<table>
<thead>
<tr>
<th>Title</th>
<th>FORTRAN Program of Preparing Contour Maps for Geologic Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Yamamoto, Kaichiro; Nishiwaki, Niichi</td>
</tr>
<tr>
<td>Citation</td>
<td>Memoirs of the Faculty of Science, Kyoto University. Series of geology and mineralogy (1975), 41(1): 1-34</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1975-01-31</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/186601">http://hdl.handle.net/2433/186601</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

Kyoto University
FORTRAN Program of Preparing Contour Maps
for Geologic Use

by

Kaichiro YAMAMOTO and Niichi NISHIWAKI

(Received May 17, 1974)

Abstract

A computer program was designed for preparing contour maps by the "Polyhedron Method". The program was written in FORTRAN for FACOM 230-60/75 by utilizing CALCOMP X-Y plotter Model 770/763. It can be easily modified for other computers which have more than 41 K words (or 164 K bites) of core memory.

The procedure of automatic contouring and the operating instructions of the program are described, and several test examples for geologic use are presented. The source list of the program is also carried in the appendix.

Introduction

A contour map is one of the most common ways of displaying geological quantitative areal data. Many mapping procedures (Bishop, 1960 for example) and their applications have been developed. They are isopachous maps (Merriam, 1955; Krumbein, 1962; Kanto Loam Research Group, 1965), isolith maps (Krumbein, 1962) and trend-surface maps (Krumbein, 1962; Merriam and Harbough, 1964; Schramm, 1968) in stratigraphy, and structure contour maps (Merriam, 1955; Kakimi et al., 1973; Robinson and Charlesworth, 1969) and beta diagrams (Robinson, 1963; Noble and Eberly, 1964) in structural geology, for example. Besides, contour maps are generally used in display of many geophysical data, e.g. magnetic and gravitational ones.

However, it consumes time and cost to prepare contour maps by hand method. The quality of contour maps, when they are prepared by hand method, depends on an operator's technique and on his interpretation of data to be mapped. For the reasons mentioned above, most maps have been prepared only for data required specially to be displayed in contour maps, and they are nothing more than the illustrative maps. Consequently, not a few informations from collected data have been left to be used.

The computer has enabled to prepare standardized contour maps inexpensively and promptly, and the several procedures have been developed for computer contouring (Harbough and Merriam, 1968, p. 32; Cottafava and Mori, 1969;
There are two kinds of output media for contour maps which can be prepared by a computer; the one is a lineprinter and the other an X-Y plotter or drafter. A contour map made by a lineprinter is shown as a characters pattern in which the same characters are printed on the places where the values fall within the same ranges (YAMAMOTO, 1973). Therefore, it requires either raw or processed data regularly and densely spaced. Although it easily and promptly makes a contour map, it cannot make any accurate one. It may be suitable for mapping functional surfaces. On the other hand, an X-Y plotter or a drafter has an advantage in that they can make much more precise and detailed ones. Besides, it can make a contour map directly even from irregularly spaced data.

The present program was designed for plotting a contour map by using an X-Y plotter in order to make the map from data irregularly spaced as well as from regularly spaced ones. It includes many options for geologic use.

The basic principle used in the program is the "Polyhedron method" described by HABBOUGH and MERRIAM (1968). The permission for using the principle is given from one of the authors (D. F. MERRIAM).

The program was made as one of the developing programs of Data Processing Center, Kyoto University (PROBLEM NO. 5001EY044 and 5001DY045). Any one who uses the program is required to have the permission from the present authors or Data Processing Center, Kyoto University.

Acknowledgement

The authors wish to thank Prof. Dr. Keiji Nakazawa of this Institute for encouraging them during the study. They are indebted to Prof. Dr. Daniel F. Merriam of Syracuse University for readily permitting them to use the contouring principle. The manuscript has benefited from the critical review of Dr. Shinjiro Mizutani of Nagoya University, whom they thank for his comments. The authors also wish to thank Mr. Kenichi Harada of this Institute for his help to prepare the manuscript, and the staffs of Data Processing Center, Kyoto University for their help in programming and computation.

General Description of Program

This program produces a contour map from regularly or irregularly spaced data by using an X-Y plotter. The arrangement of a map on a plotting paper is shown in Fig. 1. This program is written in FACOM 230-60/75 FORTRAN (Fujitsu, 1970) which corresponds to IBM 360 FORTRAN IV (GERMAIN, 1967) using FACOM 230-60/75 SSL (Scientific Subroutine Library; Fujitsu, 1972) and CALCOMP routines (Yoshizawa Business Machines, 1969a and 1969b), and requires about
41K words (or 164 K bites) of core memory and 75 cm (29.5 inches) plotter. Many options are provided for geologic uses; input data selection, insertion of geographic data as a referring map and so forth (see b. Input options and c. Processing options).

a. Contouring procedure

By assuming that a surface to be mapped is represented as a polyhedron surface of triangular elements (named faces) each of which is defined by its peak points in a three dimensional space (Fig. 2), contour lines in a triangular element can be

Fig. 1. Arrangement of output map on the plotting sheet. The contour map is drawn in the frame of broken line. The unit of length is expressed in cm.

Fig. 2. Tetrahedron model illustrating the contouring procedure. The tetrahedron is constructed by triangular faces. Broken lines are contours.
obtained by interpolation as shown in Fig. 3 (Harbough and Merriam, 1968, p. 34). Conjunction of contour lines for all the elements yields a contour map of the surface to be mapped.

In this program, faces are automatically defined even if face definition is not directly given as input.

b. Input options

The following three kinds of data can be read as input:

(1) Irregularly spaced data with face definition
(2) Irregularly spaced data without face definition

A specified area to be mapped is derived into rectangular grid units, and the units are sequentially numbered. The value at a given grid point (named the grid value) is computed by approximation of the observations in the specified

Fig. 3. Triangular face and interpolation for contouring. Slender lines are contours.

Fig. 4. Mapping area, grids, and grid units. Observations in hatched area are used to estimate the grid value at point P. The area should be specified as input using the number of units. In this case, the number is 2.
number of units around the grid point (Fig. 4). As a result, regularly spaced data are completed (see to Harbough and Merriam, 1968, p. 35). (The more number of units are specified as the area in which observations are, the map will be more smoothed.) Then each unit is subdivided into two triangular elements (faces), and the elements are automatically defined by using grid point numbers given before.

