COMMONWEALTH OF VIRGINIA

DEPARTMENT OF CONSERVATION
AND ECONOMIC DEVELOPMENT

DIVISION OF MINERAL RESOURCES

A COMPUTER-PROGRAM SYSTEM TO GRID AND CONTOUR RANDOM DATA

STANLEY S. JOHNSON, CAROL L. HUXSAW,
AND DOROTHY R. THOMAS

INFORMATION CIRCULAR 15

VIRGINIA DIVISION OF MINERAL RESOURCES

James L. Calver
Commissioner of Mineral Resources and State Geologist

CHARLOTTESVILLE, VIRGINIA
1971

COMMONWEALTH OF VIRGINIA

DEPARTMENT OF CONSERVATION
AND ECONOMIC DEVELOPMENT

DIVISION OF MINERAL RESOURCES

A COMPUTER-PROGRAM SYSTEM TO GRID AND CONTOUR RANDOM DATA

STANLEY S. JOHNSON, CAROL L. HUXSAW,
AND DOROTHY R. THOMAS

INFORMATION CIRCULAR 15

VIRGINIA DIVISION OF MINERAL RESOURCES

James L. Calver
Commissioner of Mineral Resources and State Geologist

CHARLOTTESVILLE, VIRGINIA
1971

COMMONWEALTH OF VIRGINIA
DEPARTMENT OF PURCHASES AND SUPPLY
RICHMOND
1971

Portions of this publication may be quoted if credit is given to the Virginia Division of Mineral Resources. It is recommended that reference to this report be made in the following form:

Johnson, S. S., Huxsaw, C. L., and Thomas, D. R., 1971, A computer-program system to grid and contour random data: Virginia Division of Mineral Resources Information Circ. 15, 38 p.

DEPARTMENT OF CONSERVATION AND ECONOMIC DEVELOPMENT

Richmond, Virginia

MARVIN M. SUTHERLAND, *Director*
CHARLES A. CHRISTOPHERSEN, *Deputy Director*
A. S. RACHAL, JR., *Executive Assistant*

BOARD

ANDREW A. FARLEY, Danville, *Chairman*
WILLIAM H. KING, Burkeville, *Vice Chairman*
MAJOR T. BENTON, Suffolk
JOSEPH C. CARTER, JR., Richmond
ADOLF U. HONKALA, Richmond
CLAUDE A. JESSUP, Charlottesville
GEORGE C. MCGHEE, Middleburg
ROBERT PATTERSON, Charlottesville
COLLINS SNYDER, Accomac
WILLIAM H. STANHAGEN, Alexandria
JOHN S. THORNTON, Culpeper
FREDERICK W. WALKER, Roanoke

CONTENTS

PAGE

Introduction 1

REFORMAT program 2

GRID GENERATION program 4

CONTOUR program 10

Appendix I: REFORMAT program listing 17

Appendix II: GRID GENERATION program listing 19

Appendix III: CONTOUR program listing 23

ILLUSTRATIONS

FIGURE PAGE

1. Sample contour map utilizing actual field data 16

A COMPUTER-PROGRAM SYSTEM TO GRID AND CONTOUR RANDOM DATA

BY

STANLEY S. JOHNSON, CAROL L. HUXSAW[1], and DOROTHY R. THOMAS[1]

INTRODUCTION

The computer programs that are described in this report were developed by the Computer-Science Center, University of Virginia, for the Virginia Division of Mineral Resources for use in the preparation of geophysical data, which will aid in interpreting the geology and mineral resources of the State.

The contour map is an excellent way for visual presentation of data. The computer-derived contour map has advantages of rapid preparation, objectivity, and reproducibility. If contoured by hand, a map may be subjective in that it reflects the opinion or degree of experience of the person doing the contouring. It may be desirable to make minor revisions to a computer-contoured map where indicated by other geological or geophysical information.

The purpose of the enclosed programs is to utilize the card output from the Division's gravity reduction program, and output them in a form (REFORMAT) suitable for use by a grid-generation program for random data points (GRID GENERATION). After the data have been gridded, this output and some of the output from the REFORMAT program are used by the CONTOUR program.

The polyconic projection used by REFORMAT and the basic algorithm for the generation of the grid were contributed by the U. S. Geological Survey in Washington, D. C. Modifications were necessary to make the programs compatible with the Burroughs B5500 computer and the Calcomp 570 plotter. The assistance of Gerald I. Evenden, U. S. Geological Survey, Denver, Colorado, was invaluable in the development of the CONTOUR program.

---

[1] Computer programmers, Computer-Science Center, University of Virginia, Charlottesville, Virginia.

These programs were developed into a working system for the Burroughs B5500 computer by Carol L. Huxsaw and Dorothy R. Thomas at the Computer-Science Center, University of Virginia, Charlottesville, Virginia.

REFORMAT PROGRAM

The REFORMAT program (FORTRAN IV) utilizes card input, consisting of points (z-values) and latitude and longitude determinations. These values are output in a form suitable for use in the GRID GENERATION program. Utilizing a polyconic projection, the latitude and longitude of the data points are converted to distance, in meters, from a central meridian. The program also determines the minimum and maximum latitude and longitude, both in degrees and meters, for use in the succeeding programs. The average distance between data points is calculated to assist in determining the grid size. If requested, the original input data is written on the line printer. The points with longitude and latitude in meters from the central meridian are written on a tape and are used to fill the grid. These points are followed by the same points with longitude and latitude in meters from the minimum longitude and latitude and are used to plot the original points on the contour map.

Input data for REFORMAT:

1. Latitude and longitude of the central meridian in degrees, minutes, and seconds. These numbers should be right-justified in a field of 4 digits.

cc.	1-4	XDEG—latitude degrees.
	5-8	XMIN—latitude minutes.
	9-12	XSEC—latitude seconds.
	13-16	YDEG—longitude degrees.
	17-20	YMIN—longitude minutes.
	21-24	YSEC—longitude seconds.
	80	OPT—option for printing input data.
		0 do not print input data.

1 print only latitude, longitude, and z-value within the specified box (Map Area).

2 print entire card if latitude and longitude are within the specified box (Map Area).

2. Minimum and maximum latitude and longitude of the inner box. This card is *not* read when the option on the first card is 0. The box represents the actual area to be contoured although points outside this area are necessary for accurate contours at the edges. The degrees should be integers right-justified in a field of 4 digits. The minutes are right-justified in a 6-digit field containing 2 decimal places.

cc.	1-4	LATDMX—maximum latitude degrees.
	5-10	LATMX—maximum latitude minutes.
	11-14	LATDMN—minimum latitude degrees.
	15-20	LATMN—minimum latitude minutes.
	21-24	LGDMX—maximum longitude degrees.
	25-30	LGMX—maximum longitude minutes.
	31-34	LGDMN—minimum longitude degrees.
	35-40	LGMN—minimum longitude minutes.

3. The points (z-values) with their latitude and longitude in degrees, minutes, and hundredths of a minute. The degrees are integers rightjustified in a 4-digit field. The minutes are right-justified in a 6-digit field containing 2 decimal places and the z-value in a 7-digit field with 2 decimal places.

cc.	15-17	XDEG—latitude degrees.
	18-23	XMIN—latitude minutes.
	27-29	YDEG—longitude degrees.
	30-35	YMIN—longitude minutes.
	73-79	ZVAL—z-value.

Output data from REFORMAT:

1. Tape

   a. The number of original points (format I10). The minimum and maximum longitude and latitude in meters from the central meridian (format 4F14.4).

b. The points with their corresponding longitude (x) and latitude (y) in meters from the central meridian (format 3F20.4).

c. The points with their longitude (x) and latitude (y) in meters from the minimum longitude and latitude (format 3F20.4).

2. Line Printer

a. If requested, the input data which fall within the inner box (map) are listed.

b. The number of original points: used as input to the CONTOUR program.

c. The minimum and maximum latitude and longitude in meters from the central meridian.

d. The minimum and maximum latitude and longitude in degrees: used as input to the CONTOUR program.

e. The average distance between data points: used to calculate the size of the grid.

GRID GENERATION PROGRAM

The GRID GENERATION program (FORTRAN IV) computes grid interestion points for random (irregularly-spaced) data over a defined area of a stated dimension. The maximum size of the grid is 120 x 120 points though this can be changed depending on the capacity of the computer. The input variables, DELX and DELY, correspond to the meters between grid points and are used for spacing the original points on the grid in the x-direction and y-direction, respectively. The process of GRID GENERATION is a multi-pass method that uses a general weighted average approach.

The scale factors to be used in both the x-direction (XSIZE) and y-direction (YSIZE) in the CONTOUR program are computed from the scale desired (SCALE) which is in miles per inch. Once the grid has been computed and all missing points have been calculated, the grid is scaled and divided into blocks of 9 inches in the y-direction for contouring, with three grid rows overlapping from one block to the next (maps are contoured on a 12-inch plotter).

