ON MAP SET TO 9.999,
where k = 1 or 2.

Cause: The magnitude of the gradient for the (i,j) element in gradient matrix k is greater than 9.999.

Action: Processing continues but the magnitude on the map is set to 9.999. This can be corrected by using a small Gradient Map Scaling Factor.

In addition, IBM diagnostic messages may be generated if the input data does not conform to the corresponding formats, or if data values generate such conditions as underflows, overflows, and so forth. (See IBM, 1971a, 1971b, and 1971c.) The appropriate selection of the parameter values on the two program control cards is the responsibility of the user. Inappropriate selection of these values can cause the plotter to lose origin and overprint values on the maps produced by this program.

PLOTTING OPTIONS

Several variables are used to determine the manner in which the points selected for plotting will be plotted. The points selected for plotting are those sets of points contained on the special line data cards and the endpoints of the line segments which make up the contours.

The Z INCR variable (t), contained in columns 76-77 of Program Control Card Number 1, limits the program to select only every t^{th} point out of the initial set of points, $\{P_i: i=1, 2, ...\}$, selected for plotting. Thus, only the points P_1, P_{t+1}, P_{2t+1} ... will actually be used for plotting. If the Z HT variable in columns 72-75 of Program Control Card Number 1 is less than .01, then the Z values and special symbols will not be plotted at the points selected for plotting on the contours. If the Z HT variable on an end-of-line card for a special line data set is less than .01, then the Z values will not be plotted above any of the points in that special line data set. Similarly, if the symbol height variable in columns 46-49 on an end-of-line card is less than .01 for a special line data set, then the special symbol will not be plotted at any of the points in that data set.

The CONTOUR LABELING FREQUENCY variable determines which contours will be labeled. For example, if this variable is equal to 5, then only every fifth contour will be selected for labeling as dictated by the LINE TYPE variable. Also, if this variable is less than or equal to 0, then it is set equal to 1 by the program. If a contour line is not selected for labeling, then it is plotted using the value of 999 for the LINE TYPE variable.

The LINE TYPE variable is used to determine how the points selected for plotting will be plotted. The value of the LINE TYPE variable in columns 78-80 of Program Control Card Number 1 is used for each contour line selected for labeling. The value of the LINE TYPE variable on an end-of-line data card is used for that corresponding special line data set. If the LINE TYPE variable is equal to 999 then no symbols or labels will be plotted on the contour or the corresponding special line data sets.

If the LINE TYPE variable is equal to 0, then the points will be connected by line segments, and those parts of the line segments contained in the rectangular domain will be plotted.

If the LINE TYPE variable is less than 0, say -3, then each of the points contained within the rectangular domain will be plotted as a single point; and a special symbol and label will be plotted at every third point, provided that the height variables for these labels and special symbols are not less than .01. The height of the special symbols for contours is the same as the Z HT variable referred to above.

If the LINE TYPE variable is greater than 0, say 4, then successive points selected for plotting are joined by line segments, and those parts of the line segments contained in the rectangular domain will be plotted. Also at every fourth point a label and special symbol will be plotted, provided that the height variables for these labels and special symbols are greater than or equal to .01.

The symbol which is plotted on the contours is determined by the SPEC CHAR CODE, (see CalComp, 1970) contained in columns 6-8 of Program Control Card Number 2. The symbol which is plotted for the special line data sets is determined by the code contained in columns 43-45 of the corresponding end-of-line cards.

The format of the numbers printed on the labels above these symbols is determined by the values contained in columns 1-2 of Program Control Card Number 2 for the contours and by the values contained in columns 41-42 of the end-of-line cards for the corresponding special line data sets.

A special form of the FORTRAN F format is used to convert these numbers for plotting. Define $\delta(Z)=(0 \text{ if } Z\geq 0 \text{ or } 1 \text{ if } Z<0)$. The number Z to be plotted, is converted using the FORTRAN format code Fk.j where: $j<k\leq ND+NDEC+1+\delta(Z)$; j=NDEC; ND is the maximum number of digits to the left of the decimal point; NDEC is the number of decimal places to the right of the decimal point. The actual value of k is determined by the value of Z and ND, and is chosen so that leading zeros or blanks will be deleted. Only the first ND decimal digits to the left of the decimal point are retained in the number Z.

The color of the contours and special line data sets can be varied by using different colored pens in the CalComp flatbed plotter, and to a limited extent, on the CalComp drum plotter.

The pen selected for the contours changes successively from one contour to the next, starting with pen number 1 when the flatbed subroutines are used.

The pen selected for the special line data sets is determined by the number contained in column 37 of the corresponding end-of-line card. If this number is not equal to 1-4, then pen 1 is selected.

If the drum plotter is used, then all of the contours will be plotted using the same pen. However, since each frame of the tape can be plotted individually, each frame can be plotted using a different colored pen if so desired. Each special line data set is plotted on a separate frame and therefore can be plotted in different colors.

Since all of these variables are required for each special line data set, the user has the option to plot each special line data set and the contour data independently of one another in accordance with these options.

If large line data sets are to be plotted with a LINE TYPE variable less than zero, a significant amount of plotting time can be saved by sorting the data points in such a manner as to minimize the sum of the distances between successive points in the line data set.

REFERENCES

CalComp, 1970, Programming CalComp pen plotters: Anaheim, Calif.

IBM, 1971a, Systems reference library, IBM system/360 operating system, messages and codes: File No. 360-20, Order No. GC28-6631-11.

-------1971b, IBM system/360 and system/370, FORTRAN IV language: GC28-6515-8.

-------1971c, IBM systems reference library, IBM system/360 operating system, FORTRAN IV (G and H) programmer's guide: File No. S360-25, Order No. GC28-6817-2.

SYSNØTES, Oct. 19, 1972, v. 5, no. 14, Computer Center Division, U.S.G.S. (unpublished).

ATTACHMENTS

A. PLOTS PRODUCED ON CALCOMP PLOTTER

This program can be used to prepare graphs of functions and certain kinds of maps on the CalComp plotters. The program can also prepare a gradient arrow map, which illustrates these gradients by arrows in the opposite direction of the gradient on the triangles in T_1, T_2, or both. The magnitude of the gradient can also be printed by the arrow if desired. The arrow sizes will vary in relation to their associated magnitude.

The program also allows for the maps to be plotted in different colors either by frame or by automatic pen selection on the CalComp flatbed plotter. The plottape prepared by this program is designed as follows.

Table 5.--Plottape Design Table

Frame Number	Item
1- 2	Initial positioning and selection of pen.
3	Graph frame, graph labels, and graph lines.
4	Graph titles.
5	Gradient Matrix 1 (arrows and magnitudes).
6	Gradient Matrix 2 (arrows and magnitudes).
7	Automatically generated contour lines.
8	Special line data set 1.
9	Special line data set 2.
⋮	⋮
n+7	Special line data set n.
999	Positioning of pen for next plot and end of plottape.

The special line data set frames are written on the plottape only if the line data sets are processed by the program. The first seven frames are always written by the program if it does not terminate execution due to an error condition. The frames 5, 6, and 7 will contain plotting commands only if the corresponding options were selected on the program control cards. Various graphs and maps can be plotted by selecting and plotting those frames desired.

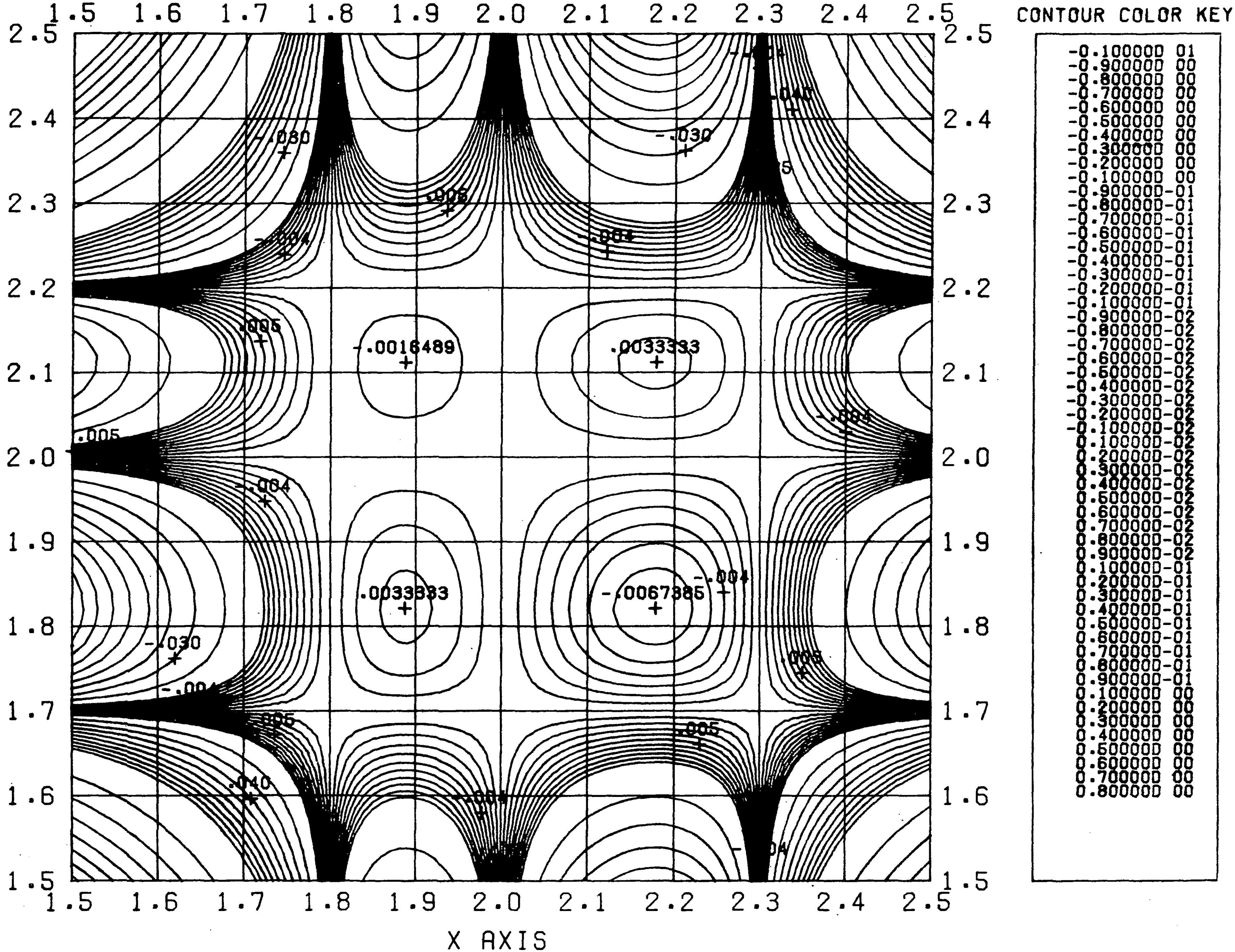
This attachment contains a graph which approximates the function $f(X,Y)=100\cdot(X-1.8)(X-2)(X-2.3)(Y-1.7)(Y-2)(Y-2.2)$ on the rectangular domain $1.5\le X\le 2.5$ and $1.5\le Y\le 2.5$. This function has 9 saddle points and 4 relative maximums or minimums. The lines X=1.8, X=2, X=2.3, Y=1.7, Y=2, and Y=2.2 are all 0's of the function f(X,Y) and divide the domain into 16 sections. The sign of the function changes in adjacent sections. The two relative minimums and two relative maximums plotted in the center sections were generated using special line data sets. The contours shown on this graph were generated using the exponential contour intervals for minimums, see "Option Indicator for Contours," p. 17. The contours are labeled on every eighth contour, which contains 50 or more line segments. The values of the contours generated are listed in the "CONTOUR COLOR KEY" table. Alternate contours are designed to be plotted in different colors to make the graph easier to read. The colors of the numbers in the "CONTOUR COLOR KEY" table are the same as the colors of the corresponding contours on the map. For the purposes of publishing this paper, this particular graph was produced using the same color pen throughout.

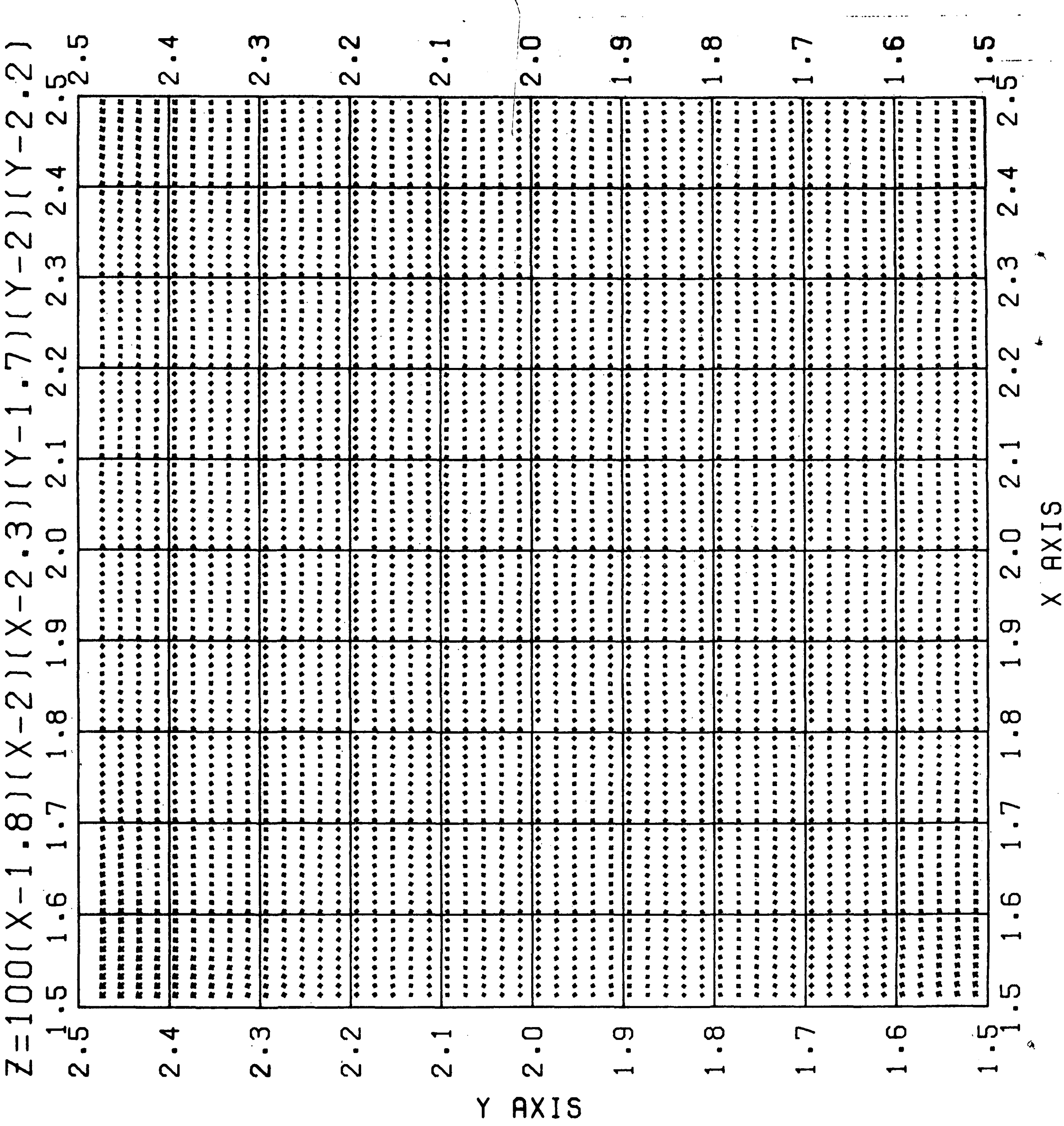
This graph was produced using 101 vertical grid lines at a spacing of .01 units in the X direction and 51 horizontal grid lines at a spacing of .02 units in the Y direction. The input data matrix for each of these grid points was generated using the function f(X,Y) in double precision. All computations for the generation of contours were performed in double precision; however, the plotting routines are coded in single precision. The computed approximation surface should be accurate to at least three places at each of the grid points. Between grid points, linear interpolation is used to estimate the function f(X,Y). Analysis of the differences between f(X,Y) and g(X,Y) as defined in "TECHNICAL DESCRIPTION," p. 10, is a lengthy and complicated procedure. However, it can be shown that the function g(X,Y) converges to f(X,Y) as the grid distances approach 0. The accuracy of this graph varies with position on the graph and is a function of the value of the function f(X,Y) on each of the triangles used to generate g(X,Y). A more accurate graph could be generated by choosing the grid sizes smaller. The grid sizes used in preparing this graph were chosen to keep the core storage requirements of the program to less than 252k bytes.