(3) Regularly spaced data
This kind of data should be grid data regularly spaced as mentioned above. The units formed by grids are subdivided into triangular elements, and they are automatically defined as in (2).

c. Processing options
The following three kinds of processings can be optionally performed:

(1) Insertion of referring point(s), line(s) and/or map scale.
(2) Plotting of data points with their numbers and values.
(3) Specification of blank unit(s), i.e., rectangular one(s), in which no contours are to be drawn; valid in the input cases of b–(2) and b–(3).

d. Input medium
Cards and an alternative tape can be used an input medium for source data input.

e. Limitations

(1) # of peaks, NOP ≤ 1000.
(2) # of faces, NOF ≤ 2000.
(3) # of rectangular units, NNN ≤ 1000.
(4) # of referring points, NUMBER ≤ 100.
(5) # of referring lines, NLINE ≤ 10.
(6) # of control points on a referring line, 3 ≤ NPFOLN ≤ 100
(7) Others: refer to Input Instructions.
(8) Map size: width (x–direction) and length (y–direction) are less than 68 cm and 88 cm respectively in the case of using Data Processing center, Kyoto University (refer to Fig. 1 for the map arrangement and to Fig. 7 for the coordinate system).

f. Output

(1) List of processing specifications
(2) Input
(3) Tracing informations of processing
(4) Error messages
(5) Contour map

g. Error treatment
Error checks are carried out on the following items, and error messages are printed out, if any errors are detected:
(1) Kind of input data:
If any code other than FDEF, SMTH, REGS are detected, the processing will be stopped with a message “ILLEGAL DATA KIND” (refer to Input Instructions-B-c).

(2) Computed results of grid values:
If a grid value is not normally obtained, a message “GAUELS ERROR, APPROXIMATING PLANE WAS NOT DETERMINED” will be printed out, and any contours will not be drawn in all the units concerned with the grid point.

(3) Specification of blank unit(s):
If any illegal specification is detected, the processing is stopped with a message “BLANK AREAS DEFINITION ERROR” (refer to Input Instructions-B-(g.)).

(4) Repetition times number of main repeating operations:
Amount checks are carried out on the items in Tab. 1. If an amount exceeds its limitation, it will be printed out in the form “**ERROR** CONTROL VALUE IS ILLEGAL.........” and the processing will be stopped.

(5) # of contour lines:
If contour lines are to be too densely drawn in a triangular element, the lines which exceed the limitation (100 lines) will not be drawn, and a message “LINES TO BE DRAWN OVER 100” will be printed out.

Processing Procedure
The processing is performed according to the following flow of steps (Fig. 5, Process flow chart).

Step 1. Data and task specifications are read as input from cards.

The program control proceeds to step 2, 3, or 4 according to the kind of source data; to Step 2, when they consist of peak and face definition data, to Step 3, when only peak ones, and to Step 4, when regularly

<table>
<thead>
<tr>
<th>#</th>
<th>Code</th>
<th>Limitation</th>
<th>Test Matter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAP$</td>
<td>10</td>
<td># of sets of regularly spaced data in a tape</td>
</tr>
<tr>
<td>2</td>
<td>MAXX</td>
<td>200</td>
<td># of grids in x-direction</td>
</tr>
<tr>
<td>3</td>
<td>MAXY</td>
<td>200</td>
<td># of grids in y-direction</td>
</tr>
<tr>
<td>4</td>
<td>NOP</td>
<td>1000</td>
<td># of peaks</td>
</tr>
<tr>
<td>5</td>
<td>NOF</td>
<td>2000</td>
<td># of faces</td>
</tr>
<tr>
<td>11</td>
<td>RPO$</td>
<td>100</td>
<td># of referring points</td>
</tr>
<tr>
<td>12</td>
<td>PLN$</td>
<td>10</td>
<td># of referring lines</td>
</tr>
<tr>
<td>13</td>
<td>RLP$</td>
<td>100</td>
<td># of control points on a referring line</td>
</tr>
<tr>
<td>14</td>
<td>MARK</td>
<td>200</td>
<td># of marks on a map scale</td>
</tr>
</tbody>
</table>
START
read: work name, task & data specifications
YES specify input formats
NO read: input formats

regularly spaced data kind
read: data interval
read: regularly spaced data
compute: coordinates (x,y)

NO specify blank unit(s)
YES specify: blank unit(s)
define: triangular elements
draw: contour map

NO plot data points
YES
plot data points

NO insert referring map
YES draw: referring map

FINISH

Fig. 5. Process flow-chart of program.
Step 2. After input of peak and face data, it proceeds to Step 7.
Step 3. Peak data are read as input. After completion of grid values, it proceeds Step 5.
Step 4. Regularly spaced data are read as input.
Step 5. Triangle definition is performed.
Step 6. (Optional) Blank unit(s) is/are defined according to input from card(s), if required.
Step 7. Contour lines are drawn.
Step 8. (Optional) Data points and their numbers and values are plotted, if required.
Step 9. (Optional) A referring map is drawn, if required.

**Input Instructions**

A. Order of cards in deck

An example is shown in Fig. 6 in FACOM 230-60/75 cases. Cards indicated by items of letter enclosed in parentheses are optional.

a. System cards*

b. Problem/data name card
c. Task and data specification card–a
d.) Input format cards
e. Task and data specification card–b
f. Data input cards if data are recorded on cards.
(g.) Blank unit(s) specification card(s)
(h.) Referring map data input cards
i. System cards*

```plaintext
$NO
$KJOB
COND=500
$FORTLINK
program deck
$PLOTRUN MAX=100
data deck

[$FD F08, FILE=(OLD,CZA037.XXX), UNIT=DD0, VOL=(SPEC,PP5014)]
If all input data are to be read from cards, this system card is not required. The underlined
are the file name and volume name.

$POUT
$JEND
```

Fig. 6. Setup example of card deck. The statements with marks "$" indicate system cards. The
one in the bracket is necessary to define an input file which is alternative to input data cards.