The algorithm used in generating the grid, as described by the U. S. Geological Survey in the documentation for Program No. W9322, consists of five steps:

1. The weight functions and weighted sums for each random data point are formed when X_k and Y_k are within the area specified. The weight functions for the four grid points surrounding X_k, Y_k are $W_p = 1/d_p^2$ ($p = 1, 2, 3, 4$) and d_p is the distance from X_k, Y_k to the pth grid point. If the distance is zero, then $W_p = -1$, noting a point that falls directly on a grid intersection. In this case, the given Z_k value is used rather than the weighted average. When $W_p \neq -1$ the following sums are computed:

Weighted Sum

$$W_{ij} = W_{ij} + W_1, \qquad p = 1$$

$$W_{ij+1} = W_{ij+1} + W_2 \qquad p = 2$$

$$W_{i+1j} = W_{i+1j} + W_3 \qquad p = 3$$

$$W_{i+1j+1} = W_{i+1j+1} + W_4 \qquad p = 4$$

Weighted Product Sum

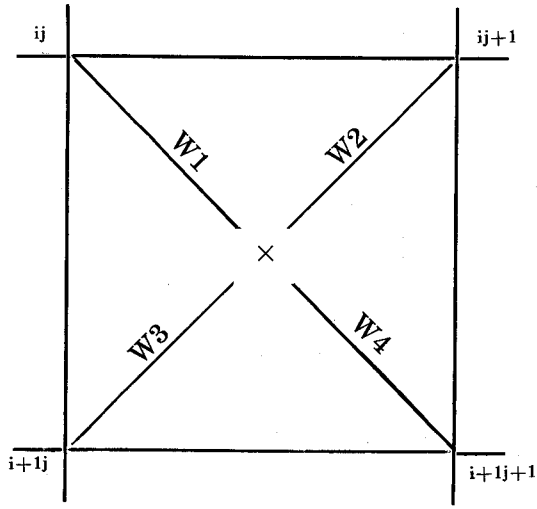
$$WZ_{ij} = WZ_{ij} + W_1 \times Z_k \qquad p = 1$$

$$WZ_{ij+1} = WZ_{ij+1} + W_2 \times Z_k \qquad p = 2$$

$$WZ_{i+1j} = WZ_{i+1j} + W_3 \times Z_k \qquad p = 3$$

$$WZ_{i+1j+1} = WZ_{i+1j+1} + W_4 \times Z_k \qquad p = 4$$

The indices i, j are the upper left corner of the grid box containing the point X_k, Y_k.



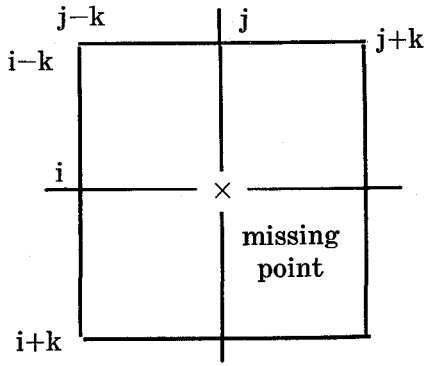
2. The weighted average for all points is computed where $W_{ij}>0$: $WZ_{ij} = WZ_{ij}/W_{ij}$ If $W_{ij} = 0$, then the grid point is missing and flagged as $WZ_{ij} = 10^{10}$.

3. The matrix WZ_{ij} is scanned for missing points. If any are found, then either step 4 or step 5 is executed, depending on option 4 of the input card, as described below. If there are no missing points, the GRID GENERATION process is complete.

4. If option 4 is zero, then at each missing point, a "K^{th} local grid model" is created with the missing point at its center. K is set to 1 for the first pass and incremented by 1 for each successive pass through this step. Step 3 is repeated after each pass. The grid model uses all possible values immediately surrounding the missing point and within the model

Note: Step 4 is generally recommended when large amounts of data are missing.

and boundary of the WZ_{ij} matrix. A weighted average is found within the grid similar to that outlined in the first two steps.



5. When option 4 is 1, a value is computed at each missing point using a four point boundary average method. The four neighboring points (i, j − k), (i − k, j), (i, j + k), (i + k, j) are summed if they are not themselves missing points and divided by the number of points used in the sum.

Input to GRID GENERATION:

1. Card

cc.	1 OPT(1)	1 if grid should be written on the line printer. 0 if grid should not be printed.
	2 OPT(2)	0 if output should be on tape. 1 if output should be on cards.
	3 OPT(3)	1 to list random data read from tape. 0 if data should not be printed.
	4 OPT(4)	0 to determine missing points by finding K local points for an average (Step 4). 1 to use the four point boundary average method to find missing points (Step 5).
	5-7 M	The number of rows of the grid (latitude distance). This is determined by dividing the size of the grid in the y-di-

rection (explained below) into the difference between the maximum and minimum latitude in meters. One (1) should be added to the rounded result to give the number of rows rather than the number of grid boxes.

8-10 N — The number of columns of the grid (longitude distance). This is determined by dividing the size of the grid in the x-direction (explained below) into the difference between the maximum and minimum longitude in meters. One (1) should be added to the rounded result to give the number of columns rather than the number of grid boxes.

11-14 DELY — Integer: the size of a grid box in the y-direction. From the output of the REFORMAT program, use the D (distance between data points) that has been computed and find the grid interval such that it is between 1/4D and 1/2D.

15-18 DELX — Integer: the size of the grid box in the x-direction. This is computed in the same manner as DELY.

19-23 SCALE — This is the scale desired on the contour map: the number of miles per inch. This is a 5-digit number including 2 decimal places.

2. Tape

a. NP — Number of original points (format I10).

XMIN — Minimum longitude in meters (format F14.4).

XMAX — Maximum longitude in meters (format F14.4).

YMIN — Minimum latitude in meters (format F14.4).

YMAX — Maximum latitude in meters (format F14.4).

b. The original points with their longitude (x) and latitude (y) in meters from the central meridian (format 3F20.4).

c. The original points with their longitude (x) and latitude (y) in meters from the minimum longitude and latitude (format 3F20.4).

Output from GRID GENERATION:

1. Tape

If OPT (2) of the input card is zero, then the grid and the original points will be written on a magnetic tape to be read into the CONTOUR program. The blocks are written in order and each block is preceded by the original points in that block. The original points are in the format 3F14.4 with the longitude (x) first, followed by the latitude (y), and then the z-value. The longitude and latitude have been converted to inches using the scale desired, so that they may be plotted directly on the map. The grid is written using the format 8F10.4 with each row beginning on a new record.

2. Cards

a. ZDATA Cards—a comma follows every value, including the last one.

The ZDATA card is punched whether the grid is written on tape or punched on cards. It contains the word "ZDATA" (including the quote marks), the number of rows and columns in the block following it, the number of original points in that block, and an indicator to tell the CONTOUR program if the grid is coming from cards or tape. There is one ZDATA card per block and they must be used as input to the CONTOUR program.

b. If OPT(2) of the input card is 1, then the grid and the original points will be punched onto cards, block by block, in the same manner as described for the tape.

3. Line Printer

a. The x, y, and z values used as input to the GRID GENERATION are printed if OPT(3) of the input card is 1.

b. Input information:

The 4 input options, the number of rows and columns, and the minimum and maximum longitude and latitude in meters from the central meridian are printed.

c. The number of blocks to be contoured and the number of rows in each block.

d. The scale factors for the x-direction (XSIZE) and y-direction (YSIZE). These must be entered in the CONTOUR program.

e. The grid is printed, row by row, if OPT(1) of the input card is 1.

f. The maximum and minimum z-values.

CONTOUR PROGRAM

The CONTOUR program (ALGOL) uses the output from GRID GENERATION and some of the output from REFORMAT to draw contour lines. An approximate bi-cubic spline technique is used to connect each contour in going from one grid box to the next. This technique uses a 16-point Lagrange interpolation scheme over the surface. The spline utilizes the derivative of the surface to assure a smooth x-y profile across each grid box boundary. For this reason, the CONTOUR program assumes data to be a uniformly spaced grid.

A polyconic projection was used by REFORMAT to compensate for the curvature of the earth. Since the curvature is not constant, the point with minimum longitude in degrees and the point with minimum longitude in meters from the central meridian are not always the same. Since the latter is used in positioning the original points on the map and the former is used in the positioning of the longitude, there may be a slight discrepancy between the actual longitude of some points and the longitude at which they appear on the map. This discrepancy is also influenced by the fact that the curvature is not accounted for in labeling the longitude or latitude. This slight discrepancy increases with the distance from the central meridian being used. This factor may also be the cause of a few points not falling precisely within the correct contour level. In most cases, this discrepancy will be slight. This same discrepancy can also be expected in the latitude.

The CONTOUR program is designed to drive a Calcomp 570 offline plotter. The plotter package itself declares and opens the plotter tape. All calls on the package are standard to Calcomp.

The CONTOUR program will do two types of contouring. The first type will contour an area from a given minimum to a given maximum contour level with a given incremental contour interval value. The second type will contour only the contour levels stated.

The latitude will be drawn and labeled on both sides of all sections or blocks of the map. The number 1849.8 used on cards 316 and 320 of the CONTOUR source program is meters per minute of latitude for central Virginia and should be changed for other regions. Longitude will be drawn and labeled on the top and bottom of the map, not on each block. The number 1477.8 used on cards 238 and 254 is meters per minute of longitude for central Virginia. The maximum and minimum values of latitude and longitude are part of the input, as well as the division in minutes for labeling. This program also draws a box (rectangular in shape) around an inner area of the map corresponding to specified minimum and maximum latitude and longitude. This must be an integer multiple of the divisions being used for labeling. That is, the latitude and longitude of the inner area must correspond to one of the latitude or longitude labels.