The arrow map contained in this attachment shows the opposite direction and relative magnitude of the gradients on each of the triangles contained in T_1, only, except for the first row and column of triangles. These are omitted due to space limitations. Arrow maps for the gradients on the triangles contained in T_2 , only, or on both T_1 and T_2 can be plotted by selecting the appropriate frames on the plottape. The arrows in the last column and last row of the triangles in T_2 are also omitted due to space limitations.

The magnitudes of the gradients are multiplied by the GRADIENT MAP SCALING FACTOR contained in columns 39-50 of Program Control Card Number 2. If this number is greater than 9.999, then an error message is printed and the number is set equal to 9.999 on the map. The size of these numbers on the map is determined by the NUMBER SIZE FACTOR contained in columns 51-54 of Program Control Card Number 2. If this factor is equal to 0, then the numbers are not printed on the map. Otherwise they are printed over the arrow representing the gradient unless: (1) the magnitude of the gradient is equal to 0, in which case the arrow and its magnitude are not plotted; or (2) the computed heights of the arrow or of its magnitude is less than .01, in which case the arrow or its magnitude are not plotted, respectively.

Z=100(X-1.8)(X-2)(X-2.3)(Y-1.7)(Y-2)(Y-2.2)





In order to improve the readability of the arrow maps, they should be plotted with larger horizontal and vertical dimensions to increase the size of the arrows. This map was plotted at this size so that it would fit in this paper. The use of a magnifying glass will facilitate the reading of this map.

B. LISTING OF SAMPLE INPUT CARDS

This attachment shows the two particular program control cards, the three map title cards, the variable format card for reading the input matrix, the automatic contour interval generation parameters card, and the four special line data set cards and associated end-of-line data cards used to generate the graph contained in attachment A and printer output contained in attachment C. These cards are placed in the job deck, as shown in attachment E, immediately following the //GO.SYSIN DD * card in the same order as shown below. A partial printout of the Z variable input matrix is shown in attachment C. The Z variable input matrix was read in from the data set with the DDNAME of FT10F001, which was stored on disc.

```
 1.5          2.5          1.5          2.5          10.0  10.0 1111 0.6503.135 1 50
30122  3 51101 .01          .02          0.1          0   3  8  .01
43Z=100(X-1.8)(X-2)(X-2.3)(Y-1.7)(Y-2)(Y-2.2)
 6X AXIS
 6Y AXIS
(1X,5E22.15)
3100 -3  4   0.0       0.0
 0.33333D-02 0.21786D 01 0.21120D 01
                                    1   170   3.135.135 1
-0.67385D-02 0.21786D 01 0.18214D 01
                                    1   170   3.135.135 1
-0.16489D-02 0.18880D 01 0.21120D 01
                                    1   170   3.135.135 1
 0.33333D-02 0.18880D 01 0.18214D 01
                                    1   170   3.135.135 1
/*
```

C. PRINTER OUTPUT

This attachment contains some of the printer output produced by this program for the particular set of input data contained in this document. Only the first page of printout of the Input Data Matrix, Gradient Matrix 1, and Gradient Angle Matrix 1 have been included in this attachment to reduce the number of pages in this paper.

GENERALIZEC GRADIENT AND CONTOUR PROGRAM INPUT PARAMETERS

```
    XMIN          XMAX          YMIN          YMAX          WIDTH    HEIGHT    NO OF TICK MARKS   FACTOR
 0.15000E 01   0.25000E 01   0.15000E 01   0.25000E 01     10.000   10.000    11 HOR  11 VERT     0.650

GR LINE IND   Z HT   Z INCR   LINE TYPE   Z FORMAT XY FORMAT XY FORMAT 2
     3        .135     1         50         F4.3     F4.1      F4.1

SPEC CHAR CODE  NO OF GRID LINES
      3           51 HOR  101 VERT

          GRID DISTANCES                      GRADIENT MAP SCALING FACTOR   NUMBER SIZE FACTOR
 0.10C00E-01 HOR    0.20000E-01 VERT                 C.10CC0E 00                  0.0

   OPTIONS INDICATOR  CONTOUR LABELING FREQUENCY  CONTOUR INTERVAL

           3                      8                  C.10000D-01
```

INPUT DATA MATRIX

-0.14400D 01	-0.13471D 01	-0.12590D 01	-0.11726D 01	-0.10908D 01	-0.10125D 01	-0.93773D 00	-0.86636D 00	-0.79834D 00	-0.73357D 00
-0.67200D 00	-0.61355D 00	-0.55814D 00	-0.50572D 00	-0.45619D 00	-0.40950D 00	-0.36557D 00	-0.32432D 00	-0.28570D 00	-0.24961D 00
-0.21600D 00	-0.18479D 00	-0.15590D 00	-0.12928D 00	-0.10483D 00	-0.82500D-01	-0.62208D-01	-0.43884D-01	-0.27456D-01	-0.12852D-01
0.0	0.11172D-01	0.20736D-01	0.28764D-01	0.35328D-01	0.40500D-01	0.44352D-01	0.46956D-01	0.48384D-01	0.48708D-01
0.48000D-01	0.46332D-01	0.43776D-01	0.40404D-01	0.36288D-01	0.31500D-01	0.26112D-01	0.20196D-01	0.13824D-01	0.70680D-02
0.0	-0.73080D-02	-0.14784D-01	-0.22356D-01	-0.29952D-01	-0.37500D-01	-0.44928D-01	-0.52164D-01	-0.59136D-01	-0.65772D-01
-0.72000D-01	-0.77748D-01	-0.82944D-01	-0.87516D-01	-0.91392D-01	-0.94500D-01	-0.96768D-01	-0.98124D-01	-0.98496D-01	-0.97812D-01
-0.96000D-01	-0.92988D-01	-0.88704D-01	-0.83076D-01	-0.76032D-01	-0.67500D-01	-0.57408D-01	-0.45684D-01	-0.32256D-01	-0.17052D-01
0.0	0.18972D-01	0.39936D-01	0.62964D-01	0.88128D-01	0.11550D 00	0.14515D 00	0.17716D 00	0.21158D 00	0.24851D 00
0.28800D 00	0.33013D 00	0.37498D 00	0.42260D 00	0.47309D 00	0.52650D 00	0.58291D 00	0.64240D 00	0.70502D 00	0.77087D 00
0.84000D 00									

-0.12580D 01	-0.11768D 01	-0.10990D 01	-0.10243D 01	-0.95288D 00	-0.88452D 00	-0.81920D 00	-0.75686D 00	-0.69743D 00	-0.64085D 00
-0.58706D 00	-0.53600D 00	-0.48759D 00	-0.44179D 00	-0.39853D 00	-0.35774D 00	-0.31936D 00	-0.28333D 00	-0.24958D 00	-0.21806D 00
-0.18870D 00	-0.16143D 00	-0.13620D 00	-0.11294D 00	-0.91581D-01	-0.72072D-01	-0.54345D-01	-0.38337D-01	-0.23986D-01	-0.11228D-01
0.0	0.97599D-02	0.18115D-01	0.25129D-01	0.30863D-01	0.35381D-01	0.38746D-01	0.41021D-01	0.42268D-01	0.42551D-01
0.41933D-01	0.40476D-01	0.38243D-01	0.35297D-01	0.31701D-01	0.27518D-01	0.22811D-01	0.17643D-01	0.12077D-01	0.61746D-02
0.0	-0.63843D-02	-0.12915D-01	-0.19530D-01	-0.26166D-01	-0.32760D-01	-0.39249D-01	-0.45570D-01	-0.51661D-01	-0.57458D-01
-0.62899D-01	-0.67921D-01	-0.72460D-01	-0.76454D-01	-0.79840D-01	-0.82555D-01	-0.84537D-01	-0.85721D-01	-0.86046D-01	-0.85449D-01
-0.83866D-01	-0.81234D-01	-0.77492D-01	-0.72575D-01	-0.66422D-01	-0.58968D-01	-0.50152D-01	-0.39910D-01	-0.28179D-01	-0.14897D-01
0.0	0.16574D-01	0.34888D-01	0.55005D-01	0.76989D-01	0.10090D 00	0.12680D 00	0.15476D 00	0.18484D 00	0.21710D 00
0.25160D 00	0.28840D 00	0.32758D 00	0.36919D 00	0.41329D 00	0.45995D 00	0.50923D 00	0.56120D 00	0.61591D 00	0.67343D 00
0.73392D 00									

-0.10908D 01	-0.10204D 01	-0.95288D 00	-0.88817D 00	-0.82621D 00	-0.76694D 00	-0.71030D 00	-0.65624D 00	-0.60471D 00	-0.55566D 00
-0.50902D 00	-0.46474D 00	-0.42278D 00	-0.38306D 00	-0.34555D 00	-0.31018D 00	-0.27691D 00	-0.24566D 00	-0.21641D 00	-0.18907D 00
-0.16361D 00	-0.13997D 00	-0.11809D 00	-0.97922D-01	-0.79407D-01	-0.62491D-01	-0.47120D-01	-0.33241D-01	-0.20797D-01	-0.97350D-02
0.0	0.84624D-02	0.15707D-01	0.21788D-01	0.26760D-01	0.30677D-01	0.33595D-01	0.35568D-01	0.36649D-01	0.36895D-01
0.36358D-01	0.35095D-01	0.33159D-01	0.30605D-01	0.27487D-01	0.23860D-01	0.19779D-01	0.15298D-01	0.10471D-01	0.53538D-02
0.0	-0.55356D-02	-0.11198D-01	-0.16934D-01	-0.22688D-01	-0.28405D-01	-0.34031D-01	-0.39512D-01	-0.44794D-01	-0.49820D-01
-0.54538D-01	-0.58892D-01	-0.62827D-01	-0.66290D-01	-0.69226D-01	-0.71581D-01	-0.73299D-01	-0.74326D-01	-0.74607D-01	-0.74089D-01
-0.72717D-01	-0.70435D-01	-0.67190D-01	-0.62927D-01	-0.57592D-01	-0.51129D-01	-0.43485D-01	-0.34604D-01	-0.24433D-01	-0.12916D-01
0.0	0.14371D-01	0.30250D-01	0.47693D-01	0.66754D-01	0.87487D-01	0.10995D 00	0.13419D 00	0.16027D 00	0.18824D 00
0.21815D 00	0.25006D 00	0.28403D 00	0.32011D 00	0.35835D 00	0.39881D 00	0.44154D 00	0.48659D 00	0.53403D 00	0.58391D 00
0.63627D 00									

-0.93773D 00	-0.87724D 00	-0.81920D 00	-0.76357D 00	-0.71030D 00	-0.65934D 00	-0.61065D 00	-0.56418D 00	-0.51988D 00	-0.47770D 00
-0.43761D 00	-0.39954D 00	-0.36346D 00	-0.32932D 00	-0.29707D 00	-0.26667D 00	-0.23806D 00	-0.21120D 00	-0.18605D 00	-0.16255D 00
-0.14066D 00	-0.12033D 00	-0.10152D 00	-0.84185D-01	-0.68267D-01	-0.53724D-01	-0.40510D-01	-0.28577D-01	-0.17879D-01	-0.83692D-02
0.0	0.72752D-02	0.13503D-01	0.18731D-01	0.23006D-01	0.26374D-01	0.28882D-01	0.30578D-01	0.31508D-01	0.31719D-01
0.31258D-01	0.30171D-01	0.28507D-01	0.26311D-01	0.23631D-01	0.20513D-01	0.17004D-01	0.13152D-01	0.90022D-02	0.46027D-02
0.0	-0.47590D-02	-0.96273D-02	-0.14558D-01	-0.19505D-01	-0.24420D-01	-0.29257D-01	-0.33969D-01	-0.38509D-01	-0.42831D-01
-0.46886D-01	-0.50629D-01	-0.54013D-01	-0.56990D-01	-0.59514D-01	-0.61538D-01	-0.63015D-01	-0.63898D-01	-0.64141D-01	-0.63695D-01
-0.62515D-01	-0.60554D-01	-0.57764D-01	-0.54099D-01	-0.49512D-01	-0.43956D-01	-0.37384D-01	-0.29749D-01	-0.21005D-01	-0.11104D-01
0.0	0.12355D-01	0.26006D-01	0.41002D-01	0.57389D-01	0.75214D-01	0.94523D-01	0.11536D 00	0.13778D 00	0.16183D 00
0.18755D 00	0.21498D 00	0.24418D 00	0.27520D 00	0.30807D 00	0.34286D 00	0.37959D 00	0.41833D 00	0.45911D 00	0.50199D 00
0.54701D 00									