* These cards are required to control a job. The forms depend on the convention of each computer
center. Consult to your computer center.
B. Card preparation
   b. Problem/data name card
      Col. 1–80 Alphanumeric problem/data name; characters only in 1–40 columns are plotted on the output map.
   c. Task and data specification card–a
      Col. 1–4 FDEF: if face definition data are to be read as input.
            SMTH: if only peak data are to be read as input.
            REGS: if regularly spaced data are to be read as input.
      10 Input device logical # for peak or regularly spaced data (1 to 4 and 8 available); if not specified, 5 (card reader) is used as a default value.
      15 Input device logical # for face definition data (1 to 4 and 8 available; must be not the same as that for peak data); default is 5 for a card reader.
      16–18 YES: if a referring map is to be plotted; otherwise leave blank.
      21–30 YES: if blank unit(s) specification is/are to be performed; otherwise leave blank.
      26–28 YES: if peak points are to be plotted; otherwise leave blank.
      31–35 Skip # for plotting peak points; when 1, all points will be plotted.
      36–40 # of digits of the decimal part to be plotted as peak values. If it is punched in a negative number, peak values are not plotted.
      41–50 Base value of contours; real with a decimal point.
      61–70 Scaling factor for plotting; if it is left blank, the map is automatically scaled.
      71–75 0: if only a map scale is to be plotted.
            1: if only referring point(s) is/are to be plotted.
            2: if only referring line(s) is/are to be plotted.
            3: if referring point(s) and line(s) are to be plotted.
            Note: if the map scale is also to be plotted in the case of 1, 2, 3, negative number should be punched, i.e. −1, −2, −3.
      76–78 YES: if input format(s) is/are to be specified; otherwise leave blank.
   (d.) Input format cards (Optional)
      If input format(s) for peak and/or face definition data is/are to be specified, both of two format cards for peak and face definition should be
prepared. If input of face definition data is unnecessary, leave the second card blank.

Card 1. For peak data input:

1. If regularly spaced data are to be read as input, this specifies the format of values (z) at regularly spaced points (i.e. at grid points). If not specified, the format (8(6X,E10.4)) is used as a default one.

2. If irregularly spaced data are to be read as input, this specifies the format of a peak #, its location (x,y) and value (z) (see Fig. 7 for the coordinate system). The default is (2(I5,3F10.0, 5X)) for two peaks in a card.

\[\begin{array}{c}
\text{Card 2. For face definition data input:}
\end{array}\]

If face definition data are to be read as input, this card should be punched; otherwise leave blank. In the case of to be read, this specifies the format for arbitrary # of face definitions, each of which consists of three peak numbers to define a triangle.

The default is (4(3I5, 5X)) for four faces in a card.

e, f. Task and data specification card–b, and data input cards.

All data input cards should be punched according to the formats specified on the input format cards or to the default formats. If source data are to be read from an alternative tape, they should be also written to the formats. In the case of using an alternative tape, task and data specification card–b should be prepared.

The following three types of data (see General Description of Program) can be chosen as the source data:

1. Peak and face definition data (The task and data specification card–b is Card 1).
Card 1. Col. 1–5 # of peaks  
6–10 # of faces

Card 2. Peak data: peak #, coordinate of location (x,y).

Card 3. Face definition data; see the explanation of (d) Card 2.

(2) Peak data without face definition (The task and data specification card-b is Card 1).

Card 1. Col. 1–5 # of peaks  
6–10 # of units for the computation of grid values (see General Description of Program); if autoextension is necessary, punch in a positive number, and if not, in a negative one.

11–20 Boundaries of the mapping area in the original coordinate system; x-under, upper, and
21–30 y-under, upper respectively; real with a decimal
31–40 point (see Fig. 4).
41–50 Side lengths of a unit in the original scale;
51–60 x and y directional ones, respectively (see Fig. 4)

(3) Regularly spaced data

If the data are to be read from an alternative tape, see Note.

Card 1. Col. 1–10 Pitch of data points in x-direction
11–20 Pitch of data points in y-direction
21–25 # of data points in x-direction
26–30 # of data points in y-direction

Card 2. Values (z): punched according to the specified or default format.

Note: If the source data are to be read from an alternative tape, an ID-card in which ID-code and # are written in the format (A4, I5) is required instead of Card 1 described above. A data set whose ID-code and # coincide with the ones of ID-card is read as input. The data in the tape should be written in the following forms:

Section 1. File code and # of data sets in the file should be written in the format (A4, I5).

Section 2. ID-code, # and pitch and #s of data points in x, y-directions are written in the format (A4, I5, 2F10.0, 2I5).

Section 3. Values (z): in the specified or default format. Sections 2 and 3 make a data set. As many sets as # of data sets
written in Section 1 should be stored in the tape.

(q.) Blank unit(s) specification card(s)

16 sets of data can be punched on a card at most. Each set of data consists of unit # (4 columns) and a delimiter (1 column); blank, comma, hyphen, or slash. If a blank or comma is punched as a delimiter, only the unit which corresponds to the # in the set is defined as a blank unit. If a hyphen is punched as a delimiter, all the units from the one of the #s punched in the set to the one of the #s punched in the next set. But this type of specification should be kept in the same card. The delimiter of slash indicates the end of the specification. An example is shown in Fig. 7.

(h.) Referring map data input cards

(1) Referring point(s) input cards
Card 1. Col. 1–5 # of referring point(s)
Card 2. Prepare one card for a referring point.
Col. 1–10 x-coordinates of a referring point; real with a decimal point
11–20 y-coordinates of a referring point; with a decimal point
21–30 Size of a referring point (cm); real with a decimal point
31–40 Size of referring point name (cm); real with a decimal point
41–60 Name of a referring point
61–70 Symbol code (numeric) of a referring point

(2) Referring line(s) input card
Card 1. Col. 1–5 # of referring lines
Card 2. Col. 1–5 # of control points on a line
6–15 Size of line name (cm); real with a decimal point
16–25 Inclination of the name (degrees anticlockwise) from y-direction; real with a decimal point
26–45 Line name
Card 3. Prepare as many as desired for a line. Four sets of coordinates (x,y), i.e. eight values, can be punched on a card at most; real with a decimal point.

Note: A set of a Card 2 and Card 3 as many as desired should
be prepared for a line. Prepare those sets as many as the # of referring lines specified in Card 1.

(3) Map scale input card

Col. 1-10 Length of the map scale (cm): actual length on the marks output map; real with a decimal point (see Fig. 8)
11-15 # of marks in the map scale: include both of side (see Fig. 8).
16-23 Actual distance of the map scale (alphanumeric): plotted on the right shoulder of the map scale (see Fig. 9).

Fig. 8. Example of blank unit specification data. This example specifies the units #1 and #5 to #20 to be blank units.

Fig. 9. Map scale to be plotted in the output map.

**Processing Examples**

A. Test examples

(1) Irregularly spaced data

The data are concerned with the water depth of the Pacific off Hachinohe, Northeastern Japan, and derived from Hydrographic Department, Maritime Safety Agency, Japan. Two kinds of contour maps were prepared; the one from both peak and face data, and the other only from peak data. The input and the output maps for the former are shown in Fig. 10, those for the latter in Fig. 11. The contour lines in Fig. 11 are more smoothed than those in Fig. 10. The contour values can be obtained from the peak values in the former case and from the grid values printed in the output list in the latter case.