Input to CONTOUR:

All input is free-field. This requires a comma after each value, including the last value.

1. Card 1—the "MAP" card.

1. TYPE	"MAP" (including quotes) are the first 5 characters.
2. JSIZE	number of rows (y-direction) of entire map (integer).
3. ISIZE	number of columns (x-direction) of entire map (integer).
4. NUMY	number of rows in a sub-grid. Each grid box is divided into a sub-grid for interpolating the curve within that grid box.

		Program will compute NUMY=YSIZE/INTVAL when zero is entered (integer).
5.	NUMX	number of columns in a sub-grid. Computed to be NUMX=XSIZE/INTVAL when zero is entered (integer).
6.	YSIZE	scale factor for y-direction (determined by GRID GENERATION).
7.	XSIZE	scale factor for x-direction (determined by GRID GENERATION).
8.	INTVAL	size of each sub-grid box. If entered as zero, it is set to 0.05.
9.	JSUB	number of rows in largest block (integer).
10.	ISUB	number of columns in largest block (integer).
11.	ADJUST	tolerance allowed between contour value and value of grid point. — Programs will compute these values when 0 is entered.
12.	ADJUST2	amount to shift contour line when tolerance is not met. — Programs will compute these values when 0 is entered.
13.	NSUB	number of blocks as determined by GRID GENERATION.
14.	DELY	size of a grid box in the y-direction as described in GRID GENERATION input.
15.	DELX	size of a grid box in the x-direction.

2. Card 2—The CONTOUR card.

1.	TYPE	"CONTR" (including quote marks) are first 7 characters.
2.	DELCV	contour increment. — 0 for type 2 contouring.
3.	MINCV	minimum contour level. — 0 for type 2 contouring.
4.	MAXCV	maximum contour level. — 0 for type 2 contouring.

5. PRIMC	prime contour value to be maintained for contour suppression (i.e., 10 for 10's, etc.). 1 for no contour suppression. *Note*: NOT FULLY DEBUGGED.
6. NCVAL	number of contour levels to be drawn for type 2 contouring; 0 for type 1 contouring.
7. LABL	option for labeling points. 0 to plot original points as +, no label. 1 to plot points and label them. 2 to read through points.
8. CPI	number of contours per inch, zero for type 2. contouring, (used for contour suppression).
9. NDEC	number of decimal places for contour labels (less than 5).
10. SCAL	final scale of map in miles/inch.
11. NORPT	number of original points (output from REFORMAT).

3. Card 3 (optional)

This card is used only when NCVAL (card 2) is greater than zero. It specifies the contour levels to be contoured, in ascending order, when type 2 contouring is being done, and must contain NCVAL contour levels in free-field.

4. Card 4—the LATITUDE card.

1. TYPE	"LAT" (including quote marks) are the first 5 characters.	
2. ZLTD	degrees	of minimum latitude of map.*
3. ZLTM	minutes	

4. ZLGD	degrees	of minimum longitude of map.*
5. ZLGM	minutes	
6. MAXLTD	degrees	of maximum latitude of map.*
7. MAXLTM	minutes	
8. MAXLGD	degrees	of maximum longitude of map.*
9. MAXLGM	minutes	
10. DLT	labeling increment in minutes for latitude.	
11. DLG	labeling increment in minutes for longitude.	
12. LATMAXD	degrees	maximum latitude of inner box (multiple of DLT).
13. LATMAXM	minutes	
14. LATMIND	degrees	minimum latitude of inner box (multiple of DLT).
15. LATMINM	minutes	
16. LONGMAXD	degrees	maximum longitude of inner box (multiple of DLT).
17. LONGMAXM	minutes	
18. LONGMIND	degrees	minimum longitude of inner box (multiple of DLG).
19. LONGMINM	minutes	

*For accurate labeling these should be latitudes or longitudes of extreme points on the map.

5. The remainder of the input will depend on the output of the GRID GENERATION program. If it has produced a deck for the grid then that deck is all that is required. If it has produced cards and tape, then the cards that were punched, one for each block, complete the input card file and the tape will also be required.

In the case where GRID GENERATION program has not been used to supply the grid, the remainder of the input must follow the description below. Data may come from cards alone or from cards and tape.

a. Cards—when cards are the only additional input, they must be in order by block.

   1. ZDATA card—a comma follows every value including the last.

      TYPE "ZDATA" (including quote marks) are the first 7 characters.

      JMAX number of rows in this block.

      IMAX number of columns in this block.

      ORPTS number of original points in this block.

      ZFILE 1 if original points and grid are coming from cards.

      2 from tape.

   2. Original Points—in 3F14.4 format. The longitude (x) comes first, followed by the latitude (y), and then the Z-value. The longitude and latitude must be already converted to inches using the scale desired, so that they may be plotted directly on the map.

   3. Grid Points—in 8F10.4 format.

b. Cards and Tape

   1. Cards—the ZDATA cards, one for each block, as described above.

   2. Tape—the blocks must be in order, with each block preceded by the original points in that block and in the formats specified above.

Output from CONTOUR:

The output from CONTOUR consists of a plotter tape and a NORMAL END OF JOB message. Should an error occur in the input data, an error message will be printed on the line printer and the job halted.

The following listing of the programs (REFORMAT, GRID GENERATION, CONTOUR) was furnished to the Division of Mineral Resources by the Computer-Science Center and any technical questions concerning the programs should be directed to the Center. Figure 1 is part of a gravity map contoured with actual field data utilizing the following program system.