-0.79834D 00	-0.74684D 00	-0.69743D 00	-0.65007D 00	-0.60471D 00	-0.56133D 00	-0.51988D 00	-0.48031D 00	-0.44260D 00	-0.40669D 00
-0.37256D 00	-0.34015D 00	-0.30944D 00	-0.28037D 00	-0.25291D 00	-0.22703D 00	-0.20267D 00	-0.17981D 00	-0.15839D 00	-0.13838D 00
-0.11975D 00	-0.10245D 00	-0.86433D-01	-0.71671D-01	-0.58119D-01	-0.45738D-01	-0.34488D-01	-0.24329D-01	-0.15222D-01	-0.71251D-02
0.0	0.61938D-02	0.11496D-01	0.15947D-01	0.19586D-01	0.22453D-01	0.24589D-01	0.26032D-01	0.26824D-01	0.27004D-01
0.26611D-01	0.25686D-01	0.24269D-01	0.22400D-01	0.20118D-01	0.17464D-01	0.14476D-01	0.11197D-01	0.76640D-02	0.39185D-02
0.0	-0.40516D-02	-0.81962D-02	-0.12394D-01	-0.16605D-01	-0.20790D-01	-0.24908D-01	-0.28920D-01	-0.32785D-01	-0.36464D-01
-0.39917D-01	-0.43103D-01	-0.45984D-01	-0.48519D-01	-0.50668D-01	-0.52391D-01	-0.53648D-01	-0.54400D-01	-0.54606D-01	-0.54227D-01
-0.53222D-01	-0.51553D-01	-0.49177D-01	-0.46057D-01	-0.42152D-01	-0.37422D-01	-0.31827D-01	-0.25327D-01	-0.17883D-01	-0.94536D-02
0.0	0.10518D-01	0.22141D-01	0.34907D-01	0.48858D-01	0.64033D-01	0.80472D-01	0.98215D-01	0.11730D 00	0.13777D 00
0.15967D 00	0.18303D 00	0.20789D 00	0.23429D 00	0.26228D 00	0.29189D 00	0.32317D 00	0.35614D 00	0.39087D 00	0.42737D 00

GRADIENT MATRIX 1

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.11037D 02	0.10443D 02	0.98706D 01	0.93192D 01	0.87883D 01	0.82774D 01	0.77863D 01	0.73144D 01	0.68613D 01
0.64266D 01	0.60099D 01	0.56108D 01	0.52289D 01	0.48637D 01	0.45148D 01	0.41819D 01	0.38645D 01	0.35622D 01	0.32746D 01
0.30013D 01	0.27420D 01	0.24962D 01	0.22635D 01	0.20437D 01	0.18363D 01	0.16410D 01	0.14576D 01	0.12857D 01	0.11252D 01
0.97599D 00	0.83803D 00	0.71159D 00	0.59726D 00	0.49621D 00	0.41054D 00	0.34362D 00	0.29984D 00	0.28237D 00	0.28951D 00
0.31451D 00	0.34963D 00	0.38909D 00	0.42934D 00	0.46836D 00	0.50499D 00	0.53860D 00	0.56898D 00	0.59564D 00	0.61882D 00
0.63843D 00	0.65448D 00	0.66704D 00	0.67616D 00	0.68194D 00	0.68447D 00	0.68385D 00	0.68023D 00	0.67379D 00	0.66474D 00
0.65341D 00	0.64020D 00	0.62569D 00	0.61065D 00	0.59611D 00	0.58340D 00	0.57425D 00	0.57070D 00	0.57505D 00	0.58961D 00
0.61642D 00	0.65697D 00	0.71206D 00	0.78191D 00	0.86630D 00	0.96484D 00	0.10771D 01	0.12027D 01	0.13414D 01	0.14929D 01
0.16574D 01	0.18347D 01	0.20250D 01	0.22285D 01	0.24454D 01	0.26758D 01	0.29202D 01	0.31787D 01	0.34517D 01	0.37396D 01
0.40428D 01	0.43614D 01	0.46961D 01	0.50470D 01	0.54147D 01	0.57994D 01	0.62017D 01	0.66218D 01	0.70602D 01	0.75173D 01
0.0	0.98390D 01	0.93032D 01	0.87869D 01	0.82898D 01	0.78115D 01	0.73515D 01	0.69096D 01	0.64853D 01	0.60781D 01
0.56879D 01	0.53141D 01	0.49564D 01	0.46144D 01	0.42877D 01	0.39759D 01	0.36787D 01	0.33957D 01	0.31265D 01	0.28707D 01
0.26280D 01	0.23981D 01	0.21805D 01	0.19745D 01	0.17809D 01	0.15983D 01	0.14268D 01	0.12660D 01	0.11158D 01	0.97589D 00
0.84624D 00	0.72687D 00	0.61799D 00	0.52016D 00	0.43441D 00	0.36255D 00	0.30731D 00	0.27193D 00	0.25825D 00	0.26430D 00
0.28462D 00	0.31319D 00	0.34546D 00	0.37854D 00	0.41074D 00	0.44111D 00	0.46910D 00	0.49444D 00	0.51699D 00	0.53665D 00
0.55356D 00	0.56761D 00	0.57891D 00	0.58751D 00	0.59347D 00	0.59688D 00	0.59783D 00	0.59642D 00	0.59278D 00	0.58709D 00
0.57958D 00	0.57058D 00	0.56050D 00	0.54993D 00	0.53965D 00	0.53069D 00	0.52432D 00	0.52213D 00	0.52590D 00	0.53753D 00
0.55878D 00	0.59111D 00	0.63551D 00	0.69248D 00	0.76215D 00	0.84439D 00	0.93898D 00	0.10457D 01	0.11643D 01	0.12948D 01
0.14371D 01	0.15911D 01	0.17572D 01	0.19352D 01	0.21256D 01	0.23284D 01	0.25439D 01	0.27725D 01	0.30144D 01	0.32698D 01
0.35393D 01	0.38230D 01	0.41213D 01	0.44345D 01	0.47631D 01	0.51073D 01	0.54676D 01	0.58442D 01	0.62376D 01	0.66481D 01
0.0	0.87289D 01	0.82474D 01	0.77837D 01	0.73374D 01	0.69082D 01	0.64958D 01	0.60998D 01	0.57199D 01	0.53556D 01
0.50067D 01	0.46728D 01	0.43536D 01	0.40487D 01	0.37577D 01	0.34803D 01	0.32162D 01	0.29651D 01	0.27265D 01	0.25001D 01
0.22356D 01	0.20827D 01	0.18911D 01	0.17104D 01	0.15402D 01	0.13804D 01	0.12307D 01	0.10907D 01	0.96025D 00	0.83923D 00
0.72752D 00	0.62515D 00	0.53233D 00	0.44955D 00	0.37772D 00	0.31835D 00	0.27356D 00	0.24556D 00	0.23513D 00	0.24021D 00
0.25646D 00	0.27927D 00	0.30514D 00	0.33179D 00	0.35786D 00	0.38256D 00	0.40545D 00	0.42630D 00	0.44501D 00	0.46154D 00
0.47590D 00	0.48812D 00	0.49825D 00	0.50635D 00	0.51246D 00	0.51664D 00	0.51896D 00	0.51949D 00	0.51833D 00	0.51558D 00
0.51142D 00	0.50606D 00	0.49980D 00	0.49308D 00	0.48644D 00	0.48063D 00	0.47661D 00	0.47554D 00	0.47880D 00	0.48789D 00
0.50434D 00	0.52951D 00	0.56448D 00	0.60999D 00	0.66642D 00	0.73392D 00	0.81246D 00	0.90195D 00	0.10023D 01	0.11135D 01
0.12355D 01	0.13683D 01	0.15120D 01	0.16668D 01	0.18328D 01	0.20102D 01	0.21993D 01	0.24003D 01	0.26135D 01	0.28391D 01
0.30774D 01	0.33288D 01	0.35935D 01	0.38719D 01	0.41642D 01	0.44709D 01	0.47921D 01	0.51283D 01	0.54799D 01	0.58470D 01
0.0	0.77028D 01	0.72721D 01	0.68575D 01	0.64587D 01	0.60754D 01	0.57074D 01	0.53542D 01	0.50156D 01	0.46912D 01
0.43808D 01	0.40839D 01	0.38004D 01	0.35298D 01	0.32719D 01	0.30264D 01	0.27928D 01	0.25710D 01	0.23606D 01	0.21613D 01
0.19728D 01	0.17948D 01	0.16270D 01	0.14690D 01	0.13207D 01	0.11818D 01	0.10519D 01	0.93089D 00	0.81855D 00	0.71474D 00
0.61938D 00	0.53249D 00	0.45427D 00	0.38516D 00	0.32594D 00	0.27779D 00	0.24227D 00	0.22067D 00	0.21300D 00	0.21724D 00
0.22997D 00	0.24777D 00	0.26800D 00	0.28894D 00	0.30952D 00	0.32912D 00	0.34741D 00	0.36420D 00	0.37943D 00	0.39307D 00
0.40516D 00	0.41571D 00	0.42477D 00	0.43239D 00	0.43860D 00	0.44345D 00	0.44696D 00	0.44918D 00	0.45016D 00	0.44996D 00
0.44867D 00	0.44643D 00	0.44343D 00	0.43995D 00	0.43637D 00	0.43319D 00	0.43109D 00	0.43093D 00	0.43373D 00	0.44067D 00
0.45302D 00	0.47201D 00	0.49876D 00	0.53415D 00	0.57878D 00	0.63302D 00	0.69706D 00	0.77095D 00	0.85470D 00	0.94832D 00
0.10518D 01	0.11652D 01	0.12886D 01	0.14222D 01	0.15660D 01	0.17202D 01	0.18851D 01	0.20608D 01	0.22477D 01	0.24458D 01
0.26556D 01	0.28772D 01	0.31109D 01	0.33571D 01	0.36160D 01	0.38878D 01	0.41730D 01	0.44717D 01	0.47844D 01	0.51112D 01
0.0	0.67572D 01	0.63739D 01	0.60051D 01	0.56506D 01	0.53101D 01	0.49834D 01	0.46700D 01	0.43698D 01	0.40825D 01
0.38077D 01	0.35452D 01	0.32947D 01	0.30559D 01	0.28285D 01	0.26123D 01	0.24069D 01	0.22121D 01	0.20276D 01	0.18531D 01
0.16884D 01	0.15332D 01	0.13871D 01	0.12500D 01	0.11215D 01	0.10016D 01	0.88978D 00	0.78602D 00	0.69011D 00	0.60191D 00

GRADIENT ANGLE MATRIX 1

90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000
90.000	134.867	135.614	136.386	137.185	138.011	138.866	139.752	140.671	141.624
142.614	143.644	144.716	145.834	147.000	148.219	149.496	150.835	152.244	153.728
155.298	156.963	158.736	160.632	162.671	164.878	167.284	169.932	172.877	176.197
180.000	-175.560	-170.258	-163.761	-155.581	-145.052	-131.455	-114.586	-95.753	-77.664
-62.399	-50.308	-40.791	-33.123	-26.737	-21.236	-16.350	-11.897	-7.745	-3.803
0.0	3.718	7.394	11.069	14.776	18.550	22.426	26.442	30.639	35.067
39.780	44.844	50.331	56.324	62.904	70.147	78.095	86.736	95.965	105.574
115.269	124.727	133.668	141.906	149.361	156.031	161.971	167.258	171.973	176.197
180.000	-176.558	-173.424	-170.557	-167.921	-165.484	-163.224	-161.118	-159.149	-157.302
-155.565	-153.926	-152.377	-150.908	-149.513	-148.186	-146.922	-145.714	-144.560	-143.455
90.000	133.325	134.072	134.844	135.644	136.473	137.332	138.224	139.151	140.114
141.116	142.160	143.249	144.386	145.575	146.821	148.129	149.504	150.953	152.484
154.106	155.831	157.672	159.646	161.774	164.082	166.605	169.387	172.487	175.987
180.000	-175.315	-169.730	-162.913	-154.399	-143.591	-129.929	-113.439	-95.453	-78.293
-63.646	-51.816	-42.323	-34.550	-27.997	-22.298	-17.203	-12.535	-8.168	-4.012
0.0	3.923	7.799	11.665	15.555	19.501	23.535	27.692	32.009	36.531
41.304	46.386	51.839	57.732	64.135	71.112	78.703	86.906	95.654	104.794
114.098	123.296	132.129	140.399	147.951	154.865	161.043	166.578	171.535	175.987
180.000	-176.367	-173.064	-170.045	-167.273	-164.718	-162.352	-160.153	-158.102	-156.183
-154.383	-152.688	-151.090	-149.579	-148.147	-146.788	-145.496	-144.265	-143.090	-141.968
90.000	131.674	132.417	133.188	133.987	134.816	135.677	136.573	137.505	138.475
139.487	140.543	141.647	142.803	144.014	145.286	146.624	148.034	149.525	151.103
152.781	154.569	156.483	158.541	160.766	163.186	165.839	168.770	172.045	175.749
180.000	-175.038	-169.133	-161.960	-153.084	-141.994	-128.307	-112.253	-95.148	-78.934
-64.942	-53.415	-43.976	-36.114	-29.393	-23.486	-18.162	-13.256	-8.647	-4.251
0.0	4.155	8.257	12.339	16.433	20.567	24.772	29.078	33.518	38.129
42.953	48.039	53.438	59.210	65.413	72.104	79.323	87.080	95.338	103.996
112.886	121.793	130.486	138.763	146.482	153.567	160.002	165.809	171.039	175.749
180.000	-176.152	-172.655	-169.465	-166.542	-163.854	-161.371	-159.071	-156.931	-154.935
-153.067	-151.314	-149.666	-148.111	-146.642	-145.252	-143.933	-142.679	-141.486	-140.349
90.000	129.900	130.638	131.403	132.198	133.025	133.885	134.781	135.715	136.690
137.708	138.774	139.891	141.062	142.294	143.590	144.957	146.402	147.934	149.561
151.296	153.151	155.143	157.292	159.623	162.167	164.964	168.065	171.539	175.476
180.000	-174.719	-168.449	-160.877	-151.610	-140.242	-126.576	-111.024	-94.838	-79.590
-66.290	-55.115	-45.769	-37.837	-30.950	-24.825	-19.251	-14.079	-9.197	-4.524
0.0	4.423	8.782	13.109	17.431	21.775	26.164	30.626	35.188	35.882
44.745	49.814	55.137	60.763	66.742	73.126	79.957	87.257	95.016	103.177
111.630	120.210	128.725	136.979	144.812	152.111	158.821	164.933	170.471	175.476
180.000	-175.904	-172.187	-168.801	-165.708	-162.870	-160.259	-157.846	-155.610	-153.531
-151.592	-149.779	-148.079	-146.481	-144.976	-143.555	-142.211	-140.937	-139.728	-138.578
90.000	127.989	128.716	129.473	130.260	131.080	131.935	132.827	133.760	134.735
135.757	136.828	137.954	139.138	140.386	141.704	143.098	144.577	146.149	147.826
149.619	151.544	153.619	155.866	158.314	160.995	163.955	167.249	170.951	175.158

GRADIENT MATRIX 1 COMPLETED.