(2) Regularly spaced data

Two kinds of processing were performed on the regularly spaced data; the one with and the other without the blank units specification. The input and the output maps are shown in Figs. 12 and 13. The contour values can be obtained from the grid values in the input.

B. Application

Isopacous map of the Imaichi Pumice Bed. Fig. 14a is by this program and Fig. 14b is in “The Kanto Loam”. The pumice bed was supplied from the mountain of Nantaisan, which is one of the Quarternary volcanos in the Kanto district, Japan. Data from Kanto Loam Research Group (1965).
Kaichiro Yamamoto and Niichi Nishiwaki

$KJOR 50eloWol.6,YAMAMoTO.KAI,500331

$PLTNKRUN MAx=1no .

$RFLECF SEA BOTTOM, NO. 1 (HACHINOHE).

258 448
eOl 4.7
OOI 7.4
nn5 le.5

233 250 231
200 201 207
258 233 234
221 222 234

4.5

25.4

11.6

SIRIYA ZAKI

HATO2

HATO3

HATO4

850KM

(a) Input data: title card, control cards, peak data (two sets in a card each of which consists of peak $\#$, coordinates and depth of the point), face definition data (four sets in a card each of which consists of three peak point $\#$s), and data for referring points, line and a map scale.

Fig. 10. Relief of sea bottom off Hachinohe, Northwestern Pacific. This contour map was prepared from peak and face definition data derived from Hydrographic Department, Maritime Safety Agency, Japan.
(b) Output map: shoreline, and data points with their $s$ and values (depth) are plotted; contour values can be obtained from peak values.

Fig. 10 (continued)
(a) Input data: the same that of Fig. 10 except for without face definition data.

Fig. 11. Relief of sea bottom off Hachinohe, Northwestern Pacific. This contour map was prepared only from peak data which are also used in Fig. 10.
(b) Output map: the grid data each of which was estimated using the peak data around the grid point by least squares method; blank areas are due to scarcity of the peak data around the grid points; shoreline and data point with their $\text{#}$s are plotted; contour values can be obtained from the grid values in the output list.

Fig. 11. (continued)
Fig. 12. Processing example of regularly spaced data with blank unit definitions. The data are arbitrarily prepared for the test processing.

(a) Input data: title card, control cards, grid data, and blank unit definition data.

(b) Output map: blank areas are due to the blank unit definitions; contour values can be obtained from the grid values in the input.
(a) Input data: the same that of Fig. 12 except for without the blank unit definition data.

(b) Output map.

Fig. 13. Processing example of regularly spaced data without blank unit definitions.
(a) Smoothed contour map drawn by the program: peak data, data for referring points and a map scale are used, all of which are from Kanto Loam Research Group (1965).

Fig. 14. Isopacous map of the Imaichi Pumice Bed (Late Quaternary), the Kanto district, Japan.
FORTRAN Program of Preparing Contour Maps for Geologic Use

Fig. 14. (continued) Both of the maps show the distinct trend that the thicker the bed the nearer to the mountain of Nantai-san. The latter is more generalized, while more detail changes are expressed in the former.

References


Schramm, M. W., Jr. (1968), Application of trend analysis to Pre-Morrow surface, southeastern


**Appendix: Computer Program**

This program is constructed in a simple structure, not overlayed. In this program, FACOM 230–60/75 SSL (Scientific Subroutine Library) and CALCOMP routines (basic and functional ones) are used. They are marked "*" and "**" respectively in the explanations below.

a. Call tree

Main- and sub-programs are connected with each other as shown in Appendix-fig. 1.

![Call tree](image)

Appendix-fig. 1. Call tree. The routine marked "*" and "**" are FACOM 230–60/75 SSL (Scientific Subroutine Library) routine and CALCOMP ones respectively.

b. Function of main- and sub-programs

1. **Main program**: reads control, source and blank unit specification data, computes grid values, and automatically defines triangular elements as well as the control of the processing flow.
2. **DRAW**: draws contour lines in each element.
3. **APPEND**: normally terminates the job, after plotting data points, if required.
4. **REFMAP**: reads referring map input data and draw a referring map.
5. **TEST**: checks the repetition times of the main repeating operations.
6. **ERROR**: detects errors.
7. **SCALP**: scales plotting data.
(8) **GAUELS**: solves a linear system by Gaussian elimination method (SSL).

(9) **PLOTS**: opens a file in which plotting data are to be stored (CALCOMP routine).

(10) **PLOT**: linearly removes a plot-pen, and in the case of CALL PLOT (0.0, 0.0, 999) close the file (CALCOMP routine).

(11) **SYMBOL**: plots symbol(s) (CALCOMP routine)

(12) **NUMBER**: plots a number (CALCOMP routine)

(13) **FLINE**: draws a smooth line through specified points (CALCOMP routine).

c. Common blocks

Relations among common blocks and main- and sub-programs are shown in Appendix tab. 1.

<table>
<thead>
<tr>
<th>MAIN &amp; SUBROUTINE</th>
<th>COMMON BLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>PNI (for peak # and value)</td>
</tr>
<tr>
<td></td>
<td>OPT1 (for face definition)</td>
</tr>
<tr>
<td></td>
<td>OPT2 (for blank unit definition)</td>
</tr>
<tr>
<td></td>
<td>PRB (for problem/data name)</td>
</tr>
<tr>
<td>DRAW</td>
<td>PNI, OPT1, OPT2, PRB, AREA (for mapping area specification)</td>
</tr>
<tr>
<td>REFMAP</td>
<td>AREA, PRINT (dummy)</td>
</tr>
<tr>
<td>APPEND</td>
<td>not used</td>
</tr>
<tr>
<td>TEST</td>
<td>not used</td>
</tr>
<tr>
<td>ERROR</td>
<td>not used</td>
</tr>
<tr>
<td>SCALP</td>
<td>not used</td>
</tr>
</tbody>
</table>

d. Program source list
C ***** CONTOUR MAP PROCESSING PROGRAM NAME CMPP *****

C COMMON /FN1/ X(1000), Y(1000), Z(1000), NPEK(1000)
C /OPT1/ NDFP(3,2000)
C /OPT2/ NVIS(2000)
C /OPT3/ NAME(20)
C /DIM/ NDFP(20),.X(1000), Y(1000), Z(1000), AA(3,4), WWK(3)
C /DATA/ NDFP(20),NAME(20), NVIS, NVIS, NPEK, NPEK, NVIS
C /JMP/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPFP/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF2000000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF20000000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
C /JMPF200000000000000000000000000000000000000/ 'illions, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi, Fi,
FORTRAN Program of Preparing Contour Maps for Geologic Use