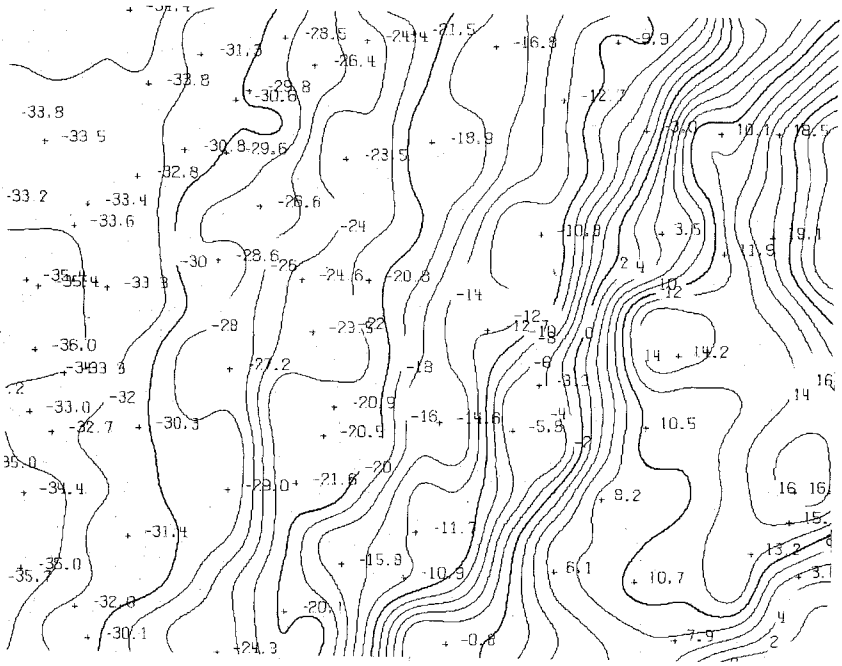


FIGURE 1. *Sample contour map utilizing actual field data.*

APPENDIX I: REFORMAT PROGRAM LISTING

```
FILE  5=CR,UNIT=READER,RECORD=10
FILE  6=LP,UNIT=PRINTER,RECORD=15
FILE  10=CONTOUR/INPUT,UNIT=TAPE,SAVE=633,BLOCKING=20,RECORD=10
C                                                                       RFMT0001
C     REFORMAT PROGRAM                                                  RFMT0002
C                                                                       RFMT0003
C                                                                       RFMT0004
C READ IN LATITUDE AND LONGITUDE OF CENTRAL MERIDIAN                    RFMT0005
C                                                                       RFMT0006
      INTEGER OPT                                                       RFMT0007
      REAL LATMX,LATMN,LGMX,LGMN                                        RFMT0008
      DIMENSION XLAT(1500),YLONG(1500),X(1500),Y(1500),Z(1500)          RFMT0009
      READ(5,10) XDEG,XMIN,XSEC,YDEG,YMIN,YSEC,OPT                      RFMT0010
 10   FORMAT(6I4,55X,I1)                                                RFMT0011
      CMLAT=SECOND(XDEG,XMIN)+XSEC                                      RFMT0012
      CMLONG=SECOND(YDEG,YMIN)+YSEC                                     RFMT0013
      IF (OPT .EQ. 0) GO TO 17                                          RFMT0014
      READ(5,15) LATUMX,LATMX,LATDMN,LATMN,LGDMX,LGMX,LGDMN,LGMN        RFMT0015
 15   FORMAT(4(I4,F6.2))                                                RFMT0016
      WRITE(6,30)                                                       RFMT0017
 30   FORMAT(1H1,29X,10HINPUT DATA//)                                   RFMT0018
C                                                                       RFMT0019
C READ IN LATITUDE AND LONGITUDE OF EACH POINT AND DO POLYCONIC         RFMT0020
C PROJECTION                                                            RFMT0021
C                                                                       RFMT0022
 17   I=0                                                               RFMT0023
 1000 READ(5,20,END=999) X1,X2,X3,XDEG,XMIN,X4,YDEG,YMIN,X5,X6,X7,X8,X9,RFMT0024
     1                   X10,X11,ZVAL                                   RFMT0025
 20   FORMAT(2A6,A2,I3,F6.2,A3,I3,F6.2,6A6,A1,F7.2)                     RFMT0026
      IF (OPT .EQ. 0) GO TO 24                                          RFMT0027
      IF ((XDEG .LT. LATDMX .OR. XDEG .EQ. LATDMX .AND. XMIN .LE. LATMX)RFMT0028
     1    .AND. (XDEG .GT. LATDMN .OR. XDEG .EQ. LATDMN .AND. XMIN .GE. RFMT0029
     2    LATMN) .AND. (YDEG .LT. LGDMX .OR. YDEG .EQ. LGDMX .AND.      RFMT0030
     3    YMIN .GE. LGMX) .AND. (YDEG .GT. LGDMN .OR. YDEG .EQ. LGDMN   RFMT0031
     4    .AND. YMIN .LE. LGMN)) GO TO 21                               RFMT0032
      GO TO 24                                                          RFMT0033
 21   IF (OPT .EQ. 1) WRITE(6,22) XDEG,XMIN,YDEG,YMIN,ZVAL              RFMT0034
 22   FORMAT(1H ,I3,F6.2,10X,I3,F6.2,15X,F7.2)                          RFMT0035
      IF (OPT .EQ. 2) WRITE(6,23) X1,X2,X3,XDEG,XMIN,X4,YDEG,YMIN,X5,X6,RFMT0036
     1                            X7,X8,X9,X10,X11,ZVAL                 RFMT0037
 23   FORMAT(1H ,2A6,A2,I3,F6.2,A3,I3,F6.2,6A6,A1,F7.2)                 RFMT0038
 24   I=I+1                                                             RFMT0039
      Z(I)=ZVAL                                                         RFMT0040
      XLAT(I)=SECOND(XDEG,XMIN)                                         RFMT0041
      YLONG(I)=SECOND(YDEG,YMIN)                                        RFMT0042
      DIFF=YLONG(I)-CMLONG                                              RFMT0043
      CALL POLY(CMLAT,XLAT(I),DIFF,X(I),Y(I))                           RFMT0044
      GO TO 1000                                                        RFMT0045
 999  CALL MINMAX(X,Y,I,XMIN,XMAX,YMIN,YMAX)                            RFMT0046
      WRITE(10,25) I,XMIN,XMAX,YMIN,YMAX                                RFMT0047
 25   FORMAT(I10,4F14.4)                                                RFMT0048
      DO 40 J=1,I                                                       RFMT0049
      WRITE(10,35) X(J),Y(J),Z(J)                                       RFMT0050
 35   FORMAT(3F20.4)                                                    RFMT0051
 40   CONTINUE                                                          RFMT0052
      WRITE(6,55) I                                                     RFMT0053
 55   FORMAT(11H1THERE ARE ,I6,16H ORIGINAL POINTS///)                  RFMT0054
      WRITE(6,60) YMIN,YMAX,XMIN,XMAX                                   RFMT0055
 60   FORMAT(53H MINIMUM AND MAXIMUM LATITUDE AND LONGITUDE IN METERS// RFMT0056
     119H MINIMUM LATITUDE =,F14.4/19H MAXIMUM LATITUDE =,F14.4/19H MINIRFMT0057
     2MUM LONGITUDE=,F14.4/19H MAXIMUM LONGITUDE=,F14.4///)             RFMT0058
      D=SQRT((XMAX-XMIN)*(YMAX-YMIN)/I)                                 RFMT0059
      DO 100 J=1,I                                                      RFMT0060
      X(J)=X(J)-XMIN                                                    RFMT0061
      Y(J)=Y(J)-YMIN                                                    RFMT0062
 100  WRITE(10,35) X(J),Y(J),Z(J)                                       RFMT0063
      CALL MINMAX(XLAT,YLONG,I,XMIN,XMAX,YMIN,YMAX)                     RFMT0064
      XDEG1=INT(XMIN/3600.)                                             RFMT0065
      XMIN1=ABS(AMOD(XMIN,3600.)/60.)                                   RFMT0066
      XDEG2=INT(XMAX/3600.)                                             RFMT0067
      XMIN2=ABS(AMOD(XMAX,3600.)/60.)                                   RFMT0068
      YDEG1=INT(YMIN/3600.)                                             RFMT0069
      YMIN1=ABS(AMOD(YMIN,3600.)/60.)                                   RFMT0070
      YDEG2=INT(YMAX/3600.)                                             RFMT0071
      YMIN2=ABS(AMOD(YMAX,3600.)/60.)                                   RFMT0072
      WRITE(6,70) XDEG1,XMIN1,XDEG2,XMIN2,YDEG1,YMIN1,YDEG2,YMIN2       RFMT0073
```

```
 70   FORMAT(//54H MINIMUM AND MAXIMUM LATITUDE AND LONGITUDE IN DEGREE
     1//19H MINIMUM LATITUDE =,I4,F6.2/19H MAXIMUM LATITUDE =,I4,F6.2/
     219H MINIMUM LONGITUDE=,I4,F6.2/19H MAXIMUM LONGITUDE=,I4,F6.2////)
      WRITE(6,80) D
 80   FORMAT(//42H AVERAGE DISTANCE BETWEEN DATA POINTS, D =,F12.2//57H
     1THE GRID INTERVAL, G, SHOULD BE SUCH THAT .25(D)≤G≤.5(D)/94H TO CA
     2LCULATE THE GRID SIZE, DIVIDE G INTO THE DIFFERENCES IN LATITUDE A
     3ND LONGITUDE IN METERS)
      STOP
      END
      FUNCTION SECOND(DEG,FMIN)
C
C CONVERTS ANGLE IN DEGREES AND MINUTES TO SECONDS.
C SIGN OF ANGLE = SIGN OF DEGREES
C
      SECOND=SIGN(1.,DEG)*(3600.*ABS(DEG)+60.*FMIN)
      RETURN
      END
      SUBROUTINE POLY(P1,P2,DL,X,Y)
C
C POLYCONIC PROJECTION OF POINT WITH LATITUDE=P2. P1=LATITUDE OF CENTRAL
C MERIDIAN. DL=LONGITUDINAL DIFFERENCE FROM CENTRAL MERIDIAN TO POINT.
C X=DISTANCE FROM CENTRAL MERIDIAN ALONG LATITUDE P2. Y=DISTANCE FROM
C P1 TO P2.  X,Y IN METERS.   P1,P2,DL IN SECONDS.
C
      REAL LA,IP,IPR
      DATA ARCONE,ESQ,LA,A0,A2,A4,A6,A8/4.8481368E-6,6.7686580E-3,
     1     6378206.4,6367399.7,32433.888,34.4187,.0454,6.0E-5/
      IP=P2-P1
      SINP2=SIN(P2*ARCONE)
      COSP2=COS(P2*ARCONE)
      THETA=DL*SINP2
      A=SQRT(1.0-(ESQ*(2.*SINP2)))/(LA*ARCONE)
      IF (SINP2) 10,20,10
 10   COT=COSP2/SINP2
      GO TO 30
 20   COT=1.0E68
 30   X=(COT*SIN(THETA*ARCONE))/(A*ARCONE)
      IPR=IP*ARCONE
      PR=((P2+P1)/2.)*ARCONE
      Y=A0*IPR-(A2*COS(2.*PR)*SIN(IPR))+(A4*COS(4.*PR)*SIN(2.*IPR))-
     1  (A6*COS(6.*PR)*SIN(3.*IPR))+A8*COS(8.*PR)*SIN(4.*IPR)
      RETURN
      END
      SUBROUTINE MINMAX(ALAT,ALONG,MPTS,XMIN,XMAX,YMIN,YMAX)
      DIMENSION ALAT(3000),ALONG(3000)
C
C FINDS THE MINIMUM AND MAXIMUM OF THE LATITUDE AND LONGITUDE VALUES
C
      XMIN=ALAT(1)
      XMAX=ALAT(1)
      YMIN=ALONG(1)
      YMAX=ALONG(1)
      DO 10 I=2,MPTS
      IF (XMIN .GT. ALAT(I)) XMIN=ALAT(I)
      IF (XMAX .LT. ALAT(I)) XMAX=ALAT(I)
      IF (YMIN .GT. ALONG(I)) YMIN=ALONG(I)
      IF (YMAX .LT. ALONG(I)) YMAX=ALONG(I)
 10   CONTINUE
      RETURN
      END
```