HORIZCNTAL GRID COORDINATES

0.0	0.10000000D-01	0.20000000D-01	0.30000000D-01	0.40000000D-01
0.50000000D-01	0.60000000D-01	0.70000000D-01	0.80000000D-01	0.90000000D-01
0.10000000D 00	0.11000000D 00	0.12000000D 00	0.13000000D 00	0.14000000D 00
0.15000000D 00	0.16000000D 00	0.17000000D 00	0.18000000D 00	0.19000000D 00
0.20000000D 00	0.21000000D 00	0.22000000D 00	0.23000000D 00	0.24000000D 00
0.25000000D 00	0.26000000D 00	0.27000000D 00	0.28000000D 00	0.29000000D 00
0.30000000D 00	0.31000000D 00	0.32000000D 00	0.33000000D 00	0.34000000D 00
0.35000000D 00	0.36000000D 00	0.37000000D 00	0.38000000D 00	0.39000000D 00
0.40000000D 00	0.41000000D 00	0.42000000D 00	0.43000000D 00	0.44000000D 00
0.45000000D 00	0.46000000D 00	0.47000000D 00	0.48000000D 00	0.49000000D 00
0.50000000D 00	0.51000000D 00	0.52000000D 00	0.53000000D 00	0.54000000D 00
0.55000000D 00	0.56000000D 00	0.57000000D 00	0.58000000D 00	0.59000000D 00
0.60000000D 00	0.61000000D 00	0.62000000D 00	0.63000000D 00	0.64000000D 00
0.65000000D 00	0.66000000D 00	0.67000000D 00	0.68000000D 00	0.69000000D 00
0.70000000D 00	0.71000000D 00	0.72000000D 00	0.73000000D 00	0.74000000D 00
0.75000000D 00	0.76000000D 00	0.77000000D 00	0.78000000D 00	0.79000000D 00
0.80000000D 00	0.81000000D 00	0.82000000D 00	0.83000000D 00	0.84000000D 00
0.85000000D 00	0.86000000D 00	0.87000000D 00	0.88000000D 00	0.89000000D 00
0.90000000D 00	0.91000000D 00	0.92000000D 00	0.93000000D 00	0.94000000D 00
0.95000000D 00	0.96000000D 00	0.97000000D 00	0.98000000D 00	0.99000000D 00
0.10000000D 01				

VERTICAL GRID COORDINATES

0.10000000D 01	0.98000000D 00	0.96000000D 00	0.94000000D 00	0.92000000D 00
0.90000000D 00	0.88000000D 00	0.86000000D 00	0.84000000D 00	0.82000000D 00
0.80000000D 00	0.78000000D 00	0.76000000D 00	0.74000000D 00	0.72000000D 00
0.70000000D 00	0.68000000D 00	0.66000000D 00	0.64000000D 00	0.62000000D 00
0.60000000D 00	0.58000000D 00	0.56000000D 00	0.54000000D 00	0.52000000D 00
0.50000000D 00	0.48000000D 00	0.46000000D 00	0.44000000D 00	0.42000000D 00
0.40000000D 00	0.38000000D 00	0.36000000D 00	0.34000000D 00	0.32000000D 00
0.30000000D 00	0.28000000D 00	0.26000000D 00	0.24000000D 00	0.22000000D 00
0.20000000D 00	0.18000000D 00	0.16000000D 00	0.14000000D 00	0.12000000D 00
0.10000000D 00	0.80000000D-01	0.60000000D-01	0.40000000D-01	0.20000000D-01
0.0				

AUTOMATIC CONTOUR INTERVAL GENERATION PARAMETERS

OPTION IND	MAX NO OF CONTOURS	CONTOUR INTERVAL FACTORS	
3	100	-3	4

TARGET CONTOUR ELEVATION	ELEVATION ADJUSTMENT FACTOR
0.0	0.0

D. PROGRAM LISTING

This attachment contains a listing of the FORTRAN source code. In addition, the program uses a number of subroutines in SYS1.FORTLIB, SYS1.FLATBEDC, XTENT.LIB, and an object deck of the subroutine CORE, which is not stored in a library. If the program is used to produce plots on the CalComp drum plotter, then an additional dummy subroutine named NEWPEN is required to resolve an external reference. This subroutine should not be included if the job is prepared for the CalComp flatbed plotter. This subroutine can be coded as shown below.

```
SUBROUTINE NEWPEN(INP)
INTEGER*4 INP,N
N=INP
RETURN
END
```