GO TO 43
43 READ (IND,1013) PX,PY,MAXX,MAXY
1013 FORMAT (2F10.0,2X)

DO 60 I=1,NOMAP
IF (IND.EQ.99) GO TO 44
READ (IND,1005) EI,DI,PX,PY,MAXX,MAXY
1005 FORMAT (2F10.0,2F10.0)
NITEM = MAXX-MAXY
IF (IFSP .NE. 'YES') GO TO 45
READ (IND,1FMP) (EZ(J),J=1,NITEM)
GO TO 48
45 READ (IND,1006) (EZ(J),J=1,NITEM)
1006 FORMAT (2F10.0,2F10.0)
IF (IND.EQ.99) GO TO 49
DO 50 J=1,NITEM
X(J) = PX+FLOAT(J-MAXX(MAXX(MAXB)-1))
Y(J) = PY+FLOAT(J-MAXX)
50 CONTINUE
WRITE (6,2003) (XC(J),Y(J),Z(J),I,J,NITEM)
2003 FORMAT (2I10,2F10.4,2X)
CALL PLOT5
CALL DRAW(NITEM,BASE,CINT,SCALE1,NOP)
CALL APPEND(NOP,NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,SKIP,LDEGT,NEK,SHL)
WRITE (6,2004) (TM(J),J=1,NITEM)
2004 FORMAT (1H1,15X,3F12,.2H )
CONTINUE
STOP 'DATA SET WAS NOT FOUND IN THE FILE'
70 IF (IND.EQ.99) REIND IND
GO TO 200
C****** INPUT & PROCESSING OF FACE DEFINITION DATA ******
80 IF (KDATA .EQ. 'FDEF') GO TO 100
READ (9,1007) NOP,NOF,BOUND
1007 FORMAT (2I10,2F10.0)
IF (IND.EQ.99) REIND IND
IF (IND.EQ.99) REIND IND
READ (IND,1FMP) (NPX(J),J=1,NOF)
1008 FORMAT (2F10.0,2X)
READ (IND,1FMP) (NPF(J),J=1,NOP)
1009 FORMAT (4F10.0)
WRITE (6,2005) (NPX(J)+NPY(J)+NPZ(J),J=1,NOF)
WRITE (6,2005) (NPF(J)+NFD(J),J=1,NOP)
WRITE (6,2005) (NPX(J)+NPY(J)+NPZ(J),J=1,NOF)
WRITE (6,2005) (NPF(J)+NFD(J),J=1,NOP)
WRITE (6,2005) (NPX(J)+NPY(J)+NPZ(J),J=1,NOF)
WRITE (6,2005) (NPF(J)+NFD(J),J=1,NOP)
WRITE (6,2005) (NPX(J)+NPY(J)+NPZ(J),J=1,NOF)
WRITE (6,2005) (NPF(J)+NFD(J),J=1,NOP)
WRITE (6,2005) (NPX(J)+NPY(J)+NPZ(J),J=1,NOF)
WRITE (6,2005) (NPF(J)+NFD(J),J=1,NOP)
CALL PLOTS
CALL APPEND(NOP,NOPT1,NOPT2,NOPT3,NOPT4,NOPT5,SKIP,LDEGT,NEK,SHL)
WRITE (6,2005) (TM(J),J=1,NITEM)
C****** OUTPUT: PEAK DATA WITHOUT FACE DEFINITION ******
100 CONTINUE
STOP 'ILLEGAL DATA KIND'
READ (9,1010) NOP,NUM,BOUND,PX,PY
1010 FORMAT (2I10,2F10.0)
IF (IND.EQ.99) REIND IND
READ (IND,1FMP) (NPX(J),J=1,NOF)
1013 FORMAT (2I10,2F10.0)
IF (IND.EQ.99) REIND IND
READ (IND,1FMP) (NPX(J),J=1,NOF)
1016 FORMAT (2I10,2F10.0)
IF (IND.EQ.99) REIND IND
READ (IND,1FMP) (NPX(J),J=1,NOF)
1019 FORMAT (2I10,2F10.0)
IF (IND.EQ.99) REIND IND
READ (IND,1FMP) (NPX(J),J=1,NOF)
1022 FORMAT (2I10,2F10.0)
C ***** ESTIMATION OF GRID VALUES *****
MAXX = (BOUND(2)-BOUND(1))/PX + 1.0
MAXY = (BOUND(4)-BOUND(3))/PY + 1.0
II1 = 0
CALL TEST (4,NOP,1000,NOP)
CALL TEST (2,MAY,200,MAY)
CALL TEST (3,MAY,200,MAY)
DO 180 J=1,MAY
   TEMP = BOUND(J) + PY*FLOAT(I-1)
   DO 170 I=1,MAXX
      XTEMP = BOUND(I) + PX*FLOAT(J-1)
      X1 = XTEMP + PX*FLOAT(I-1)
      X2 = XTEMP + PX*FLOAT(I)
      Y1 = YTEMP + PY*FLOAT(J-1)
      Y2 = YTEMP + PY*FLOAT(J)
      DT 110 M=1,3
      DO 100 N=M,3
         AA(M,N) = AA(M,N) + XW(M,N)*XW(M,N)
      DO 110 M=1,3
      DT 100 M=1,3
      AA(N) = AA(N) + WW(N)*WW(N)
      AA(N) = AA(N) + WW(N)*WW(N)
   DT 110 CONTINUE
   NNP = NNP + 1
   IF (X1(L) .LT. XI OR, XI(L) .GT. X2) GO TO 140
   IF (Y1(L) .LT. YI OR, YI(L) .GT. Y2) GO TO 140
   NNP = NNP + 1
   WW(3) = 1.0
   WW(3) = YI(L)
   DO 130 M=1,3
   DO 120 M=1,3
      AA(M) = AA(M) + WW(M)*WW(M)
   AA(N) = AA(N) + WW(N)*WW(N)
   AA(N) = AA(N)
   DT 120 CONTINUE
   GO TO 170
   DT 130 CONTINUE
   IF (NNP .LT. 5) GO TO 150
   CALL GAUESS(AA,3,3,4,1,0E+00,ILL)
   IF (ILL .NE. 0) CALL ERRN(1)
   X(11) = XTEMP
   Y(11) = YTEMP
   Z(11) = AA(1,4) + XTEMP*XW(2,4) + YTEMP*XW(3,4)
   GO TO 170
   DT 150 CONTINUE
   IF (NNUNIT .LE. 0) GO TO 160