APPENDIX II: GRID GENERATION PROGRAM LISTING

```
FILE   2=CPH,UNIT=PUNCH,RECORD=10
FILE   5=CR,UNIT=READER,RECORD=10
FILE   6=LP,UNIT=PRINTER,RECORD=15
FILE   10=CONTOUR/INPUT,UNIT=TAPE,BLOCKING=20,RECORD=10
FILE   11=GRIDDED/DATA,UNIT=TAPE,SAVE=633,BLOCKING=20,RECORD=10
C                                                                       GRID0001
C      GRID GENERATION PROGRAM                                          GRID0002
C                                                                       GRID0003
C                                                                       GRID0004
C        RANDOM DATA TAPE -- BINARY, UNIT 10.                           GRID0005
C        GRID OUTPUT TAPE -- BINARY, UNIT 11.                           GRID0006
C                                                                       GRID0007
      INTEGER OPT(4)                                                    GRID0008
      DIMENSION WZ(122,122),W(122,122),X(1500),Y(1500),Z(1500),         GRID0009
     1INX(4),JNX(4)                                                     GRID0010
      DIMENSION XTEMP(1000),YTEMP(1000),ZTEMP(1000)                     GRID0011
 10   READ(5,20) OPT,M,N,DELY,DELX,SCALE                                GRID0012
 20   FORMAT(4I1,2I3,2I4,F5.2)                                          GRID0013
      IF (M.LE.0.OR.N.LE.0) GO TO 90                                    GRID0014
      IF (M .GT. 122 .OR. N .GT. 122) GO TO 90                          GRID0015
 30   READ(10,35) NP,XMIN,XMAX,YMIN,YMAX                                GRID0016
 35   FORMAT(I10,4F14.4)                                                GRID0017
      GO TO 120                                                         GRID0018
   90 WRITE (6,100)                                                     GRID0019
 100  FORMAT (57H1PARAMETER CARD ERROR -- M OR N .GT. 122 OR MIN=MAX ERRGRID0020
     1OR)                                                               GRID0021
      CALL EXIT                                                         GRID0022
C  CHECKS ON PARAMETERS OK.                                             GRID0023
  120 M1=M+1                                                            GRID0024
      N1=N+1                                                            GRID0025
      M2=M+2                                                            GRID0026
      N2=N+2                                                            GRID0027
      DELX2=DELX**2                                                     GRID0028
      DELY2=DELY**2                                                     GRID0029
      XFACT=DELX*6.1237E-4/SCALE                                        GRID0030
      YFACT=DELY*6.1237E-4/SCALE                                        GRID0031
C     READ RANDOM DATA RECORD                                           GRID0032
      DO 140 J=1,NP                                                     GRID0033
 140  READ(10,145) X(J),Y(J),Z(J)                                       GRID0034
 145  FORMAT(3F20.4)                                                    GRID0035
  150 IF (OPT(3).EQ.1) GO TO 530                                        GRID0036
      K=1                                                               GRID0037
      KK=NP+1                                                           GRID0038
 160  IF (K.GE.KK) GO TO 310                                            GRID0039
      XK=X(K)                                                           GRID0040
      YK=Y(K)                                                           GRID0041
      ZK=Z(K)                                                           GRID0042
  220 IF (XK.LT.XMIN-DELX.OR.XK.GE.XMAX+DELX.OR.YK.LT.YMIN-DELY.OR.YK.GEGRID0043
     1.YMAX+DELY) GO TO 260                                             GRID0044
C     (I,J) ARE INDICES OF UPPER LEFT CORNER OF NEAREST GRID POINT.     GRID0045
      J=(XK-XMIN)/DELX+2.                                               GRID0046
      I=(YK-YMIN)/DELY+2.                                               GRID0047
      X1=XMIN+DELX*FLOAT(J-2)                                           GRID0048
      Y1=YMIN+DELY*FLOAT(I-2)                                           GRID0049
      IF (XK.EQ.X1.AND.YK.EQ.Y1) GO TO 520                              GRID0050
      X2=X1+DELX                                                        GRID0051
      Y2=Y1+DELY                                                        GRID0052
      XD1=(XK-X1)**2                                                    GRID0053
      XD2=(XK-X2)**2                                                    GRID0054
      YD1=(YK-Y1)**2                                                    GRID0055
      YD2=(YK-Y2)**2                                                    GRID0056
      W1=1./(XD1+YD1)                                                   GRID0057
      W2=1./(XD2+YD1)                                                   GRID0058
      W3=1./(XD1+YD2)                                                   GRID0059
      W4=1./(XD2+YD2)                                                   GRID0060
      IF (W(I,J).NE.-1.) GO TO 270                                      GRID0061
  230 IF (W(I,J+1).NE.-1.) GO TO 280                                    GRID0062
  240 IF (W(I+1,J).NE.-1.) GO TO 290                                    GRID0063
  250 IF (W(I+1,J+1).NE.-1.) GO TO 300                                  GRID0064
  260 K=K+1                                                             GRID0065
C     PROCESS NEXT K-TH OBSERVATION                                     GRID0066
      GO TO 160                                                         GRID0067
  270 W(I,J)=W(I,J)+W1                                                  GRID0068
      WZ(I,J)=WZ(I,J)+W1*ZK                                             GRID0069
      GO TO 230                                                         GRID0070
```

```
  280 W(I,J+1)=W(I,J+1)+W2                                              GRID0071
      WZ(I,J+1)=WZ(I,J+1)+W2*ZK                                         GRID0072
      GO TO 240                                                         GRID0073
  290 W(I+1,J)=W(I+1,J)+W3                                              GRID0074
      WZ(I+1,J)=WZ(I+1,J)+W3*ZK                                         GRID0075
      GO TO 250                                                         GRID0076
  300 W(I+1,J+1)=W(I+1,J+1)+W4                                          GRID0077
      WZ(I+1,J+1)=WZ(I+1,J+1)+W4*ZK                                     GRID0078
      GO TO 260                                                         GRID0079
C     END-OF-DATA OF CURRENT FILE                                       GRID0080
  310 DO 340 I=1,M2                                                     GRID0081
      DO 340 J=1,N2                                                     GRID0082
      IF (W(I,J)) 340,320,330                                           GRID0083
C     MISSING POINT FLAG=1.0E10 - OR POLE DATA IF OPT(4)=1              GRID0084
  320 WZ(I,J)=1.0E10                                                    GRID0085
      GO TO 340                                                         GRID0086
C     COMPUTE WEIGHTED AVERAGE                                          GRID0087
  330 WZ(I,J)=WZ(I,J)/W(I,J)                                            GRID0088
  340 CONTINUE                                                          GRID0089
      K=0                                                               GRID0090
C     SCAN WZ(,) FOR MISSING POINTS -- IF ANY                           GRID0091
  350 DO 360 I=2,M1                                                     GRID0092
      DO 360 J=2,N1                                                     GRID0093
      IF (WZ(I,J).EQ.1.0E10) GO TO 370                                  GRID0094
  360 CONTINUE                                                          GRID0095
C     NO POINTS MISSING -- PREPARE FOR OUTPUT                           GRID0096
      GO TO 400                                                         GRID0097
C     SOME POINTS MISSING -- INCR K BY 1 (KTH LOCAL GRID MODEL)         GRID0098
  370 K=K+1                                                             GRID0099
      IF (OPT(4).EQ.1) GO TO 590                                        GRID0100
      DO 390 I=2,M1                                                     GRID0101
      DO 390 J=2,N1                                                     GRID0102
      IF (WZ(I,J).NE.1.0E10) GO TO 390                                  GRID0103
C     MISSING POINT I,J -- FIND KTH LOCAL POINTS FOR AVERAGE (K.GE.1)   GRID0104
      WSUM=0.                                                           GRID0105
      WZSUM=0.                                                          GRID0106
      IK1=I-K                                                           GRID0107
      IK2=I+K                                                           GRID0108
      JK1=J-K                                                           GRID0109
      JK2=J+K                                                           GRID0110
      DO 380 IK=IK1,IK2                                                 GRID0111
      DO 380 JK=JK1,JK2                                                 GRID0112
      IF (IK.LT.1.OR.IK.GT.M2.OR.JK.LT.1.OR.JK.GT.N2) GO TO 380         GRID0113
      IF (WZ(IK,JK).EQ.1.0E10) GO TO 380                                GRID0114
      WT=1./(FLOAT(IK-I)**2*DELY2+FLOAT(JK-J)**2*DELX2)                 GRID0115
      WSUM=WSUM+WT                                                      GRID0116
      WZSUM=WZSUM+WT*WZ(IK,JK)                                          GRID0117