In addition the JCL used to invoke the CalComp drum subroutine package is slightly different. This is discussed in SYSNØTES, (1972).

```
C     GENERALIZED GRADIENT AND CONTOUR PROGRAM E874 BY MARSHALL STRONG
C     HELLMANN JAN 18, 1973
      INTEGER*4           ITITLX(18),ITITLY(18),IHEAD(18)
      LOGICAL*1 ZRO,PL,MIN
      REAL*4 H1(1000),PHI(1000),X(1000),Y(1000),PHI1(101,51)
      REAL*4 IMAGE(4000)
      REAL*8 ZXZ(13),X11,XN1,Y11,YM1
      REAL*8 Z(1000),Z2,H(101,51)
      REAL*8 CI,CIN
      COMMON LD,MD,CI(101),CIN,M100
      COMMON /CON/ ZXZ,H,PL,ZRO,MIN
      COMMON /C1C1/ IC2,ZROSUP
      COMMON /DIST/ SX,SY,ZFAC,HTF,IMII
      COMMON /FACS/ FAC
      COMMON /H1/H1,PHI,XL,YL,INT,LT,NDEC,ND,IC,HT2,LTF,ND4,NDEC2
      COMMON /ZXY/ Z,X,Y
      CALL PLOTS (IMAGE(1),4000,9)
      ZROSUP=0
      MD=51
      LD=101
      M100=1000
      M99=M100-1
      LK=5
      WRITE (6,1)
    1 FORMAT (1H1,37X,'GENERALIZED GRADIENT AND CONTOUR PROGRAM INPUT PA
     1RAMETERS'///)
      READ (5,2,END=30) X11,XN1,Y11,YM1,W,U,IW,IV,FAC,LN,HT2,IC,LT,NDEC,
     1ND,NDEC2,ND2,ND3,INT,M1,N1,SX,SY,ZFAC,HTF,IMII,LTF,CIN
    2 FORMAT (4E12.5,2F6.3,2I2,F6.3,I1,F4.3,I2,I3/5I1,3I3,3E12.5,F4.3,I1
     1,I3,E12.5)
      IF (FAC.EQ.0) FAC=1
      X1=X11
      XN=XN1
      IW = AMAX0(IW,2)
      IV = AMAX0(IV,2)
      Y1=Y11
      YM=YM1
      CALL FACTOR (FAC)
      WRITE (6,3) X1,XN,Y1,YM,W,U,IW,IV,FAC
    3 FORMAT (5X,'XMIN',10X,'XMAX',10X,'YMIN',10X,'YMAX',10X,'WIDTH',4X,
     1'HEIGHT',4X,'NO OF TICK MARKS',' FACTOR'/4(1X,E12.5,1X),4X,F7.3,2
     2X,F7.3,4X,I2,' HOR',2X,I2,' VERT',3X,F7.3/)
      RAD=57.2958
      NFZ=ND+NDEC+1
      NFL=ND2+NDEC2+1
      NFL2=ND3+NDEC2+1
      ND4=MAX0(ND2,ND3)
      WRITE (6,4) LN,HT2,IC,LT,NFZ,NDEC,NFL,NDEC2,NFL2,NDEC2
    4 FORMAT (1H0,'GR LINE IND  Z HT   Z INCR  LINE TYPE  Z FORMAT XY FO
     1RMAT XY FORMAT 2'/7X,I1,6X,F4.3,5X,I3,6X,I3,7X,'F',I1,'.',I1,5X,'F
     2',I1,'.',I1,5X,'F',I1,'.',I1//)
      WRITE (6,5) INT,M1,N1
```

```
 5 FORMAT (/1X,'SPEC CHAR CODE  NO OF GRID LINES'/8X,I2,7X,I3,' HOR',
  12X,I3,' VERT'//)
   WRITE (6,6) SX,SY,ZFAC,HTF
 6 FORMAT (1H0,15X,'GRID DISTANCES',19X,'GRADIENT MAP SCALING FACTOR
  1  NUMBER SIZE FACTOR'/E13.5,' HOR',2X,E13.5,' VERT',15X,E13.5,17X,
  2F5.3///)
   WRITE (6,7) IMII,LTF,CIN
 7 FORMAT (5X,'OPTIONS INDICATOR',2X,'CONTOUR LABELING FREQUENCY',2X,
  1'CONTOUR INTERVAL'//13X,I1,21X,I3,15X,E12.5//)
   IF (LT.EQ.999) LT=M100
   IF (M1.LE.2.OR.M1.GT.MD) GO TO 26
   IF (N1.LE.2.OR.N1.GT.LD) GO TO 28
   AS1=W/(N1-1)
   AS2=U/(M1-1)
   IF (AS1.LE.AS2) GO TO 8
   AS1=AS2
 8 CONTINUE
   DO 9 I=1,LD
   DO 9 J=1,MD
 9 H(I,J)=0
   XL=W
   YL=U
   READ (5,10) NH,(IHEAD(I),I=1,18)
10 FORMAT (I2,(18A4))
   READ (5,10) NX,(ITITLX(I),I=1,18)
   READ (5,10) NY,(ITITLY(I),I=1,18)
   YD=YM-Y1
   XD=XN-X1
   CALL GRAPH (X1,XN,Y1,YM,IV,IW,XL,YL,LN,ND2,NDEC2,ND3)
   CALL PLOT (0.0,0.0,-3)
   CALL HEADS (IHEAD,ITITLX,ITITLY,NX,NY,NH)
   CALL PLOT (0.0,0.0,-3)
   DO 11 J=1,M100
11 PHI(J)=0.0
   XF=XL
   IF (IMII.EQ.0) IC2=1
   IF (IMII.GT.4) GO TO 15
   CALL HEAD (X,Y,Z,H,PHI1,XL,YL,XD,YD,X1,Y1,YM,HT2,N1,M1,AS1)
   ZROSUP=0
   CALL PLOT (0.0,0.0,-3)
   IF (IMII.EQ.0) GO TO 13
   IF (IMII.EQ.1) GO TO 12
   IF (IMII.EQ.3) GO TO 13
   CALL HEAD (X,Y,Z,H,PHI1,XL,YL,XD,YD,X1,Y1,YM,HT2,N1,M1,AS1)
12 CONTINUE
   CALL PLOT (0.0,0.0,-3)
   IF (IMII.EQ.4) GO TO 14
   GO TO 16
13 CALL PLOT (0.0,0.0,-3)
14 CALL CONT (X11,XN1,Y11,YM1,N1,M1)
```

```
      XF=X(1)
      GO TO 16
   15 CALL PLOT (0.0,0.0,-3)
      CALL PLOT (0.0,0.0,-3)
   16 CALL PLOT (0.0,0.0,-3)
      J=0
   17 J=J+1
      READ (LK,18,END=21) Z(J),X(J),Y(J),ISOUR,LT,NDEC,ND,INT,HTS,HT2,IC
   18 FORMAT (3E12.5,I1,I3,2I1,I3,2F4.3,I2)
      IF (ISOUR.NE.0) GO TO 23
      IF (J.EQ.M100) GO TO 19
      GO TO 17
   19 WRITE (6,20) M100
   20 FORMAT (1H1,'THERE ARE',I5,' OR MORE LINE DATA CARDS IN A SINGLE S
     1ET OF LINE DATA CARDS.'/)
      J=J-1
      GO TO 17
   21 IF (J.EQ.1) GO TO 30
      WRITE (6,22)
   22 FORMAT (1H1,'NO END-OF-LINE DATA CARD FOUND FOR LAST SET OF LINE D
     1ATA CARDS.'/)
      GO TO 30
   23 N=J-1
      IF (ISOUR.EQ.0.OR.ISOUR.GT.4) ISOUR=1
      CALL NEWPEN (ISOUR)
      IF (N.LE.0) GO TO 25
      DO 24 IJ=1,N
   24 H1(IJ)=HTS
      CALL PTPLOT (X,Y,Z,IC,LT,INT,NDEC,ND,HT2,PHI,XL,YL,XD,YD,X1,Y1,N,H
     11)
      CALL PLOT (0.0,0.0,-3)
   25 J=0
      GO TO 17
   26 CONTINUE
      WRITE (6,27) MD
   27 FORMAT (1H1,5X,'NUMBER OF HORIZONTAL GRID LINES MUST BE IN THE RAN
     1GE 3 TO ',I3,'.'/)
      IF (N1.LE.2.OR.N1.GT.LD) GO TO 28
      STOP 1
   28 WRITE (6,29) LD
   29 FORMAT (1H1,5X,'NUMBER OF VERTICAL GRID LINES MUST BE IN THE RANGE
     1 3 TO ',I3,'.'/)
      STOP 1
   30 XXL=XF+10
      CALL PLOT (XXL,-60.0,999)
      STOP
      END
```

```
      SUBROUTINE LABELS
      REAL*8 DPN,FPN
      ENTRY XLAB(HT,ND,X,Y,IY,FPN,PHI,NDEC)
      DPN=FPN
      MD=MDIG(DPN,NDEC,ND)
      IF (MD.EQ.0) RETURN
      X1=X-(MD/2.0)*HT
      Y1=Y+IY*(HT+.04)+.02
      CALL NUMBIR (X1,Y1,HT,DPN,PHI,MD,NDEC)
      RETURN
      ENTRY YLAB(HT,ND,X,Y,IX,FPN,PHI,NDEC)
      DPN=FPN
      MD=MDIG(DPN,NDEC,ND)
      IF (MD.EQ.0) RETURN
      X1=X+(MD*HT+.04)*IX+.02
      Y1=Y-HT/2.0
      CALL NUMBIR (X1,Y1,HT,DPN,PHI,MD,NDEC)
      RETURN
      END

      SUBROUTINE HEAD (X,Y,Z,H,PHI1,XL,YL,XD,YD,X1,Y1,YM,HT2,M1,N1,AS1)
      REAL*8 RMAX
      REAL*8 Z(1),H(LD,MD)
      REAL*4 X(1),Y(1),           H2(101),PHI1( LD,MD),PHI2(101)
      INTEGER*4 F(20)
      COMMON LD,MD,CI(101),CIN,M100
      COMMON /C1C1/ IC2,ZROSUP
      COMMON /DIST/ SX,SY,ZFAC,HTF,IMII
      DATA ISII/1/
      ZROSUP=1
      L=10
      FIM=.625
      IF (IMII.EQ.1) FIM=1.0
      IF (HTF.LT..01) FIM=1.25*FIM
      IF (HTF.EQ.0) FIM=1.25
      RAD=57.2957795
      ZMAX=9.999
      IO=2
      ZZ=ZMAX*ZFAC
      M2=M1-1
      N2=N1-1
      YDD=YD/N2
      YDEL=YDD/3
      XDD=XD/M2
      XDEL=XDD/3
      IF (ISII.EQ.2) GO TO 2
      DO 1 J=1,N1
    1 Y(J)=YM-(J-1)*YDD-YDEL
      GO TO 4
```

```
    2 DO 3 J=1,N1
    3 Y(J)=YM-J*YDD+YDEL
    4 CONTINUE
      IF (ISII.EQ.2) GO TO 9
      WRITE (6,5)
    5 FORMAT (1H1,57X,'INPUT DATA MATRIX'/)
      READ (5,6) F
    6 FORMAT (20A4)
      DO 8 J=1,N1
      READ (L,F,END=39) (H(I,J),I=1,M1)
      WRITE (6,7) (H(I,J),I=1,M1)
    7 FORMAT (/(1X,10E13.5))
    8 CONTINUE
      WRITE (1) H
      REWIND 1
      GO TO 10
    9 READ (1) H
   10 CONTINUE
      IF (IC2.EQ.1) GO TO 38
      DO 18 J=1,N1
      DO 18 I=1,M1
      HY=0
      HX=0
      IF (I.EQ.M1.OR.J.EQ.N1) GO TO 12
      IF (ISII.EQ.1) GO TO 11
      IF (I.EQ.M2.OR.J.EQ.N2) GO TO 12
      HX=(H(I,J+1)-H(I+1,J+1))/SX
      HY=(H(I+1,J+1)-H(I+1,J))/SY
      GO TO 12
   11 IF (I.EQ.1.OR.J.EQ.1) GO TO 12
      HX=(H(I,J)-H(I+1,J))/SX
      HY=-(H(I,J)-H(I,J+1))/SY
   12 CONTINUE
      H(I,J)=SQRT(HX**2+HY**2)
      IF (HX.EQ.0) GO TO 16
      GO TO 17
   13 WRITE (6,14) H(I,J),I,J,ISII,ZMAX
   14 FORMAT (1X,'MAGNITUDE OF GRADIENT = ',E13.5,' FOR THE (',I3,',',I3
     1,') ELEMENT IN GRADIENT MATRIX ',I1,'.  MAGNITUDE ON MAP SET TO ',
     2F5.3/)
      GO TO 18
   15 PHI1(I,J)=0
      GO TO 18
   16 IF (HY.EQ.0) GO TO 15
   17 CONTINUE
      PHI1(I,J)=ATAN2(HY,HX)*RAD-90.0
      IF (H(I,J)*ZFAC.GT.ZMAX) GO TO 13
   18 CONTINUE
      LDLD=LD*MD
      HMAX=RMAX(H,LDLD)
```

```
      IF (HMAX.GT.0) GO TO 20
      WRITE (6,19)
   19 FORMAT (1H1,5X,'ALL GRADIENT VALUES ARE ZERO, CHECK INPUT DATA TO
     1SEE IF THE INPUT DATA MATRIX WAS READ IN CORRECTLY.'/)
      STOP 2
   20 CONTINUE
      HT22=AS1*HTF
      AS2=AS1/5
      IF (HT22.GT.AS2) HT22=AS2
      HT23=HT22
      DO 26 I=1,M2
      IF (ISII.EQ.2) GO TO 22
      DO 21 J=1,N2
   21 X(J)=X1+(I-1)*XDD+XDEL
      GO TO 24
   22 DO 23 J=1,N2
   23 X(J)=X1+I*XDD-XDEL
   24 CONTINUE
      DO 25 J=1,N2
      PHI2(J)=PHI1(I,J)
      H2(J)=(DABS(H(I,J))/HMAX)*(AS1*3/4.0-HT22-.05)+AS1/4
      H2(J)=H2(J)*FIM
      IF (H2(J).GT.AS1*1.1) H2(J)=AS1*1.1
      Z(J)=H(I,J)*ZFAC
      IF (Z(J).GT.ZMAX) Z(J)=ZMAX
   25 CONTINUE
      HT22=HT23*FIM
      CALL PTPLOT (X,Y,Z,1,-1,6,3,1,HT22,PHI2,XL,YL,XD,YD,X1,Y1,N2,H2)
      HT22=HT23
   26 CONTINUE
      WRITE (6,27) ISII
   27 FORMAT (1H1,59X,'GRADIENT MATRIX ',I1/)
      DO 29 J=1,N2
      WRITE (6,7) (H(I,J),I=1,M2)
      WRITE (IO,28) (H(I,J),I=1,M2)
   28 FORMAT (5E16.8)
   29 CONTINUE
      WRITE (6,30) ISII
   30 FORMAT (1H1,49X,'GRADIENT ANGLE MATRIX ',I1/)
      DO 33 J=1,N2
      DO 31 I=1,M2
   31 PHI2(I)=PHI1(I,J)+90
      WRITE (IO,32) (PHI2(I),I=1,M2)
   32 FORMAT (8F10.5)
   33 WRITE (6,34) (PHI2(I),I=1,M2)
   34 FORMAT (/(1X,10F13.3))
      IF (ISII.EQ.2) GO TO 36
      IF (IMII.EQ.1) RETURN
      WRITE (6,35)
   35 FORMAT (1H1,'GRADIENT MATRIX 1 COMPLETED.'/)
```

```
      ISII=2
      RETURN
   36 WRITE (6,37)
   37 FORMAT (1H1,'GRADIENT MATRIX 2 COMPLETED.'/)
   38 RETURN
   39 WRITE (6,40)
   40 FORMAT (1H1,5X,'INSUFFICIENT INPUT DATA FOUND'//)
      STOP 3
      END

      FUNCTION RMAX (X,N)
      REAL*8 X(1),XMAX,RMAX
      XMAX=X(1)
      DO 3 I=1,N
      IF (XMAX-X(I)) 1,1,2
    1 XMAX=X(I)
    2 CONTINUE
    3 CONTINUE
      RMAX=XMAX
      RETURN
      END

      SUBROUTINE GRAPH (X1,XN,Y1,YM,N,M,XL,YL,LN,ND,NDEC,ND2)
      REAL*8 YR,XR,DELX,DELY,DELXR,DELYR
      COMMON /FACS/ FAC
      REAL*4 ZRO/0.0/,HT/.2/,TIK/.125/
      LN=IABS(LN)
      YFAC=4/FAC
      DELX=XL/(N-1)
      DELY=YL/(M-1)
      CALL PLOT (5.0,-30.0,-3)
      CALL PLOT (ZRO,YFAC,-3)
      HTT=AMIN1(XL/((ND2+NDEC+2)*(N-2)+ND+NDEC+2),(YL/M-.01))
      YTK=TIK
      CALL PLOT (XL,ZRO,2)
      CALL PLOT (XL,YL,2)
      CALL PLOT (ZRO,YL,2)
      CALL PLOT (ZRO,ZRO,2)
      XR=X1
      YR=Y1
      Y=-YTK
      X=0
      DELXR=(XN-X1)/(N-1)
      DELYR=(YM-Y1)/(M-1)
      IF(HTT.LT.HT)  HT=AMAX1(HTT,.01)
      DO 2 I=1,N
      IF (LN.NE.1.AND.LN.NE.2)  GO TO 1
      CALL PLOT (X,ZRO,3)
      CALL PLOT (X,Y,2)
    1 CONTINUE
      NXD=ND2
```

```
      IF (I.EQ.1.OR.I.EQ.N) NXD=ND
      CALL XLAB (HT,NXD,X,Y,-1,XR,ZRO,NDEC)
      XR=X1+DELXR*I
      X=X+DELX
    2 CONTINUE
      Y=0
      X=XL+TIK
      DO 4 I=1,M
      IF (LN.NE.1.AND.LN.NE.2)  GO TO 3
      CALL PLOT (XL,Y,3)
      CALL PLOT (X,Y,2)
    3 CONTINUE
      NYD=ND2
      IF (I.EQ.1.OR.I.EQ.M) NYD=ND
      CALL YLAB (HT,NYD,X,Y,0,YR,ZRO,NDEC)
      YR=Y1+DELYR*I
      Y=Y+DELY
    4 CONTINUE
      XR=XN
      Y=YL+TIK
      X=XL
      DO 6 I=1,N
      IF (LN.NE.1.AND.LN.NE.2)  GO TO 5
      CALL PLOT (X,YL,3)
      CALL PLOT (X,Y,2)
    5 CONTINUE
      NXD=ND2
      IF (I.EQ.1.OR.I.EQ.N) NXD=ND
      CALL XLAB (HT,NXD,X,Y,0,XR,ZRO,NDEC)
      X=X-DELX
      XR=XN-DELXR*I
    6 CONTINUE
      YR=YM
      X=-TIK
      Y=YL
      DO 8 I=1,M
      IF (LN.NE.1.AND.LN.NE.2)  GO TO 7
      CALL PLOT (ZRO,Y,3)
      CALL PLOT (X,Y,2)
    7 NYD=ND2
      IF (I.EQ.1.OR.I.EQ.M) NYD=ND
      CALL YLAB (HT,NYD,X,Y,-1,YR,ZRO,NDEC)
      YR=YM-DELYR*I
      Y=Y-DELY
    8 CONTINUE
      IF (LN.EQ.1.OR.LN.EQ.3) GO TO 9
      RETURN
    9 X=0
      Y=0
      J1=0
```

```
      X2=DELX
      DO 10 I=1,N,2
      CALL PLOT (X,Y,3)
      CALL PLOT (X,YL,2)
      J1=J1+1
      IF (J1.EQ.N) GO TO 11
      CALL PLOT (X2,YL,3)
      CALL PLOT (X2,Y,2)
      J1=J1+1
      IF (J1.EQ.N) GO TO 11
      X=J1*DELX
      X2=(J1+1)*DELX
   10 CONTINUE
   11 J1=0
      Y=0
      X=0
      Y2=DELY
      DO 12 I=1,M,2
      CALL PLOT (XL,Y,3)
      CALL PLOT (X,Y,2)
      J1=J1+1
      IF (J1.EQ.M) GO TO 13
      CALL PLOT (X,Y2,3)
      CALL PLOT (XL,Y2,2)
      J1=J1+1
      IF (J1.EQ.M) GO TO 13
      Y=J1*DELY
      Y2=(J1+1)*DELY
   12 CONTINUE
   13 RETURN
      ENTRY HEADS(IHEAD,ITITLX,ITITLY,NX,NY,NH)
      INTEGER*4            ITITLX(18),ITITLY(18),IHEAD(18)
      HT=.2
      N4=(MAX0(ND,ND2)+1+NDEC)
      YT=YL+HT*2+.25
      XT=XL+N4*HT*2+.25
      IF (NX.EQ.0) GO TO 14
      Y=-4*HT
      X=XL/2-NX*HT/2.0
      IF (X.LT.-N4*HT-.125) HT=XT/NX
      X=XL/2-NX*HT/2.0
      CALL PLOT (X,Y,3)
      CALL SYMBOL (X,Y,HT,ITITLX,0.0,NX)
      HT=.2
   14 IF (NY.EQ.0) GO TO 15
      X=-(NYD+NDEC+3)*HT
      Y=YL/2+NY*HT/2.0
      IF (Y.GT.YT-HT-.125) HT=YT/NY
      Y=YL/2+NY*HT/2
      CALL PLOT (X,Y,3)
```

```
      CALL SYMBOL (X,Y,HT,ITITLY,270.0,NY)
      HT=.2
      HHT=1.3333*HT
   15 IF (NH.EQ.0) RETURN
      X=XL/2-NH*HHT/2.0
      Y=YL+TIK+HT*2.
      IF (X.LT.-N4*HT-.125) HHT=XT/NH
      X=XL/2-NH*HHT/2
      CALL PLOT (X,Y,3)
      CALL SYMBOL (X,Y,HHT,IHEAD,0.0,NH)
      RETURN
      ENTRY XYPLOT(XD,YD,NPTS,INC,LINTYP,INTEQ,XP,YP)
      REAL*4 XD(1),YD(1),XP(1),YP(1)
      DO 16 J=1,NPTS
      XP(J)=XL*(XD(J)-X1)/(XN-X1)
   16 YP(J)=YL*(YD(J)-Y1)/(YM-Y1)
      CALL LINE (XP,YP,NPTS,INC,LINTYP,INTEQ)
      RETURN
      END

      SUBROUTINE NUMBIR (X,Y,HT,Z,PHI,ND,NDEC)
      REAL*8 Z
      LOGICAL*1 W(99),IZ2(99),FMT(8)/'(','F','0','0','.','0','0',')'/
      LOGICAL*1 FM(8)/'(','F','0','0','.','0','0',')'/,BLK/' '/
      INTEGER*4 IZ3(25)
      INTEGER*2 N1,N2
      EQUIVALENCE (IZ3(1),IZ2(1))
      DO 1 J=1,8
    1 FMT(J)=FM(J)
      K2=ND/10
      K1=ND-K2*10
      N1=K2*256+K1
      CALL LOR (FMT(3),2,N1,2)
      K4=NDEC/10
      K3=NDEC-K4*10
      N2=K4*256+K3
      CALL LOR (FMT(6),2,N2,2)
      CALL CORE (W,99)
      WRITE (100,FMT) Z
      CALL CORE (W,99)
      READ (100,2) (IZ2(I),I=1,ND)
    2 FORMAT (99A1)
      DO 3 I=1,ND
      IF(IZ2(I).EQ.BLK)  IZ2(I)=FM(3)
    3 CONTINUE
      CALL SYMBOL(X,Y,HT,IZ3,PHI,ND)
      RETURN
      END

      SUBROUTINE ENDPT (X1,Y1,X2,Y2,XK,YK,ISW)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 X1,Y1,X2,Y2,XK,YK
      EPS=1.0D-7
```

```
      XL=DBLE(XK)
      YL=DBLE(YK)
      XO=DBLE(X1)
      YO=DBLE(Y1)
      RX=XO
      RY=YO
      XF=DBLE(X2)
      YF=DBLE(Y2)
      XE=XF
      YE=YF
      KK=0
      IF (XE.EQ.RX) GO TO 7
      SLOPE=(YE-RY)/(XE-RX)
    1 KK=KK+1
      IF (KK.GT.4) GO TO 4
      XSW=0
      YSW=0
      IF (XE.LT.0) XSW=-1
      IF (YE.LT.0) YSW=-1
      IF (XE.GT.XL) XSW=1
      IF (YE.GT.YL) YSW=1
      IF (XSW.NE.0) GO TO 5
      IF (YSW.NE.0) GO TO 6
    2 D3=DSQRT((XF-XO)**2+(YF-YO)**2)
      D1=DSQRT((XE-XO)**2+(YE-YO)**2)
      D2=DSQRT((XE-XF)**2+(YE-YF)**2)
      IF ((D1+D2).EQ.0) GO TO 3
      IF ((D3/(D1+D2)+EPS).LT.1) GO TO 4
    3 ISW=0
      X2=XE
      Y2=YE
      RETURN
    4 ISW=2