C ---- ESTIMATION OF GRID VALUES ----
MAXX = (BOUND(2)-BOUND(1))/PX + 1.0
MAXY = (BOUND(4)-BOUND(3))/PY + 1.0
II1 = 0
CALL TEST (4,NOP,1000,NOP)
CALL TEST (2,MAY,200,MAY)
CALL TEST (3,MAY,200,MAY)
DO 180 J=1,MAY
   TEMP = BOUND(J) + PY*FLOAT(I-1)
   DO 170 I=1,MAXX
      XTEMP = BOUND(I) + PX*FLOAT(J-1)
      X1 = XTEMP + PX*FLOAT(I-1)
      X2 = XTEMP + PX*FLOAT(I)
      Y1 = YTEMP + PY*FLOAT(J-1)
      Y2 = YTEMP + PY*FLOAT(J)
      DT 110 M=1,3
      DO 100 N=M,3
         AA(M,N) = AA(M,N) + XW(M,N)*XW(M,N)
      DO 110 M=1,3
      DT 100 M=1,3
      AA(N) = AA(N) + WW(N)*WW(N)
      AA(N) = AA(N) + WW(N)*WW(N)
   DT 110 CONTINUE
   NNP = NNP + 1
   IF (X1(L) .LT. XI OR, XI(L) .GT. X2) GO TO 140
   IF (Y1(L) .LT. YI OR, YI(L) .GT. Y2) GO TO 140
   NNP = NNP + 1
   WW(3) = 1.0
   WW(3) = YI(L)
   DO 130 M=1,3
   DO 120 M=1,3
      AA(M) = AA(M) + WW(M)*WW(M)
   AA(N) = AA(N) + WW(N)*WW(N)
   AA(N) = AA(N)
   DT 120 CONTINUE
   GO TO 170
   DT 130 CONTINUE
   IF (NNP .LT. 5) GO TO 150
   CALL GAUESS(AA,3,3,4,1,0E+00,ILL)
   IF (ILL .NE. 0) CALL ERRN(1)
   X(11) = XTEMP
   Y(11) = YTEMP
   Z(11) = AA(1,4) + XTEMP*XW(2,4) + YTEMP*XW(3,4)
   GO TO 170
   DT 150 CONTINUE
   IF (NNUNIT .LE. 0) GO TO 160

WRITE (6,2005) NOP,NNUNIT,BOUND,PX,PY,X(I),Y(I),Z(I),I=1,NOP
2005 FORMAT (1X,NOP,'INPUT FOR OPTION=1/'
   1 I1 'NUMBER OF OBSERVATIONS ...'//
   2 I1 'BOUNDARY ...'//
   3 I1 'BOUNDARY ...'//
   4 I1 'MAXX = BOUND(2)-BOUND(1)/PX + 1.0'
   5 I1 'MAXY = BOUND(4)-BOUND(3)/PY + 1.0'
   6 I1 'CALL TEST (4,NOP,1000,NOP)'
   7 I1 'CALL TEST (2,MAY,200,MAY)'
   8 I1 'CALL TEST (3,MAY,200,MAY)'
   9 I1 'DO 180 J=1,MAY'
   10 I1 'TEMP = BOUND(J) + PY*FLOAT(I-1)'
   11 I1 'DO 170 I=1,MAXX'
   12 I1 'XTEMP = BOUND(I) + PX*FLOAT(J-1)'
   13 I1 'X1 = XTEMP + PX*FLOAT(I-1)'
   14 I1 'X2 = XTEMP + PX*FLOAT(I)'
   15 I1 'Y1 = YTEMP + PY*FLOAT(J-1)'
   16 I1 'Y2 = YTEMP + PY*FLOAT(J)'
   17 I1 'DO 110 M=1,3'
   18 I1 'DO 100 N=M,3'
   19 I1 'AA(M,N) = AA(M,N) + XW(M,N)*XW(M,N)'
   20 I1 'DO 110 M=1,3'
   21 I1 'DO 100 M=1,3'
   22 I1 'AA(N) = AA(N) + WW(N)*WW(N)'
   23 I1 'AA(N) = AA(N) + WW(N)*WW(N)'
   24 I1 'DO 110 CONTINUE'
   25 I1 'GO TO 170'
   26 I1 'DO 130 M=1,3'
   27 I1 'DO 120 M=1,3'
   28 I1 'AA(M) = AA(M) + WW(M)*WW(M)'
   29 I1 'DO 130 M=1,3'
   30 I1 'DO 120 M=1,3'
   31 I1 'AA(N) = AA(N) + WW(N)*WW(N)'
   32 I1 'AA(N) = AA(N)'
   33 I1 'DO 120 CONTINUE'
   34 I1 'GO TO 170'
   35 I1 'IF (NNP .LT. 5) GO TO 150'
   36 I1 'CALL GAUESS(AA,3,3,4,1,0E+00,ILL)'
   37 I1 'IF (ILL .NE. 0) CALL ERRN(1)'
   38 I1 'X(11) = XTEMP'
   39 I1 'Y(11) = YTEMP'
   40 I1 'Z(11) = AA(1,4) + XTEMP*XW(2,4) + YTEMP*XW(3,4)'
   41 I1 'GO TO 170'
   42 I1 'IF (NNUNIT .LE. 0) GO TO 160
FORTRAN Program of Preparing Contour Maps for Geologic Use

27

NIN = NIN + 1
GO TO 105

160 X(1:1) = XTEMP
Y(1:1) = YTEMP
Z(1:1) = 1.0E20

270 CONTINUE
LS = MAXX*(J-1) + 1
LE = LS + MAXX - 1
WHITE(*,9001) J = (NSTR(I),I=1,MAXX)

1
WRITE (6,9002) (X(I),Y(I),Z(I),I=LS,LE)

9001 FORMAT(1H10X,SMOOTHED/2H OF UNITS & X,Y,Z/Y/
WRITE (6,9002) (X(I),Y(I),Z(I),I=LS,LE)

9002 FORMAT(1H5X,10E15.0)

180 CONTINUE
200 CONTINUE
210 CONTINUE
220 IF (CNCN(I) .EQ. 0) Go TO 250
230 IF (CNCI1(I) .EQ. 1) Go TO 250
240 IF (CNCN(I) .EQ. 1) Go TO 250
250 CONTINUE
NCPC = MAXX-MAXY
DO 255 J = 1,NCPC
DD 255 = J
DO 260 = J+1,NNN
JTEM = J + (I-1)/(MAXX-1)
JJJ = 2*J - 1
NDFD(1,JJJ) = JTEM
NDFD(2,JJJ) = JTEM + 1
NDFD(3,JJJ) = JTEM + MAXX
IF (Z(JTEM) .EQ. 1.0E20) NVIS(JJJ) = 'NO'