  380 CONTINUE                                                          GRID0118
      IF (WSUM.EQ.0.) GO TO 390                                         GRID0119
C     COMPUTE LOCAL AVERAGE FOR WZ(I,J) AND CONTINUE MISSING POINT SCAN.GRID0120
      WZ(I,J)=WZSUM/WSUM                                                GRID0121
  390 CONTINUE                                                          GRID0122
C     FINISHED KTH SCAN---CHECK IF THERE ARE STILL SOME POINTS MISSING  GRID0123
      GO TO 350                                                         GRID0124
C     ALL POINTS FOUND IN WZ(I,J) - OR POLE DATA IF OPT(4)=1            GRID0125
  400 ZMIN=1.0E30                                                       GRID0126
      ZMAX=-ZMIN                                                        GRID0127
C     DETERMINE ZMIN,ZMAX BUT EXCLUDING ALL FLAGGED MISSING PTS (1.0E10)GRID0128
      DO 410 I=2,M1                                                     GRID0129
      DO 410 J=2,N1                                                     GRID0130
      IF (WZ(I,J).LT.ZMIN.AND.WZ(I,J).NE.1.0E10) ZMIN=WZ(I,J)           GRID0131
      IF (WZ(I,J).GT.ZMAX.AND.WZ(I,J).NE.1.0E10) ZMAX=WZ(I,J)           GRID0132
  410 CONTINUE                                                          GRID0133
      DO 415 I=1,NP                                                     GRID0134
      READ(10,412) X(I),Y(I),Z(I)                                       GRID0135
  412 FORMAT(3F20.4)                                                    GRID0136
      X(I)=X(I)/DELX*XFACT                                              GRID0137
      Y(I)=Y(I)/DELY*YFACT                                              GRID0138
  415 CONTINUE                                                          GRID0139
      WRITE(6,420) OPT,M,N,XMIN,XMAX,YMIN,YMAX                          GRID0140
  420 FORMAT(9H1OPTIONS=,4I1,9H  ROWS=M=,I3,12H  COLUMNS=N=,I3//7H  XMINGRID0141
     1=,E15.8,7H  XMAX=,E15.8,7H  YMIN=,E15.8,7H  YMAX=,E15.8//18H GRID GRID0142
     2GENERATED...//)                                                   GRID0143
      NROWS=9./YFACT                                                    GRID0144
      IF (NROWS .GT. M) NROWS=M                                         GRID0145
      K=N+1                                                             GRID0146
      J=NROWS+1                                                         GRID0147
      I=2                                                               GRID0148
      NBLKS=0                                                           GRID0149
      TYMAX=M*YFACT                                                     GRID0150
      TMAX=3.*YFACT                                                     GRID0151
      RATIO=FLOAT(NROWS)/FLOAT(M)                                       GRID0152
```

```
421   IJ=MINO(J,M+1)                                                    GRID0153
      TMIN=TMAX-(3.*YFACT)                                              GRID0154
      TMAX=TMIN+NROWS*YFACT                                             GRID0155
      NBLKS=NBLKS+1                                                     GRID0156
      NPTS=0                                                            GRID0157
      DO 425 IL=1,NP                                                    GRID0158
      IF (Y(IL) .LT. TMIN .OR. Y(IL) .GT. TMAX) GO TO 425               GRID0159
      NPTS=NPTS+1                                                       GRID0160
      XTEMP(NPTS)=X(IL)                                                 GRID0161
      YTEMP(NPTS)=Y(IL)-TMIN                                            GRID0162
      ZTEMP(NPTS)=Z(IL)                                                 GRID0163
425   CONTINUE                                                          GRID0164
      JM=IJ-I+1                                                         GRID0165
      JN=N                                                              GRID0166
      WRITE(2,426) JM,JN,NPTS,OPT(2)                                    GRID0167
426   FORMAT(8H"ZDATA",,I3,1H,,I3,1H,,I4,1H,,I1,1H,)                    GRID0168
      IF (NPTS .EQ. 0) GO TO 430                                        GRID0169
      DO 430 KK=1,NPTS                                                  GRID0170
      IF (OPT(2) .EQ. 0) WRITE(11,427) XTEMP(KK),YTEMP(KK),ZTEMP(KK)    GRID0171
      IF (OPT(2) .EQ. 1) WRITE(2,427) XTEMP(KK),YTEMP(KK),ZTEMP(KK)     GRID0172
427   FORMAT(3F14.4)                                                    GRID0173
430   CONTINUE                                                          GRID0174
      DO 435 KK=I,IJ                                                    GRID0175
      IF (OPT(2) .EQ. 0) WRITE(11,428) (WZ(KK,JJ),JJ=2,K)               GRID0176
      IF (OPT(2) .EQ. 1) WRITE(2,428) (WZ(KK,JJ),JJ=2,K)                GRID0177
428   FORMAT(8F10.4)                                                    GRID0178
435   CONTINUE                                                          GRID0179
      J=J+NROWS-3                                                       GRID0180
      I=IJ-2                                                            GRID0181
      IF (IJ .LT. (M+1)) GO TO 421                                      GRID0182
      WRITE(6,437) NBLKS,NROWS,XFACT,YFACT                              GRID0183
437   FORMAT(36H NUMBER OF BLOCKS TO BE CONTOURED = ,I2/14H DIVIDED INTOGRID0184
     1 ,I3,58H ROWS EACH, WITH 3 ROWS OVERLAPPING ONE BLOCK AND THE NEXTGRID0185
     2///25H SCALE FACTOR (XSIZE) IS ,F5.3                              GRID0186
     3  /25H SCALE FACTOR (YSIZE) IS ,F5.3//)                           GRID0187
      IF (OPT(1).NE.1) GO TO 500                                        GRID0188
C     PRINT GRID OPT(1)=1                                               GRID0189
      DO 450 I=2,M1                                                     GRID0190
      IR=I-1                                                            GRID0191
      WRITE (6,440) IR,(WZ(I,J),J=2,N1)                                 GRID0192
  440 FORMAT (5H ROW ,I3/(1P10E12.4))                                   GRID0193
  450 CONTINUE                                                          GRID0194
  500 WRITE (6,510) ZMIN,ZMAX                                           GRID0195
  510 FORMAT  (6H0ZMIN=,E15.8,/H  ZMAX=,E15.8)                          GRID0196
      CALL EXIT                                                         GRID0197
C     DATA POINT EXACTLY ON GRID INTERSECTION -- FLAG BY W(I,J)=-1.     GRID0198
  520 W(I,J)=-1.                                                        GRID0199
      WZ(I,J)=ZK                                                        GRID0200
      GO TO 260                                                         GRID0201
C     OPT(3)=1 -- LIST RANDOM DATA READ FROM TAPE (RECORD AT A TIME)    GRID0202
  530 WRITE (6,540)                                                     GRID0203
  540 FORMAT (12H1RANDOM DATA          //,7X,1HX,11X,1HY,10X,1HZ)       GRID0204
      WRITE(6,550) (X(J),Y(J),Z(J),J=1,NP)                              GRID0205
  550 FORMAT (1X,1P3E12.4)                                              GRID0206
      K=1                                                               GRID0207
      KK=NP+1                                                           GRID0208
      GO TO 160                                                         GRID0209
  590 DO 610 I=2,M1                                                     GRID0210
      DO 610 J=2,N1                                                     GRID0211
      IF (WZ(I,J).NE.1.0E10) GO TO 610                                  GRID0212
C     MISSING PT -- USE 4 PT BOUNDARY AVG METHOD (WT=1.0) -- OPT(4)=1   GRID0213
      WSUM=0.                                                           GRID0214
      WZSUM=0.                                                          GRID0215
      INX(1)=I                                                          GRID0216
      JNX(1)=J-1                                                        GRID0217
      INX(2)=I+1                                                        GRID0218
      JNX(2)=J                                                          GRID0219
      INX(3)=I                                                          GRID0220
      JNX(3)=J+1                                                        GRID0221
      INX(4)=I-1                                                        GRID0222
      JNX(4)=J                                                          GRID0223
      DO 600 L=1,4                                                      GRID0224
      IK=INX(L)                                                         GRID0225
      JK=JNX(L)                                                         GRID0226
      IF (WZ(IK,JK).EQ.1.0E10) GO TO 600                                GRID0227
      WSUM=WSUM+1.                                                      GRID0228
      WZSUM=WZSUM+WZ(IK,JK)                                             GRID0229
```

```
  600 CONTINUE                                                          GRTD0230
      IF (WSUM.EQ.0.) GO TO 610                                         GRTD0231
C     COMPUTE 1 TO 4 PT AVG NEAR BOUNDARY OF KNOWN PTS...CONTINUE SCAN  GRTD0232
      WZ(I,J)=WZSUM/WSUM                                                GRTD0233
  610 CONTINUE                                                          GRTD0234
      GO TO 350                                                         GRTD0235
      STOP                                                              GRTD0236
      END                                                               GRTD0237
```