      X2=XF
      Y2=YF
      RETURN
    5 CONTINUE
      IF (XSW.GT.0) XE=XL
      IF (XSW.LT.0) XE=0
      YE=RY+SLOPE*(XE-RX)
      GO TO 1
    6 IF (YSW.GT.0) YE=YL
      IF (YSW.LT.0) YE=0
      IF (SLOPE.EQ.0) GO TO 1
      XE=(YE-RY)/SLOPE+RX
      GO TO 1
    7 IF (YE.LT.0) YE=0
      IF (YE.GT.YL) YE=YL
      IF (XE.LT.0) GO TO 4
```

```
      IF (XE.GT.XL) GO TO 4
      GO TO 2
      END

      SUBROUTINE CONTIN (ZMAX,ZMIN,N,DI)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 CI,CIN,DI
      COMMON LD,MD,CI(101),CIN
      ZO=0
      READ (5,1) IS,M,L1,L2,ZO,DI
    1 FORMAT (I1,3I3,2E12.5)
      IF (IS.EQ.0.OR.IS.GT.5) GO TO 2
      N=M
    2 IF (N.LE.0.OR.N.GT.100) GO TO 26
      IF (L2.GE.L1) GO TO 3
      LM2=L2
      L2=L1
      L1=LM2
    3 WRITE (6,4) IS,M,L1,L2
    4 FORMAT (1H1,42X,'AUTOMATIC CONTOUR INTERVAL GENERATION PARAMETERS'
     1//5X,'OPTION IND',2X,'MAX NO OF CONTOURS',2X,'CONTOUR INTERVAL FAC
     2TORS'/10X,I1,13X,I4,15X,I3,6X,I3//)
      WRITE (6,5) ZO,DI
    5 FORMAT (/5X,'TARGET CONTOUR ELEVATION ',2X,'ELEVATION ADJUSTMENT F
     1ACTOR'//13X,E12.5,18X,E12.5///)
      IF (IS.EQ.0.OR.IS.GT.5) IS=5
      GO TO (6,11,14,19,24), IS
    6 READ (5,7) (CI(I),I=1,N)
    7 FORMAT (6E12.5)
    8 IF (N.EQ.1) GO TO 10
      N2=N-1
      DO 9 I=1,N2
      I2=I+1
      DO 9 J=I2,N
      IF (CI(I).LE.CI(J)) GO TO 9
      TCI=CI(I)
      CI(I)=CI(J)
      CI(J)=TCI
    9 CONTINUE
   10 CONTINUE
      RETURN
   11 J=0
      DO 12 I=1,N
      J=J+1
      CI(J)=CIN*(L1+I-1)
      IF (CI(J).LT.ZMIN) J=J-1
      IF (CI(J).GT.ZMAX) GO TO 13
   12 CONTINUE
      RETURN
   13 N=J-1
```

```
      RETURN
   14 J=0
      DO 17 L=L1,L2
      DO 17 I=1,9
      J=J+1
      IF (J.GT.N) GO TO 8
      CI(J)=I*10.0D0**L+Z0
      CIJ=CI(J)
      IF (CI(J).GT.ZMAX.OR.CIJ.LT.ZMIN) GO TO 15
      J=J+1
      IF (J.GT.N) GO TO 8
   15 IF (-CIJ.LT.ZMIN.OR.-CIJ.GT.ZMAX) GO TO 16
      CI(J)=-CIJ
      GO TO 17
   16 J=J-1
      IF (CIJ.GT.ZMAX.AND.-CIJ.LT.ZMIN) GO TO 18
   17 CONTINUE
      N=J
      GO TO 8
   18 N=J
      GO TO 8
   19 J=0
      DO 22 L=L1,L2
      DO 22 I=1,9
      J=J+1
      IF (J.GT.N) GO TO 8
      CI(J)=-I*10.0D0**L+Z0
      CIJ=CI(J)
      IF (CI(J).LT.ZMIN.OR.CI(J).GT.ZMAX) GO TO 20
      J=J+1
      IF (J.GT.N) GO TO 8
   20 IF (-CIJ.GT.ZMAX.OR.-CIJ.LT.ZMIN) GO TO 21
      CI(J)=-CIJ
      GO TO 22
   21 J=J-1
      IF (CIJ.LT.ZMIN.AND.-CIJ.GT.ZMAX) GO TO 23
   22 CONTINUE
      N=J
      GO TO 8
   23 N=J
      GO TO 8
   24 CONTINUE
      L1=ZMIN/CIN
      IF (ZMIN.LT.0) GO TO 25
      IF (L1*CIN.EQ.ZMIN) GO TO 25
      L1=L1+1
   25 L2=ZMAX/CIN
      GO TO 11
   26 WRITE (6,27)
   27 FORMAT (1H1,'NUMBER OF CONTOUR LEVELS REQUESTED IS INVALID.')
```

```
      STOP 10
      END

      SUBROUTINE KEY (XL1,YL1,PCI,XF)
      REAL*8 PCI
      INTEGER*4 IPCIA(3)
      LOGICAL*1 PCIA(12),WW(12)
      EQUIVALENCE (PCIA(1),IPCIA(1))
      DATA IPEN,IYL,YD,HY/ 0,0,.165,.125/
      IYL=IYL+1
      XXXL=XL1
      YYYL=YL1-IYL*YD-.1
      IF (YYYL.GT.0) GO TO 1
      XL1=XL1+17*HY
      XXXL=XL1
      IYL=1
      YYYL=YL1-IYL*YD
    1 CALL PLOT (XXXL,YYYL,3)
      CALL CORE (WW,12)
      WRITE (100,2) PCI
    2 FORMAT (E12.5)
      CALL CORE (WW,12)
      READ (100,3) PCIA
    3 FORMAT (12A1)
      CALL SYMBOL (XXXL,YYYL,HY,IPCIA,0.0,12)
      XXXL=XXXL+14*HY
      XF=XXXL
      IPEN=IPEN+1
      IPEN=IPEN-(IPEN/4)*4
      JPEN=IPEN+1
      CALL NEWPEN (JPEN)
      RETURN
      END

      SUBROUTINE PTPLOT (X,Y,Z,INC,LINTYP,INT,NDEC,ND,HT,PHI,XL,YL,XD,YD
     1,X1,Y1,N,H2)
      REAL*4 PHI(1),H2(1)
      REAL*4 X(1),Y(1),XXN,YYN,RF
      REAL*8 Z(1)
      REAL*8 Q/0.0D0/
      COMMON /C1C1/ IC2,ZROSUP
      IF (INC.LE.0) INC=1
      M=0
      K=0
      L=IABS(LINTYP)
      DO 14 J=1,N,INC
      K=K+1
      XI=(X(J)-X1)*XL/XD
      YI=(Y(J)-Y1)*YL/YD
      XB=XI
```

```
      YB=YI
      IF ((J+INC).GT.N.OR.LINTYP.LT.0)  GO TO 1
      XE=(X(J+INC)-X1)*XL/XD
      YE=(Y(J+INC)-Y1)*YL/YD
      GO TO 2
    1 XE=XI
      YE=YI
    2 IE=0
      II=0
      IF (IC2.EQ.1) GO TO 3
      CALL ENDPT (XI,YI,XE,YE,XL,YL,ISW)
      IF (ISW.NE.0) IE=1
      CALL ENDPT (XE,YE,XI,YI,XL,YL,ISW)
      IF (ISW.NE.0) II=1
      IF (IE.NE.0.AND.II.NE.0) GO TO 11
    3 CONTINUE
      IF (K.LT.L) GO TO 6
      IF (XB.NE.XI.OR.YB.NE.YI) GO TO 9
      IF ((Z(J).EQ.0).AND.(LINTYP.LT.0).AND.ZROSUP.EQ.1) GO TO 14
      IF (LINTYP.EQ.0) GO TO 7
      H1=H2(J)
      PHI1=PHI(J)
      IF (HT.LT..01) GO TO 4
      MD=MDIG(Z(J),NDEC,ND)
      IF (MD.EQ.0) GO TO 6
      XX=XI-(MD/2.0)*HT
      IF (XX.LT.0.05) XX=0.05
      IF (XX.GT.(XL-MD*HT-0.05)) XX=XL-MD*HT-0.05
      YY=YI+.05+H1/2
      IF (YY.GT.(YL-HT/2-.05)) YY=YL-HT/2-H1-.05
      CALL NUMBIR (XX,YY,HT,Z(J),0.0,MD,NDEC)
    4 CONTINUE
      IF (H1.LT..01) GO TO 5
      CALL SYMBOL (XI,YI,H1,INT,PHI1,-1)
    5 K=0
    6 IF (LINTYP.LT.0) GO TO 10
    7 CONTINUE
      IF (XI.EQ.XXN.AND.YI.EQ.YYN) GO TO 8
      CALL PLOT (XI,YI,3)
    8 CONTINUE
      XXN=XE
      YYN=YE
      CALL PLOT (XE,YE,2)
      GO TO 14
    9 IF (LINTYP.LT.0) GO TO 14
      GO TO 7
   10 CONTINUE
      CALL PLOT (XI,YI,3)
      CALL PLOT (XI,YI,2)
      GO TO 14
```

```
11 CONTINUE
   IF ((J+INC).GT.N.AND.LINTYP.GE.0.AND.M.EQ.1) GO TO 14
   WRITE (6,12)
12 FORMAT (1X,'RAY DOES NOT INTERSECT GRAPH OR POINT BEING PLOTTED IS
  1 NOT ON GRAPH.')
   WRITE (6,13) XI,YI,XE,YE,Z(J)
13 FORMAT (1X,5E24.16)
14 M=1
   RETURN
   END

   FUNCTION MDIG(FPN,NDEC,ND)
   REAL*8 EJX
   REAL*8 FPN,DPN,EPN
   INTEGER*4 IBIG/Z7FFFFFFF/
   MDIG=0
   IF (NDEC.EQ.0.AND.ND.EQ.0) RETURN
   IO=6
   IF (FPN.LT.0) MDIG=1
   IF ((DABS(FPN)/10**ND).GT.IBIG) GO TO 2
   IX=DABS(FPN)/10**ND
   FPN=DABS(FPN)-IX*10**ND
   FPN=FPN*(-1)**MDIG
   EPN=DABS(FPN)+.5D0*(10.0D0**(-NDEC))
   JX=10**NDEC
   IX=EPN*DFLOAT(JX)
   EPN=DFLOAT(IX)/DFLOAT(JX)
   EJX=DFLOAT(JX)/DFLOAT(JX)
   IF (EPN.LT.EJX) GO TO 6
   I=0
   IF (ND.EQ.0) GO TO 4
   DO 1 I=1,ND
   DPN=EPN/(10**I)
   IF (DPN-EJX) 4,5,1
 1 CONTINUE
 2 WRITE (IO,3) FPN,NDEC,ND
 3 FORMAT (1X,'THE FOLLOWING NUMBER CANNOT BE FORMATTED FOR PLOTTING
  1AS REQUESTED.',2X,E16.9,'  NDEC=',I2,'  ND=',I2)
   MDIG=ND+NDEC+MDIG+1
   GO TO 7
 4 MDIG=MDIG+I+NDEC+1
   GO TO 7
 5 MDIG=MDIG+I+NDEC+2
   GO TO 7
 6 MDIG=1+NDEC+MDIG+MINO(1,ND)
 7 RETURN
   END
```

```
      FUNCTION RMIN (X,N)
      REAL*8 X(1),XMIN,RMIN
      XMIN=X(1)
      DO 3 I=1,N
      IF (XMIN-X(I)) 2,1,1
    1 XMIN=X(I)
    2 CONTINUE
    3 CONTINUE
      RMIN=XMIN
      RETURN
      END

      SUBROUTINE CONT (X11,XN1,Y11,YM1,M1,N1)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 AM(101,51),AN(101,51),MIN,PL,ZRO
      LOGICAL*1 PL1,MIN1,ZRO1
      INTEGER*4 CKEY(5)/'CONTOUR COLOR KEY    '/
      REAL*4 HZL
      REAL*4 XF,YF,XL1,YL1
      REAL*4 XN2,YM2,X12,Y12,XL,YL,HT2,XXXL
      REAL*4 U(2000),V(2000),H1(1000),PHI(1000)
      REAL*8 Z(1000),H(101,51)
      REAL*8 X(1000),Y(1000)
      REAL*8 XP(3),YP(3),EPS/1.0D-13/
      EQUIVALENCE (XP,XX1),(YP,YY1)
      EQUIVALENCE (U,X),(V,Y)
      COMMON LD,MD,CI(101),CIN,M100
      COMMON /CON /X1,XN,Y1,YM,XD,YD,XX1,XX2,XX3,YY1,YY2,YY3,PCI,H,PL,
     1ZRO,MIN
      COMMON /C1C1/ IC2
      COMMON /DIST/SX,SY,ZFAC,HTF,IMII
      COMMON /H1/H1,PHI,XL,YL,INT,LT,NDEC,ND,IC,HT2,LTF,ND4,NDEC2
      COMMON /IJ/I11,J11,I12,J12,II1,II2
      COMMON /ZXY/ Z,X,Y
      DATA  PL1,MIN1,ZRO1/'+','-','0'/
      DIS(A,B,C,E)=DSQRT((A-C)**2+(B-E)**2)
      M99=M100-1
      IC2=1
      IRS=21
      IRS=0
      IRR=100
      XL1=XL+(NDEC2+ND4+5)*.2
      YL1=YL
      DO 1 I=1,M100
      PHI(I)=0
    1 H1(I)=HT2
      I=1
      J=1
      X1=X11
      Y1=Y11
```

```
      XN=XN1
      YM=YM1
      YM=YM-Y1
      Y1=0
      XN=XN-X1
      X1=0
      IF (LTF.EQ.0) LTF=1
      PL=PL1
      MIN=MIN1
      ZRO=ZRO1
      N2=N1-1
      M2=M1-1
      LDLD=LD*MD
      DO 2 I=1,LD
      DO 2 J=1,MD
      AM(I,J)=PL
    2 AN(I,J)=PL
      IBC=0
      YD=(YM-Y1)/N2
      XD=(XN-X1)/M2
      YM=YD*N2
      XN=XD*M2
      REWIND 1
      READ (1) H
      REWIND 1
      DO 3 I=1,M1
    3 X(I)=X1+(I-1)*XD
      DO 4 J=1,N1
    4 Y(J)=YM-(J-1)*YD
      WRITE (6,5)
    5 FORMAT (1H1,53X,'HORIZONTAL GRID COORDINATES'/)
      WRITE (6,7) (X(I),I=1,M1)
      WRITE (6,6)
    6 FORMAT (1H0,54X,'VERTICAL GRID COORDINATES'/)
      WRITE (6,7) (Y(J),J=1,N1)
    7 FORMAT (/(1X,5E15.8))
      HMAX=-1.0E50
      HMIN=1.0E50
      DO 8 J=1,N1
      HMM=RMIN(H(1,J),M1)
      IF (HMIN.GT.HMM) HMIN=HMM
      HMM=RMAX(H(1,J),M1)
      IF (HMAX.LT.HMM) HMAX=HMM
    8 CONTINUE
      NCIN=100
      CALL CONTIN (HMAX,HMIN,NCIN,DI)
      IF (DI.EQ.0) GO TO 13
      DO 9 I=1,M1
      DO 9 J=1,N1
      DO 9 L=1,NCIN
      ACI=DABS(DI*CI(L))
```

```
      IF (ACI.EQ.0) ACI=DI
      IF (DABS(CI(L)-H(I,J)).LT.ACI) H(I,J)=CI(L)+ACI
    9 CONTINUE
      WRITE (6,10)
   10 FORMAT (1H1,53X,'ADJUSTED INPUT DATA MATRIX'/)
      DO 12 J=1,N1
      WRITE (6,11) (H(I,J),I=1,M1)
   11 FORMAT (/(1X,10E13.5))
   12 CONTINUE
   13 CONTINUE
      K1=1
      K10=0
      DO 59 K=1,NCIN
      PCI=CI(K)
      I11=1
      J11=1
      I12=1
      J12=1
      II1=1
      II2=1
      DO 15 I=1,M1
      DO 15 J=1,N1
      IF (AM(I,J).EQ.MIN) GO TO 14
      AM(I,J)=PL
   14 CONTINUE
      IF (AN(I,J).EQ.MIN) GO TO 15
      AN(I,J)=PL
   15 CONTINUE
      M=0
      IK3=0
      JK1=0
      JK2=0
      IK1=0
      IK2=0
   16 CONTINUE
      IF (IK3.EQ.1) GO TO 18
      IF (K1.EQ.2) GO TO 17
      CALL FINDTB (K1,M1,N1,AM,I,J)
      GO TO 24
   17 CALL FINDTB (K1,M1,N1,AN,I,J)
      GO TO 24
   18 IF (K1.EQ.2) GO TO 19
      CALL FINDTI (K1,M1,N1,AM,I,J)
      GO TO 20
   19 CALL FINDTI (K1,M1,N1,AN,I,J)
   20 CONTINUE
      IF (I.EQ.0) GO TO 21
      IF (K1.EQ.2) GO TO 26
      GO TO 25
   21 IF (K1.EQ.1) GO TO 22
      IF (K1.EQ.2) GO TO 23
```

```
      STOP 5                                                            CON 141
   22 JK1=1                                                             CON 142
      IF (JK2.EQ.1) GO TO 58                                            CON 143
      K1=2                                                              CON 144
      GO TO 18                                                          CON 145
   23 JK2=1                                                             CON 146
      IF (JK1.EQ.1) GO TO 58                                            CON 147
      K1=1                                                              CON 148
      GO TO 18                                                          CON 149
   24 CONTINUE                                                          CON 150
      IF (I.EQ.0) GO TO 29                                              CON 151
      IF (K1.EQ.2) GO TO 26                                             CON 152
   25 CONTINUE                                                          CON 153
      CALL CORD (I,J,NP,AM,K1,K2)                                       CON 154
      GO TO 27                                                          CON 155
   26 CALL CORD (I,J,NP,AN,K1,K2)                                       CON 156
   27 CONTINUE                                                          CON 157
      IF (NP.EQ.3) K10=K                                                CON 158
      IF (NP.EQ.3) IRR=89                                               CON 159
      IF (NP.EQ.0) GO TO 36                                             CON 160
      IF (NP.LT.0) STOP 9                                               CON 161
      DO 28 L=1,NP                                                      CON 162
      IF ((XP(L).EQ.X1).OR.(XP(L).EQ.XN).OR.(YP(L).EQ.Y1).OR.(YP(L).EQ.YCON 163
     1M)) GO TO 33                                                      CON 164
   28 CONTINUE                                                          CON 165
      GO TO 34                                                          CON 166
   29 IF (K1.EQ.1) GO TO 30                                             CON 167
      IF (K1.EQ.2) GO TO 31                                             CON 168
      STOP 6                                                            CON 169
   30 IK1=1                                                             CON 170
      IF (IK2.EQ.1) GO TO 32                                            CON 171
      K1=2                                                              CON 172
      GO TO 16                                                          CON 173
   31 IK2=1                                                             CON 174
      IF (IK1.EQ.1) GO TO 32                                            CON 175
      K1=1                                                              CON 176
      GO TO 16                                                          CON 177
   32 IK3=1                                                             CON 178
      GO TO 18                                                          CON 179
   33 TE=XX1                                                            CON 180
      XX1=XP(L)                                                         CON 181
      XP(L)=TE                                                          CON 182
      TE=YY1                                                            CON 183
      YY1=YP(L)                                                         CON 184
      YP(L)=TE                                                          CON 185
   34 DO 35 L=1,NP                                                      CON 186
      X(L)=XP(L)                                                        CON 187
      Y(L)=YP(L)                                                        CON 188
   35 Z(L)=PCI                                                          CON 189
   36 M=NP                                                              CON 190
```

```
37 CONTINUE
   CALL FINDT (K1,M1,N1,I,J,M)
   IF (I.EQ.0.OR.J.EQ.0) GO TO 51
   IF (K1.EQ.2) GO TO 38
   CALL CORD (I,J,NP,AM,K1,K2)
   GO TO 39
38 CALL CORD (I,J,NP,AN,K1,K2)
39 CONTINUE
   IF (NP.EQ.2.OR.NP.EQ.1) GO TO 40
   IF (NP.EQ.3) K10=K
   IF (NP.EQ.3) IRR=89
   IF (NP.EQ.0) GO TO 51
40 CONTINUE
   IF (NP.EQ.3) WRITE (6,41)
41 FORMAT (///1X,'CONTOUR PASSES THROUGH 3 POINTS ON BOUNDARY OF A TR
  1IANGLE.'//)
   IF (NP.LE.0.OR.NP.GE.3) STOP 9