IF (Z(JTEM+1) .EQ. 1.0E20) NVIS(JJJ) = 'NO'
IF (Z(JTEM+MAXX) .EQ. 1.0E20) NVIS(JJJ) = 'NO'
JJJ = 2*J
NDFD(1,JJJ) = JTEM + 1
NDFD(2,JJJ) = JTEM + MAXX
NDFD(3,JJJ) = JTEM + MAXX
IF (Z(JTEM+MAXX+1) .EQ. 1.0E20) NVIS(JJJ) = 'NO'
IF (Z(JTEM+MAXX) .EQ. 1.0E20) NVIS(JJJ) = 'NO'

IF (Z(JTEM+MAXX)) .EQ. 1.0E20) NVIS(JJJ) = 'NO'

260 CONTINUE
CALL PLOT5
NCPC = 2NNN
CALL DRAW(NOP,BASE,CNT,SCALE1,NCPC)
DO 280 = 1,NCPC
X(I) = X(I) - BOUNDC1
Y(I) = Y(I) - BOUNDC3

280 CONTINUE
CALL APPEND(NOP1,NOP2,NOP3,NOP4,KSkip,LOG9T,NPEK,1
KREF = SCALE1)
END
C ***** CONTOUR DRAW NAME..DRAW *****
C
SUBROUTINE DRAN(NOF,NPEF,CINT,SCALEI,NOP)
COMMON /PNBF/ X(1000),Y(1000),Z(1000),NPEF(1000)
I = 1
INPUT/ NAME(I)
2 I = I + 1
5 NPEF = NPEF + 3
3 NPEF = NPEF + 10000
5 DIMENSION Z(NOP), X(NOP) + YT(10), YT(10)
C
K = 1
C ***** MAP LABEL PLOTTING *****
CALL SYMBOL (6,0:0.5,1.5:NAME,90,0,40)
CALL PLOT(10,0+5+++)
WRITE (6,2001)
2001 FORMAT (1H1)//H1,10X,‘CONTOUR DRAW START’/
CALL TEST (4,NOP,1000,’NOP’)
CALL TEST (5,NOP,2000,’NOP’)
C
C ***** SCALING OF DATA (X,Y,Z) *****
CALL SCALP(Y,65,0,NOP+2)
CALL SCALP(Y,50,0,NOP+1)
CALL SCALP(z,1200.0,NOP+1)
IF (SCALEI,NE,0.0) GO TO 20
SCALEI = X(NOP+2)
IF (Y(NOP+2) .LT. SCALEI) SCALEI = Y(NOP+2)
20 DO 30 I = 1,NOP
X(I) = (X(I) - X(NOP+1))/SCALEI
Y(I) = (Y(I) - Y(NOP+1))/SCALEI
30 CONTINUE
BOUND(I) = X(NOP+2)
BOUND(O) = Y(NOP+1)
C
C ***** CONTOUR START *****
DO 250 I = 1,NOP
IF (NVIS(I),EW,’NOP’) GO TO 250
DO 42 J = 1,100
DO 40 J = 1,NOP
40 IF (NPEF(J),EW,NPEF(N(J))) GO TO 41
41 NVIS(O) = J
42 CONTINUE
ZMIN = Z(J)
CONTINUE
B16 = Z(M)
DO 50 J = 13
IF (Z(M) .LE. B16) GO TO 50
ZMIN = Z(M)
2M(J) = Z(M)
Z(M) = B16
MTLM(M) = M
40 CONTINUE
IF (Z(M) .LE. ZMIN) GO TO 60
50 CONTINUE
C
FORTRAN Program of Preparing Contour Maps for Geologic Use

BIG = ZM(3)
ZM(3) = ZM(2)
ZM(2) = ZM(1)
MTEM = MO(3)
MO(3) = MO(2)
MO(2) = MTEM

60 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,GT , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

70 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,LE , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

80 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,LE , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

90 IF (MINT ,GT , 3) GO TO 110
IF (LRR ,LE , 3) GO TO 100
IF (CON ,LE , 3) GO TO 100
Z22 = ZM(1)
M1 = MO(1)
Z21 = ZM(2)
M2 = MO(2)
LRR = LRR + 1
GO TO 70
CONTINUE

100 Z21 = ZM(3)
Z22 = ZM(1)
M1 = MO(1)
M2 = MO(2)
LRR = LRR + 1
GO TO 70
CONTINUE

40 **** PLOT A LINE ****
110 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,LE , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

200 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,LE , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

WRITE (4,2004) MINT
200 WRITE (4,2003) 1 , (NPDF(K+1) ,K=1,3)+ZM:MO:IL
210 CONTINUE = BASE + INT((FLOAT(INT((ZM(3)+BASE)/CINT)))*1.0)
MINT = 100
DO 210 J = 1
IL = IL + 1
IF (CON ,LE , ZM(1)) GO TO 220
LRR = LRR + 1
MINT = 0
Z21 = ZM(3)
M1 = MO(3)
Z22 = ZM(2)
M2 = MO(2)
IF (CON ,LE , Z21 , AND , CON ,LE , Z22) GO TO 80
GO TO 90
CONTINUE