APPENDIX III: CONTOUR PROGRAM LISTING

```
BEGIN
                        %********************
COMMENT CONTOUR PROGRAM.
        ALL PARAMETERS IN THIS BLOCK FOR GLOBAL CONTROL;
                        %********************
    DEFINE FORI=FOR I←0 STEP 1 UNTIL #,FORJ=FOR J←0 STEP 1 UNTIL #;
    FILE   IN TAPED "GRIDDED" "DATA" (2, 10, 200);
    REAL LY,LX,LD,CPI,YSIZE,XSIZE,SCAL,ADJUST,ADJUST2,DX,DY,DELCV,
        MINCV,MAXCV,PRIMC,TYPE;
    INTEGER IMAX,JMAX,I,J,JSUB,ISUB,NSUB,NUMX,NUMY,NCVAL,LABL,BLOCK,
            ORPTS,NDEC,NORPT,ISIZE,JSIZE,DELY,DELX;
    ARRAY PCODE[0:127],DUM[0:0];
    INTEGER ARRAY IJA [0 : 19], SDE [0 : 4];
    BOOLEAN ADJST, CARRA;
    FORMAT FMTED("ERROR END OF JOB"),
          FMTEN("NORMAL END OF JOB");
    LABEL EOJE, EOJ, START, QUIT, EOFA;
                        %********************
    COMMENT  PLOTTER PACKAGE FOR CALCOMP 570 HAS BEEN INSERTED HERE,
        ALL CALLS ARE STANDARD;
                        %********************
    BOOLEAN STREAM PROCEDURE COMPR (A, B);
    VALUE B;
    BEGIN
        LABEL TR, QUIT;
        SI ← A;
        DI ← LOC B;
        IF 8 SC = DC THEN GO TO TR;
        DI ← LOC COMPR;
        DS ← 8 LIT "0";
        GO TO QUIT;
        TR: DI ← LOC COMPR;
        DS ← 8 LIT "00000001";
        QUIT:
    END;
                        %********************
    FILL IJA[*] WITH 0,1,0,0,0,0,1,1,1,0,0,0,0,1,0,1,1,0,1,1;
    FILL SDE [*] WITH 0, 2, 1, 3, 0;
    START:
COMMENT GENERAL MAP INPUT SEGMENT;
                        %********************
    BEGIN
        REAL INTVAL;
        LABEL START, EOFE, QUIT;
COMMENT READ IN GENERAL CONTROL PARAMETERS;
        START: READ (CR [NO], /, TYPE) [EOJ];
        IF NOT COMPR (TYPE, "MAP") THEN
        BEGIN
            READ (CR);
            GO TO START
        END;
        READ(CR,/,TYPE,JSIZE,ISIZE,NUMY,NUMX,YSIZE,XSIZE,INTVAL,JSUB,
            ISUB,ADJUST,ADJUST2,NSUB,DELY,DELX);
        IF ADJUST = 0 THEN ADJST ← TRUE ELSE ADJST ← FALSE;
        IF INTVAL = 0 THEN INTVAL ← 0.05;
        IF NUMX = 0 AND NUMY = 0 THEN
        BEGIN
            NUMX ← XSIZE / INTVAL;
            NUMY ← YSIZE / INTVAL;
        END;
        IF NUMX = 0 THEN NUMX ← 1 ELSE IF NUMY = 0 THEN NUMY ← 1;
        IF JSUB = 0 THEN JSUB ← 64;
        IF ISUB = 0 THEN ISUB ← 64;
        DX ← 1 / NUMX;
        DY ← 1 / NUMY;
COMMENT READ CONTOUR SPECIFICATION CARD;
        READ (CR [NO], /, TYPE) [EOFE];
        IF NOT COMPR (TYPE, "CONTR") THEN
        BEGIN
            WRITE(LP,<"CANNOT FIND CONTR CARD">);
            GO TO EOJE
        END;
        READ(CR,/,TYPE,DELCV,MINCV,MAXCV,PRIMC,NCVAL,LABL,CPI,NDEC,
            SCAL,NORPT);
        IF NCVAL ≠ 0 THEN NCVAL ← NCVAL - 1;
        IF DELCV = 0 THEN CARRA ← TRUE ELSE CARRA ← FALSE;
```

```
        % INITIALIZE PLOTTER TAPE                                        CNTR0077
        PLOT (0, 0, - 5);                                                CNTR0078
        IF CPI > 0 THEN                                                  CNTR0079
        BEGIN                                                            CNTR0080
             LX ← DELCV × XSIZE;                                         CNTR0081
             LY ← DELCV × YSIZE;                                         CNTR0082
             LD ← SQRT (XSIZE ↑ 2 + YSIZE ↑ 2) × DELCV;                  CNTR0083
        END;                                                             CNTR0084
        IF CPI > 0 THEN FORI 127 DO PCODE [I] ← 0;                       CNTR0085
        GO TO QUIT;                                                      CNTR0086
        EOFE: WRITE(LP,<"TRYING TO READ CONTR CARD, BUT NO MORE "        CNTR0087
                  "CARDS IN FILE">);                                     CNTR0088
        GO TO EOJE;                                                      CNTR0089
COMMENT COMPLETED INPUT OF GENERAL MAP PARAMETERS;                       CNTR0090
        QUIT;                                                            CNTR0091
    END;                                                                 CNTR0092
                        %********************                            CNTR0093
COMMENT READING IN SUB-BLOCKS AND CONTROL;                               CNTR0094
                        %********************                            CNTR0095
    BEGIN                                                                CNTR0096
COMMENT START OF SUB-BLOCK 2.                                            CNTR0097
       THIS IS ITERATIVE READING OF GRID SUB-SECTIONS AND CONTOURING.    CNTR0098
       DYNAMIC ARRAYS ALSO REQUIRE SUB-BLOCKING;                         CNTR0099
         ARRAY Z[0:JSUB,0:ISUB],CVALA[0:NCVAL],CODE[0:511],DUM[0:5];     CNTR0100
         INTEGER ARRAY JA, JB, IA, IB [0 : 4];                           CNTR0101
         INTEGER BLOCK,POINTS,CVP,NTIME,FD,FM,LGMIND,LGMINM,LGMAXM,      CNTR0102
                 LGMAXD,PTSGEN;                                          CNTR0103
         BOOLEAN PRIMF;                                                  CNTR0104
         REAL ZMAX,ZMIN,CVAL,ENDCV,TEMP,LTMN,LTMX,ZLTD,ZLTM,ZLGD,ZLGM,   CNTR0105
              MAXLTD,MAXLTM,MAXLGD,MAXLGM,DLT,DLG,LAST,HGT,LL,SD,LASTA,  CNTR0106
              DT,DIFF,DG,KY;                                             CNTR0107
         OWN REAL TMIN,LGMN,LGMX;                                        CNTR0108
         OWN INTEGER LATMAXM,LATMINM,LONGMAXM,LONGMINM,LTMINM,LTMIND,    CNTR0109
                     LATMAXD,LATMIND,LONGMAXD,LONGMIND;                  CNTR0110
         LABEL EOFK,SKIPCON;                                             CNTR0111
                        %********************                            CNTR0112
         REAL PROCEDURE AJUST (Z);                                       CNTR0113
         REAL Z;                                                         CNTR0114
         BEGIN                                                           CNTR0115
              IF ABS (Z - CVAL) ≤ ADJUST THEN AJUST ← Z + ADJUST2 ELSE   CNTR0116
              AJUST ← Z                                                  CNTR0117
         END;                                                            CNTR0118
                        %********************                            CNTR0119
         READ (CR [NO], /, TYPE);                                        CNTR0120
         IF NOT COMPR (TYPE, "LAT") THEN GO TO EOFK ELSE READ (CR, /,    CNTR0121
         TYPE, ZLTD, ZLTM, ZLGD, ZLGM, MAXLTD, MAXLTM, MAXLGD, MAXLGM,   CNTR0122
         DLT, DLG, LATMAXD, LATMAXM, LATMIND, LATMINM, LONGMAXD,         CNTR0123
         LONGMAXM, LONGMIND, LONGMINM);                                  CNTR0124
         WHILE FALSE DO                                                  CNTR0125
         BEGIN                                                           CNTR0126
              EOFK: WRITE(LP,<"CANNOT FIND LAT CARD">);                  CNTR0127
              GO TO EOJE;                                                CNTR0128
         END;                                                            CNTR0129
         NSUB ← NSUB - 1;                                                CNTR0130
         FOR BLOCK ← 0 STEP 1 UNTIL NSUB DO                              CNTR0131
         BEGIN                                                           CNTR0132
         %  ADVANCE PLOTTER AND SET UP NEW BLOCK NUMBER                  CNTR0133
              IF BLOCK > 0 THEN PLOT (XSIZE × IMAX + 6, 0, - 3);         CNTR0134
              IF BLOCK = 1 THEN PLOT (5, 0, - 5);                        CNTR0135
COMMENT Z DATA INPUT AND CVALA INPUT BLOCK;                              CNTR0136
                        %********************                            CNTR0137
              BEGIN                                                      CNTR0138
                   REAL A,B,ORX,ORY,ORZ,DG;                              CNTR0139
                   INTEGER N, ZFILE, K, J08;                             CNTR0140
                   ARRAY AA [0 : JSUB - 1];                              CNTR0141
                   FORMAT FMT07 (8 F10 . 4);                             CNTR0142
                   FORMAT FMT06 (3 F14 . 4);                             CNTR0143
                   LABEL EOFA,QUIT,EOFE,EOF,ONCE,NEXT,NAXT,FINLG,FXT,    CNTR0144
                         LGO,EOFC;                                       CNTR0145
                   BOOLEAN ONC,MAXLAT,MINLAT;                            CNTR0146
                   OWN BOOLEAN BOX;                                      CNTR0147
                   DEFINE P5 = 0.5 #;                                    CNTR0148
                        %********************                            CNTR0149
                   STREAM   PROCEDURE RESETT (A);                        CNTR0150
                   BEGIN                                                 CNTR0151
                        DI ← A;                                          CNTR0152
                        4 (32 (DS ← 32 LIT "0"))                         CNTR0153
                   END;                                                  CNTR0154
                        %********************                            CNTR0155
```

```
COMMENT READ IN CONTOUR LIST , IF NECESSARY;                          CNTR0156
                IF CARRA AND BLOCK = 0 THEN READ (CR, /, FORI NCVAL CNTR0157
                DO CVALA [I]) [EOFA];                                 CNTR0158
COMMENT Z DATA INPUT SECTION;                                         CNTR0159