   CALL XMSORT (NP,M)
   L=1
   IF (X(M).EQ.XP(L).AND.Y(M).EQ.YP(L)) GO TO 48
   BD=DIS(X(M),Y(M),XP(1),YP(1))
   IF (DABS(X(M)-XP(L)).LT.EPS.AND.DABS(Y(M)-YP(L)).LT.EPS) GO TO 42
   GO TO 44
42 IRS=IRS+1
   IF (IRS.GT.20) GO TO 48
   WRITE (6,43) X(M),Y(M),XP(1),YP(1)
43 FORMAT (1X,'CONTOUR DOES NOT CONNECT EXACTLY'/1X,4(Z16,2X))
   GO TO 48
44 CONTINUE
   IBC=IBC+1
   DO 45 L=1,NP
   X(M+L)=XP(L)
   Y(M+L)=YP(L)
45 Z(M+L)=PCI
   M=M+NP
   IF (IBC.GT.10) GO TO 47
   WRITE (6,46) X(M),Y(M),XP(1),YP(1),BD
46 FORMAT (/1X,'BREAK IN CONTOUR DETECTED'/1X,4E16.9,'  DIST = ',E16.
  19/)
47 CONTINUE
   GO TO 50
48 IF (NP.EQ.1) GO TO 37
   X(M+1)=XP(2)
   Y(M+1)=YP(2)
   Z(M+1)=PCI
   IF (NP.LT.3) GO TO 49
   X(M+1)=XP(3)
   Y(M+1)=YP(3)
   Z(M+1)=PCI
   M=M+1
```

```
49 M=M+1
   IF (M.GE.M99) GO TO 60
   IF ((X(M).EQ.X(1)).AND.(Y(M).EQ.Y(1))) GO TO 51
50 IF (K2.EQ.0) GO TO 37
51 CONTINUE
   IF ((X(M).EQ.X1.OR.X(M).EQ.XN).OR.(Y(M).EQ.Y1.OR.Y(M).EQ.YM)) GO T
  10 54
   IF ((X(M).EQ.X(1)).AND.(Y(M).EQ.Y(1))) GO TO 54
   DM1=DIS(X(1),Y(1),X(M),Y(M))
   IF (DM1.GT.EPS) GO TO 52
   M=M+1
   IF (M.GE.M99) GO TO 60
   X(M)=X(1)
   Y(M)=Y(1)
   Z(M)=Z(1)
52 WRITE (6,53) PCI
53 FORMAT (///1X,G12.5,' CONTOUR DOES NOT CLOSE OR END ON BOUNDARY OF
  1 DOMAIN.'///)
54 CONTINUE
   IF (IRR.GT.90) GO TO 56
   WRITE (6,55) (X(L),Y(L),L=1,M)
55 FORMAT (/(1X,10E13.6))
56 CONTINUE
   DO 57 J=1,M
   U(J)=X(J)
57 V(J)=Y(J)
   XN2=XN
   YM2=YM
   X12=X1
   Y12=Y1
   JT=M99
   IF ((K-1).EQ.((K-1)/LTF)*LTF) JT=LT
   CALL PTPLOT (U,V,Z,IC,JT,INT,NDEC,ND,HT2,PHI,XL,YL,XN2,YM2,X12,Y12
  1,M,H1)
   GO TO 16
58 CONTINUE
   CALL KEY (XL1,YL1,PCI,XF)
59 CONTINUE
   XXXL=XL+(ND4+NDEC2+3)*.2
   YF=YL
   CALL NEWPEN (1)
   CALL PLOT (XXXL,0.0,3)
   CALL PLOT (XF,0.0,2)
   CALL PLOT (XF,YF,2)
   CALL PLOT (XXXL,YF,2)
   CALL PLOT (XXXL,0.0,2)
   HZL=.15
   XXXL=((XXXL+XF)/2)-17*HZL/2
   YF=YL+HZL
   CALL SYMBOL (XXXL,YF,HZL,CKEY,0.0,17)
```

```
      X(1)=XF
      IC2=0
      RETURN
   60 WRITE (6,61)
   61 FORMAT (1H1,'TOO MANY LINE SEGMENTS IN A CONTOUR'/)
      STOP 999
      END

      SUBROUTINE XMSORT (NP,M)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 XP(3),YP(3)
      REAL*8 Z(1000),X(1000),Y(1000),T,XPD,XPD2
      COMMON /CON /X1,XN,Y1,YM,XD,YD,XP,YP
      COMMON /ZXY/ Z,X,Y
      IF (NP.LE.1) RETURN
      MP=NP
      XPD=DSQRT((X(M)-XP(1))**2+(Y(M)-YP(1))**2)
      I1=1
      DO 2 I=2,MP
      XQD=DSQRT((X(M)-XP(I))**2+(Y(M)-YP(I))**2)
      IF (XPD.LE.XQD) GO TO 2
    1 T=XP(I1)
      XP(I1)=XP(I)
      XP(I)=T
      T=YP(I1)
      YP(I1)=YP(I)
      YP(I)=T
      I1=I
      XPD=XQD
    2 CONTINUE
      IF (MP.EQ.2) RETURN
      XPD=DSQRT((X(M)-XP(2))**2+(Y(M)-YP(2))**2)
      XQD=DSQRT((X(M)-XP(3))**2+(Y(M)-YP(3))**2)
      IF (XPD.LE.XQD) RETURN
      MP=2
      I=2
      I1=3
      GO TO 1
      END

      SUBROUTINE FINDT (K,M1,N1,I,J,M)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(1000),Y(1000),Z(1000)
      COMMON /CON /X1,XN,Y1,YM,XD,YD,XX1,XX2,XX3,YY1,YY2,YY3,PCI
      COMMON /ZXY/Z,X,Y
      IF (M.LE.0) RETURN
      IF (M.GT.999) GO TO 16
      Y2=YM-YD*(J-1)
      Y3=YM-YD*J
      X2=X1+XD*(I-1)
```

```
      X3=X1+XD*I
      IF (X(M).EQ.X2) GO TO 1
      IF (X(M).EQ.X3) GO TO 4
      GO TO 2
    1 IF (Y(M).EQ.Y2) GO TO 13
      IF (Y(M).EQ.Y3) GO TO 10
      GO TO 7
    2 IF (Y(M).EQ.Y2) GO TO 5
      IF (Y(M).EQ.Y3) GO TO 6
      IF (K.EQ.1) GO TO 3
      K=1
      RETURN
    3 K=2
      RETURN
    4 IF (Y(M).EQ.Y2) GO TO 11
      IF (Y(M).EQ.Y3) GO TO 12
      I=I+1
      GO TO 8
    5 J=J-1
      GO TO 8
    6 J=J+1
      GO TO 8
    7 I=I-1
    8 IF (J.EQ.N1) J=0
      IF (I.EQ.M1) I=0
      IF (K.EQ.1) GO TO 9
      K=1
      RETURN
    9 K=2
      RETURN
   10 J1=J+1
      I1=I
      GO TO 14
   11 I1=I+1
      J1=J
      GO TO 14
   12 I1=I+1
      J1=J+1
      GO TO 14
   13 I1=I
      J1=J
   14 WRITE (6,15)
   15 FORMAT (1X,'CONTOUR PASSES THROUGH THE VERTEX OF A TRIANGLE.')
      STOP 4
   16 WRITE (6,17)
   17 FORMAT (1H1,'TOO MANY LINE SEGMENTS IN A CONTOUR'/)
      STOP 999
      END
```

```
      SUBROUTINE CORD (I,J,NP,A,K1,K2)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 XS(3),YS(3),HH(3),H(101,51)
      LOGICAL*1 A( LD,MD),MIN,PL,ZRO
      COMMON LD,MD
      COMMON /CON /X1,XN,Y1,YM,XD,YD,XX1,XX2,XX3,YY1,YY2,YY3,PCI,H,PL,
     1ZRO,MIN
      PCD=PCI
      XX3=0
      YY3=0
      K2=0
      IF (A(I,J).NE.PL) K2=1
      NP=2
      IF (K1.EQ.2) GO TO 17
      HH(1)=H(I,J)
      HH(2)=H(I+1,J)
      HH(3)=H(I,J+1)
      XS(1)=X1+(I-1)*XD
      XS(2)=X1+I*XD
      XS(3)=XS(1)
      YS(1)=YM-(J-1)*YD
      YS(2)=YS(1)
      YS(3)=YM-J*YD
      CALL HSORT (HH,XS,YS)
      IF (HH(3).GT.PCI) GO TO 1
      IF (HH(3).EQ.PCI) GO TO 14
      NP=0
      A(I,J)=MIN
      RETURN
    1 IF (HH(2).GT.PCI) GO TO 7
      IF (XS(2).EQ.XS(1)) GO TO 5
      IF (YS(2).EQ.YS(1)) GO TO 3
      IF (XS(3).EQ.XS(1)) GO TO 2
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY1=YS(3)
      XX2=XS(3)
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(2))
      GO TO 30
    2 CONTINUE
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(2))
      YY1=YS(3)
      XX2=XS(3)
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      GO TO 30
    3 IF (XS(3).EQ.XS(2)) GO TO 4
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(2))
      YY1=YS(1)-YD*(PCD-HH(2))/(HH(3)-HH(2))
      XX2=XS(3)
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      GO TO 30
```

```
    4 CONTINUE
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY1=YS(2)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      XX2=XS(3)
      YY2=YS(2)-YD*(PCD-HH(2))/(HH(3)-HH(2))
      GO TO 30
    5 IF (YS(1).EQ.YS(3)) GO TO 6
      XX1=XS(2)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY1=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      XX2=XS(2)+XD*(PCD-HH(2))/(HH(3)-HH(2))
      YY2=YS(3)
      GO TO 30
    6 CONTINUE
      XX1=XS(1)+XD*(PCD-HH(2))/(HH(3)-HH(2))
      YY1=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(2))
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(3)
      GO TO 30
    7 IF (HH(1).GT.PCI) GO TO 13
      IF (YS(3).EQ.YS(2)) GO TO 10
      IF (XS(2).EQ.XS(3)) GO TO 9
      IF (YS(1).EQ.YS(2)) GO TO 8
      XX1=XS(1)
      YY1=YS(1)-YD*(PCD-HH(1))/(HH(2)-HH(1))
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(1)
      GO TO 30
    8 CONTINUE
      XX1=XS(1)
      YY1=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(2)-HH(1))
      YY2=YS(1)
      GO TO 30
    9 IF (YS(3).EQ.YS(1)) GO TO 11
      XX1=XS(3)+XD*(HH(2)-PCD)/(HH(2)-HH(1))
      YY1=YS(1)
      XX2=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      GO TO 30
   10 IF (XS(2).EQ.XS(1)) GO TO 12
      XX1=XS(1)+XD*(PCD-HH(1))/(HH(2)-HH(1))
      YY1=YS(2)-YD*(HH(2)-PCD)/(HH(2)-HH(1))
      XX2=XS(1)
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      GO TO 30
   11 CONTINUE
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY1=YS(1)
      XX2=XS(3)+XD*(HH(2)-PCD)/(HH(2)-HH(1))
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(2)-HH(1))
```

```
      GO TO 30
   12 CONTINUE
      XX1=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY1=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      XX2=XS(1)
      YY2=YS(2)-YD*(HH(2)-PCD)/(HH(2)-HH(1))
      GO TO 30
   13 A(I,J)=ZRO
      NP=0
      RETURN
   14 IF (HH(2).EQ.PCI) GO TO 15
      XX1=XS(3)
      YY1=YS(3)
      A(I,J)=ZRO
      NP=1
      RETURN
   15 XX1=XS(3)
      YY1=YS(3)
      XX2=XS(2)
      YY2=YS(2)
      A(I,J)=ZRO
      IF (HH(1).EQ.PCI) GO TO 16
      RETURN
   16 XX3=XS(1)
      YY3=YS(1)
      NP=3
      RETURN
   17 HH(1)=H(I+1,J+1)
      HH(2)=H(I,J+1)
      HH(3)=H(I+1,J)
      XS(1)=X1+I*XD
      XS(2)=X1+(I-1)*XD
      XS(3)=XS(1)
      YS(1)=YM-J*YD
      YS(2)=YS(1)
      YS(3)=YM-(J-1)*YD
      CALL HSORT (HH,XS,YS)
      IF (HH(3).GT.PCI) GO TO 18
      IF (HH(3).EQ.PCI) GO TO 14
      NP=0
      A(I,J)=MIN
      RETURN
   18 IF (HH(2).GT.PCI) GO TO 24
      IF (YS(2).EQ.YS(1)) GO TO 21
      IF (XS(2).EQ.XS(1)) GO TO 20
      IF (YS(3).EQ.YS(2)) GO TO 19
      XX1=XS(3)
      YY1=YS(2)-YD*(PCD-HH(2))/(HH(3)-HH(2))
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(3)
```

```
      GO TO 30
   19 CONTINUE
      XX1=XS(3)
      YY1=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(3)
      XX2=XS(2)+XD*(PCD-HH(2))/(HH(3)-HH(2))
      GO TO 30
   20 IF (YS(3).EQ.YS(2)) GO TO 23
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY1=YS(3)
      XX2=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(2))
      YY2=YS(2)-YD*(PCD-HH(2))/(HH(3)-HH(2))
      GO TO 30
   21 IF (XS(3).EQ.XS(2)) GO TO 22
      XX1=XS(3)
      YY1=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      XX2=XS(2)+XD*(PCD-HH(2))/(HH(3)-HH(2))
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(2))
      GO TO 30
   22 CONTINUE
      XX1=XS(3)
      YY1=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(2))
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      GO TO 30
   23 CONTINUE
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(2))
      YY1=YS(3)
      XX2=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      GO TO 30
   24 IF (HH(1).GT.PCI) GO TO 13
      IF (YS(2).EQ.YS(3)) GO TO 28
      IF (XS(2).EQ.XS(3)) GO TO 26
      IF (YS(1).EQ.YS(2)) GO TO 25
      XX1=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY1=YS(1)
      XX2=XS(1)
      YY2=YS(2)-YD*(HH(2)-PCD)/(HH(2)-HH(1))
      GO TO 30
   25 CONTINUE
      XX1=XS(2)+XD*(HH(2)-PCD)/(HH(2)-HH(1))
      YY1=YS(1)
      XX2=XS(1)
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      GO TO 30
   26 IF (YS(1).EQ.YS(2)) GO TO 27
      XX1=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY1=YS(1)
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(2)-HH(1))
```

```
      YY2=YS(2)-YD*(HH(2)-PCD)/(HH(2)-HH(1))
      GO TO 30
   27 CONTINUE
      XX1=XS(1)+XD*(PCD-HH(1))/(HH(2)-HH(1))
      YY1=YS(1)
      XX2=XS(1)+XD*(PCD-HH(1))/(HH(3)-HH(1))
      YY2=YS(3)-YD*(HH(3)-PCD)/(HH(3)-HH(1))
      GO TO 30
   28 IF (XS(3).EQ.XS(1)) GO TO 29
      XX1=XS(1)
      YY1=YS(1)-YD*(PCD-HH(1))/(HH(2)-HH(1))
      XX2=XS(3)+XD*(HH(3)-PCD)/(HH(3)-HH(1))
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      GO TO 30
   29 CONTINUE
      XX1=XS(1)
      YY1=YS(1)-YD*(PCD-HH(1))/(HH(3)-HH(1))
      XX2=XS(2)+XD*(HH(2)-PCD)/(HH(2)-HH(1))
      YY2=YS(1)-YD*(PCD-HH(1))/(HH(2)-HH(1))
      GO TO 30
   30 A(I,J)=ZRO
      RETURN
      END

      SUBROUTINE HSORT (HH,XS,YS)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 HH(3),XS(3),YS(3),TE
      ITE=0
    1 IF (HH(2).GE.HH(1)) GO TO 2
      TE=HH(2)
      HH(2)=HH(1)
      HH(1)=TE
      TE=XS(2)
      XS(2)=XS(1)
      XS(1)=TE
      TE=YS(2)
      YS(2)=YS(1)
      YS(1)=TE
    2 CONTINUE
      IF (ITE.EQ.1) GO TO 4
      IF (HH(3).GE.HH(2)) GO TO 3
      TE=HH(2)
      HH(2)=HH(3)
      HH(3)=TE
      TE=XS(2)
      XS(2)=XS(3)
      XS(3)=TE
      TE=YS(2)
      YS(2)=YS(3)
      YS(3)=TE
```

```
3 ITE=1
  GO TO 1
4 RETURN
  END

  SUBROUTINE FINDTB (K,M1,N1,A,I,J)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON LD,MD
  LOGICAL*1 A(LD,MD)
  LOGICAL*1 PL,MIN,ZRO
  REAL*8 HH(3),PH(3),H(101,51)
  COMMON /CON /X1,XN,Y1,YM,XD,YD,XX1,XX2,XX3,YY1,YY2,YY3,PCI,H,PL,
 1ZRO,MIN
  COMMON /IJ/I11,J11,I12,J12,II1,II2
  N2=N1-1
  M2=M1-1
  IF (K.EQ.2) GO TO 11
  I=1
  DO 5 J=J11,N2
  IF (A(I,J).NE.PL) GO TO 5
  HH(1)=H(I,J)
  HH(3)=H(I+1,J)
  HH(2)=H(I,J+1)
  PS=0
  DO 4 L=1,3
  PH(L)=0
  IF (HH(L)-PCI) 1,2,3
1 PH(L)=-1
  GO TO 4
2 J11=J
  GO TO 22
3 PH(L)=1
  PS=1
4 CONTINUE
  J11=J
  IF (PH(1)*PH(2).LT.0) RETURN
  IF (PS.EQ.0) A(I,J)=MIN
5 CONTINUE
  J=1
  DO 10 I=I11,M2
  IF (A(I,J).NE.PL) GO TO 10
  HH(1)=H(I,J)
  HH(2)=H(I+1,J)
  HH(3)=H(I,J+1)
  PS=0
  DO 9 L=1,3
  PH(L)=0
  IF (HH(L)-PCI) 6,7,8
6 PH(L)=-1
  GO TO 9
```

```
    7 I11=I
      GO TO 22
    8 PH(L)=1
      PS=1
    9 CONTINUE
      I11=I
      IF (PH(1)*PH(2).LT.0) RETURN
      IF (PS.EQ.0) A(I,J)=MIN
   10 CONTINUE
      I=0
      RETURN
   11 I=M2
      DO 16 J=J12,N2
      IF (A(I,J).NE.PL) GO TO 16
      HH(1)=H(I+1,J)
      HH(2)=H(I+1,J+1)
      HH(3)=H(I,J+1)
      PS=0
      DO 15 L=1,3
      PH(L)=0
      IF (HH(L)-PCI) 12,13,14
   12 PH(L)=-1
      GO TO 15
   13 J12=J
      GO TO 22
   14 PH(L)=1
      PS=1
   15 CONTINUE
      J12=J
      IF (PH(1)*PH(2).LT.0) RETURN
      IF (PS.EQ.0) A(I,J)=MIN
   16 CONTINUE
      J=N2
      DO 21 I=I12,M2
      IF (A(I,J).NE.PL) GO TO 21
      HH(1)=H(I,J+1)
      HH(2)=H(I+1,J+1)
      HH(3)=H(I+1,J)
      PS=0
      DO 20 L=1,3
      PH(L)=0
      IF (HH(L)-PCI) 17,18,19
   17 PH(L)=-1
      GO TO 20
   18 I12=I
      GO TO 22
   19 PH(L)=1
      PS=1
   20 CONTINUE
      I12=I
```

```
      IF (PH(1)*PH(2).LT.0) RETURN
      IF (PS.EQ.0) A(I,J)=MIN
   21 CONTINUE
      I=0
      RETURN
   22 WRITE (6,23)
   23 FORMAT (1X,'CONTOUR PASSES THROUGH THE VERTEX OF A TRIANGLE.')
      STOP 4
      END

      SUBROUTINE FINDTI (K,M1,N1,A,I,J)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 A(LD,MD)
      LOGICAL*1 PL,MIN,ZRO
      REAL*8 HH(3),H(101,51)
      COMMON LD,MD
      COMMON /CON /X1,XN,Y1,YM,XD,YD,XX1,XX2,XX3,YY1,YY2,YY3,PCI,H,PL,
     1ZRO,MIN
      COMMON /IJ/I11,J11,I12,J12,II1,II2
      N2=N1-1
      M2=M1-1
      IF (K.EQ.2) GO TO 6
      DO 5 I=II1,M2
      II1=I
      DO 5 J=1,N2
      IF (A(I,J).NE.PL) GO TO 5
      HH(1)=H(I,J)
      HH(3)=H(I+1,J)
      HH(2)=H(I,J+1)
      PS=0
      PR=0
      DO 4 L=1,3
      IF (HH(L)-PCI) 1,2,3
    1 PR=-1
      GO TO 4
    2 GO TO 12
    3 PS=1
    4 CONTINUE
      IF (PR*PS.LT.0) RETURN
      IF (PS.EQ.0) A(I,J)=MIN
    5 CONTINUE
      I=0
      RETURN
    6 CONTINUE
      DO 11 I=II2,M2
      II2=I
      DO 11 J=1,N2
      IF (A(I,J).NE.PL) GO TO 11
      HH(1)=H(I+1,J)
      HH(2)=H(I+1,J+1)
```

```
      HH(3)=H(I,J+1)
      PS=0
      PR=0
      DO 10 L=1,3
      IF (HH(L)-PCI) 7,8,9
    7 PR=-1
      GO TO 10
    8 GO TO 12
    9 PS=1
   10 CONTINUE
      IF (PR*PS.LT.0) RETURN
      IF (PS.EQ.0) A(I,J)=MIN
   11 CONTINUE
      I=0
      RETURN
   12 WRITE (6,13)
   13 FORMAT (1X,'CONTOUR PASSES THROUGH THE VERTEX OF A TRIANGLE.')
      STOP 4
      END
```