RETURN
END
C ***** PLOT REFERENCE POINTS/LINES, NAME...REFMAP ****
C
SUBROUTINE REFMAP(KIND,SCALE)
C
COMMON /PRINT/ MOUT
C /AREA/ BOUND(6)
DIMENSION XP(100), YP(100), NAMEP(100,5), SIZEP(100,2),
1
2
3
4
NAMEML(10,5), SIZEL(10,2),
5
6
NAMTEM(5), AX(102), YY(102), NVALUE(2)
C
IF (KIND .EQ. 0) GO TO 250
KXX = IABS(KIND)
GO TO (10,50,100,KXX)
C ***** INPUT/REFERENCE POINT(S) DATA *****
10 READ (5,1001) NUMBER
1001 FORMAT (Ik8) NUMBER
CALL TEST (11,NUMBER,100,'RODF')
DO 30 I=1,NUMBER
READ (5,1002) XP(I),YP(I),SIZEP(I,1),SIZEP(I,2),
5
6
1
(NAMEP(I,J),J=1,5),NPOM(I)
30 CONTINUE
1002 FORMAT (4F10.5,5A4(15))
WRITE (6,1003) NUMBER,XP(I),YP(I),SIZEP(I,1),SIZEP(I,2),
5
6
1
(NAMEP(I,J),J=1,5),NPOM(I)
CONTINUE
1003 FORMAT (Ik8) NUMBER
CALL TEST (11,NUMBER,100,'RODF')
DO 30 I=1,NUMBER
READ (5,1004) NPFOLN(I),SIZEL(I,1),SIZEL(I,2),
5
6
1
(NAMEL(I,J),J=1,5)
30 CONTINUE
1004 FORMAT (Ik8) NPFOLN(I)
CALL TEST (13,NPFOLN(I),100,'MPF')
READ (5,1005) XL(I),YL(I),J=1,NPFOLN(I)
CONTINUE
1005 FORMAT (Ik8) XL(I),YL(I)
WRITE (6,1006) NPFOLN(I),SIZEL(I,1),SIZEL(I,2),
5
6
1
(NAMEL(I,J),J=1,5)
CONTINUE
1006 FORMAT (Ik8) SIZEL(I,1),SIZEL(I,2)
WRITE (6,1007) NAMEML(I,J),J=1,5
CONTINUE
1007 FORMAT (Ik8) NAMEML(I,J)
WRITE (6,1008) X(102),Y(102),NVALUE(2)
CONTINUE
1008 FORMAT (Ik8) X(102),Y(102),NVALUE(2)
GO TO 100,200,100,KXX
FORTRAN Program of Preparing Contour Maps for Geologic Use

C **** PLOT, REFERING POINT(S) ****
100 DO 120 I=1,NUMBER
    NAMTEM(J) = NAMEPC(I+J)
    CONTINUE
    YTEM = (XP(I)-BOUND(1))/SCALE
    XTEM = (YP(I)-BOUND(2))/SCALE
    CALL SYMBOL(XTEM,YTEM,SIZEP(I),NFOM(I),90,O,SIZEP(I,2))
    120 CONTINUE
    WRITE(6,2005)
    IF (KKK .EQ. 1) GO TO 250
    200 CONTINUE
C **** PLOT, REFERING LINE(S) ****
   DO 240 J=1,NLINE
      DO 220 I=1,NPFOLN(J)
         YTEM = (XL(I,J)-BOUND(1))/SCALE
         XTEM = (YL(I,J)-BOUND(2))/SCALE
         CALL SYMBOL(XTEM,YTEM,SIZEL(I),NAMTEM(SIZEL(I,1))
         CALL SYMBOL(XTEM,YTEM,SIZEL(I,2),NAMTEM,SIZEL(I,2))
      220 CONTINUE
      WRITE(6,2006)
      240 CONTINUE
    WRITE(6,2007)
      RETURN
1006 FORMAT (F10.0,15.2A4)
    WRITE (6,2006) DIST,MARK,NVALUE
    2006 FORMAT (1H INPUT REFERENCE SCALE',/)
      IN 15.2A4, 'LENGTH # OF MARKS VALUE,15.2A4)
      WID = 0.030D10
      DL = WID/40.0
      PMA = 10.0
      PIC = DIST/FLOAT(MARK+1)
      XTEM = -2.0 - WID
      CALL TEST (14,MARK+200,'MARK')
      250 CONTINUE
    RETURN
1005 FORMAT (F10.0,15.2A4)
    WRITE (6,2005) 250
      RETURN
1006 FORMAT (F10.0,15.2A4)
    WRITE (6,2006) DIST,MARK,NVALUE
      2006 FORMAT (1H REFERENCE POINTS PLOTTED',/)
      2007 FORMAT (1H REFERING LINES PLOTTED',/)
    RETURN
C ***** DATA POINTS PLOTTING AND JOB TERMINATING ROUTINE: NAME... APPENDAP000010
SUBROUTINE APPEND(NOP,NL2,N3,N4,N5,N6,LIMPEK,KY,KZ,KNREF,SCALE) APDO0020
C
DIMENSION X(NOP),Y(NOP),Z(NOP),LPEK(NOP) APDO0040
IF (N2.NE. 'YES') GO TO 100
IF (X .EQ. 0) Y =
DO 50 J = L,NOP
CALL SYMB(L,J),X(J),Y(J),Z(J),K,J,LPEK,X,Y,Z,KNREF,SCALE
50 CONTINUE
WRITE(6,200)
200 FORMAT('DATA POINTS PLOTTED') APDO0050
C... REFERING MAP PLOTTING..... APDO0060
IF (N3 .EQ. 'YES') CALL REFMAP(KNREF,SCALE)
C ***** NORMALLY TERMINATE THE JOB... APDO0070
CALL PLOT(0,0,0,999) APDO0080
STOP 'NORMAL END OF JOB'
END APDO0090

C ***** TEST ROUTINE: NAME... TEST... TST0000
SUBROUTINE TEST(ALTEM,NV,NAME) TST00020
IF (NV .GT. 0) AND (NV .LE. LIMIT) RETURN TST00030
TST00040
ALTEM(1) = ALTEM(1) + VALUE TST00050
STOP 'TST00050'
END TST00060

C ***** ERROR ROUTINE: NAME... ERROR... ERR0000
SUBROUTINE ERROR(NV) ERR00080
CALL PLOT(0,0,0,999) ERR00090
GO TO (10,20) I
10 STOP 'GAUGES ERROR: APPROXIMATING PLANE WAS NOT DETERMINED'
ERR00050
20 STOP 'BLANK AREAS DEFINITION ERROR (-)'
ERR00070
END ERR00090

SUBROUTINE SCALP(A,MM,NM)
DIMENSION A(N)
BIG = A(1)
SMAL = A(1)
DO 20 I = 1,NM
IF (A(I) .GT. BIG) BIG = A(I)
IF (A(I) .LT. SMAL) SMAL = A(I)
20 CONTINUE
A(MM) = SMAL
A(MM) = 0
RETURN
END
Supplement figure. Output example of the revised program: map is framed and contour values are written at ends of every two lines.
Supplement

The program was revised. The revised one can frame the output map and write their values along the contours as shown in the supplement figure in which the values are written every two contours. (Supplement figure)