E. DECK SETUP

This program can be executed on the U.S.G.S. IBM 360/65 computer using the following arrangement of JCL cards, program source deck, object deck of CORE, and data cards.

```
//  JOB CARD (SEE USGS JOB CARD SPECS.)
/*SETUP     NNNNNN/9R
//  EXEC FORTHCLG,PARM.FORT='NOMAP',REGION.FORT=252K,
//   REGION.GO=252K,TIME.GO=M
//FORT.SYSIN DD *

     PROGRAM SOURCE DECK

/*
//LKED.SYSLIB DD DISP=SHR,DSN=SYS1.FORTLIB
//      DD DISP=SHR,DSN=SYS1.FLATBEDC
//     DD DISP=SHR,DSN=XTENT.LIB
//     DD DISP=SHR,DSN=SSP.LIB2
//LKED.SYSIN DD *

     OBJECT DECK OF CORE SUBROUTINE

/*
//GO.FT01F001 DD UNIT=SYSDK,DISP=(NEW,DELETE,DELETE),DSN=&&TEMP,
//   DCB=(RECFM=VBS,BLKSIZE=7294,LRECL=20008),SPACE=(TRK,(10,2))
//GO.FT02F001 DD DUMMY
//GO.FT09F001 DD DSN=MTF1,UNIT=TAPE9,DISP=(,KEEP),VOL=SER=NNNNNN,
//  DCB=(RECFM=VS,LRECL=484,BLKSIZE=488,DEN=2)
//GO.FT10F001 DD *


     Z VARIABLE INPUT MATRIX DECK

/*
//GO.SYSIN DD *

     PROGRAM CONTROL CARDS (2)
     MAP TITLE CARDS (3)
     VARIABLE FORMAT CARD FOR Z VARIABLE INPUT MATRIX
     ACIGP CARD
     SPECIAL LINE DATA SETS

/*
//
```

If the OPTIONS INDICATOR in column 55 of Control Card Number 2 is equal to 0-4, then the variable format card for the input matrix is required. If the OPTIONS INDICATOR is equal to 0, 3 or 4, then the ACIP card is also required. If the OPTIONS INDICATOR ≠ 0-4, then these two cards must be omitted from the GO.SYSIN data set. The special line data sets are optional for all values of the OPTIONS INDICATOR.

The maximum CPU time for the GO step (M) depends on the size of the Z variable input data matrix and a number of other variables and options chosen by the user. For most applications M can be safely chosen as 5 minutes. NNNNNN is the volume serial number of the plottape. The FT02F001 DD card is used to specify a data set for recording the values of the gradients'magnitudes followed by their associated angles. If the user does not wish to store these values for later use, he should use a //GO.FT02F001 DD DUMMY card as shown above. The values of the magnitudes are written on the FT02F001 data set using the format statement (5E16.8) row by row in the same order as they are printed in the output. Their associated angles are written on this data set using the format statement (8F10.5) row by row in the same order as they are printed in the output. The FT10F001 DD statement must describe the data set containing the Z variable input matrix.

Be sure to include the volume serial numbers of any additional tapes or discs referred to on these DD cards on the SETUP card if they are not permanently mounted volumes.