                READ (CR [N0], /, TYPE) [EOFE];                       CNTR0160
                IF NOT COMPR (TYPE, "ZDATA") THEN                     CNTR0161
                BEGIN                                                 CNTR0162
                     WRITE(LP,<"CANNOT FIND ZDATA CARD">);            CNTR0163
                     GO TO EOJE;                                      CNTR0164
                END;                                                  CNTR0165
                READ (CR, /, TYPE, JMAX, IMAX, ORPTS, ZFILE);         CNTR0166
                IMAX ← IMAX - 1;                                      CNTR0167
                JMAX ← JMAX - 1;                                      CNTR0168
COMMENT   DRAW LINES .5 INCHES LONG AT THE EDGES ON TOP & BOTTOM OF   CNTR0169
          BLOCKS FOR MATCHING SECTIONS;                               CNTR0170
                IF BLOCK < NSUB THEN                                  CNTR0171
                BEGIN                                                 CNTR0172
                     SYMBOL(0.0,YSIZE×(JMAX-1),.6,DUM,90,-8);         CNTR0173
                     SYMBOL(XSIZE×(IMAX+1)+.25,YSIZE×(JMAX-1),.6,     CNTR0174
                            DUM,90,-8);                               CNTR0175
                END;                                                  CNTR0176
                IF BLOCK > 0 THEN                                     CNTR0177
                BEGIN                                                 CNTR0178
                     SYMBOL (0.0, YSIZE, .6, DUM, 90, - 8);           CNTR0179
                     SYMBOL(XSIZE×(IMAX+1)+.25,YSIZE,.6,DUM,90,-8);   CNTR0180
                END;                                                  CNTR0181
COMMENT   LABEL EACH BLOCK OF THE MAP;                                CNTR0182
                DUM [0] ← "BLOCK ";                                   CNTR0183
                SYMBOL (0, 9.5, .15, DUM, 0, 6);                      CNTR0184
                NUMBER (.5, 9.5, .15, BLOCK, 0, 0);                   CNTR0185
COMMENT   READ IN AND PLOT ORIGINAL POINTS,                           CNTR0186
          READ THROUGH POINTS IF LABL =2 ,                            CNTR0187
          DRAW A + FOR EACH POINT WITHOUT LABELING IF LABL ≠ 1 OR 2   CNTR0188
          LABEL POINTS IF LABL = 1 ;                                  CNTR0189
                FOR K ← 1 STEP 1 UNTIL ORPTS DO                       CNTR0190
                BEGIN                                                 CNTR0191
                     IF ZFILE = 1 THEN READ (CR, FMT06, ORX, ORY,     CNTR0192
                     ORZ) ELSE READ (TAPED, FMT06, ORX, ORY, ORZ);    CNTR0193
                     IF LABL ≠ 2 THEN                                 CNTR0194
                     BEGIN                                            CNTR0195
                          SYMBOL (ORX, ORY, .05, DUM, 0, - 13);       CNTR0196
                          IF LABL = 1 THEN                            CNTR0197
                          BEGIN                                       CNTR0198
                               DG ← 0;                                CNTR0199
                               FOR N ← 1 STEP 1 WHILE DG = 0 DO IF    CNTR0200
                               ENTIER (ABS (ORZ) / 10 * N) = 0 THEN   CNTR0201
                               DG ← N;                                CNTR0202
                               IF ORZ < 0 THEN DG ← DG + 1;           CNTR0203
                               DG ← - .1 × (4 - DG);                  CNTR0204
                               NUMBER(ORX+DG,ORY,.1,ORZ,0,1);         CNTR0205
                          END;                                        CNTR0206
                     END;                                             CNTR0207
                END;                                                  CNTR0208
COMMENT  LABEL LONGITUDE ON TOP AND BOTTOM OF MAP;                    CNTR0209
                IF BLOCK = 0 OR BLOCK = NSUB THEN                     CNTR0210
                BEGIN                                                 CNTR0211
                     IF BLOCK = 0 AND NSUB ≠ 0 THEN                   CNTR0212
                     BEGIN                                            CNTR0213
                          HGT ← 0;                                    CNTR0214
                          ONC ← TRUE;                                 CNTR0215
                          GO TO ONCE;                                 CNTR0216
                     END ELSE IF NSUB ≠ 0 THEN                        CNTR0217
                     BEGIN                                            CNTR0218
                          HGT ← (JMAX) × YSIZE;                       CNTR0219
                          ONC ← TRUE;                                 CNTR0220
                          GO TO ONCE;                                 CNTR0221
                     END ELSE IF NSUB=0 THEN FOR HGT←0,JMAX×YSIZE DOCNTR0222
          ONCE:      BEGIN                                            CNTR0223
                          FOR LL←DLG STEP DLG UNTIL 60 DO             CNTR0224
                          IF ZLGM/LL≤1 THEN                           CNTR0225
                          BEGIN                                       CNTR0226
                               LGMINM ← LL;                           CNTR0227
                               LGMIND ← ZLGD;                         CNTR0228
                               DIFF ← ZLGM - LGMINM;                  CNTR0229
                               IF LGMINM ≥ 60 THEN                    CNTR0230
                               BEGIN                                  CNTR0231
                                    LGMIND ← ZLGD + 1;                CNTR0232
                                    LGMINM ← LGMINM - 60;             CNTR0233
                               END;                                   CNTR0234
                               GO TO NEXT;                            CNTR0235
                          END;                                        CNTR0236
```

```
COMMENT 1477.8=METERS/MINUTE OF LONGITUDE FOR CENTRAL VIRGINIA;
                NEXT:     DIFF←DIFF×1477.8/DELX×XSIZE;
                          FOR LL ← DLG STEP DLG UNTIL 60 DO IF
                          ENTIER (MAXLGM) / LL ≤ 1 THEN
                          BEGIN
                               LGMAXM ← LL;
                               LGMAXD ← MAXLGD;
                               IF LGMAXM = 60 THEN
                               BEGIN
                                    LGMAXM ← 0;
                                    LGMAXD ← LGMAXD - 1;
                               END;
                               GO TO NAXT;
                          END;
                NAXT:     NTIME←(LGMIND-LGMAXD)×60/DLG+(LGMINM-
                               LGMAXM)/DLG;
COMMENT 1477.8=METERS/MINUTE OF LONGITUDE FOR CENTRAL VIRGINIA;
                          DG ← DLG × 1477.8 / DELX × XSIZE;
        %  TICK MARKS FOR LONGITUDE SCALE
                          FOR LL ← 0 STEP 1 UNTIL NTIME DO SYMBOL (
                          DIFF + LL × DG, HGT, .1, DUM, 0, - 8);
        % DRAW LONGITUDE SCALE LINE
                          PLOT ((IMAX) × XSIZE, HGT, 3);
                          PLOT (DIFF, HGT, 2);
                          IF HGT = (JMAX) × YSIZE THEN HGT ← HGT +
                          .15 ELSE HGT ← HGT - .15;
                          FD ← LGMIND;
                          FM ← LGMINM;
                          FOR LL ← 0 STEP 1 UNTIL NTIME DO
                          BEGIN
                               IF FM = - DLG THEN
                               BEGIN
                                    FM ← 60 - DLG;
                                    FD ← FD - 1;
                               END;
                               IF FD = LONGMAXD AND FM = LONGMAXM
                               THEN LGMX ← DIFF + LL × DG ELSE IF
                               FD = LONGMIND AND FM = LONGMINM THEN
                               LGMN ← DIFF + LL × DG;
        %  LABEL LONGITUDE SCALE
        %  DEGREES
                               NUMBER(DIFF+LL×DG-.8,HGT,.12,FD,0,0);
        %  DEGREE SYMBOL
                               SYMBOL (DIFF + LL × DG - .1, HGT +
                               .08, .07, DUM, 0, - 5);
        %  MINUTES
                               NUMBER(DIFF+LL×DG-.4,HGT,.12,FM,0,0);
        %  MINUTE SYMBOL
                               SYMBOL (DIFF + LL × DG + .3, HGT +
                               .08, .07, DUM, 0, - 8);
                               IF FM ≤ LGMAXM AND FD = LGMAXD THEN
                               GO TO LGO;
                               FM ← FM - DLG;
                          END;
                          LGO: IF ONC THEN GO TO FINLG;
                     END;
                     FINLG:
                END;
COMMENT LABEL LATITUDE ON BOTH SIDES OF ALL BLOCKS;
                FOR SD ← (IMAX) × XSIZE, 0 DO
                BEGIN
COMMENT  FIND MINIMUM LATITUDE LOCATION;
                     MINLAT ← MAXLAT ← FALSE;
                     IF BLOCK = 0 THEN
                     BEGIN
                          FOR LL ← DLT STEP DLT UNTIL 60 DO IF
                          ENTIER (ZLTM) / LL ≤ 1 THEN
                          BEGIN
                               LTMINM ← LL + DLT;
                               LTMIND ← ZLTD;
                               DIFF ← LTMINM - ZLTM;
                               IF LTMINM ≥ 60 THEN
                               BEGIN
                                    LTMINM ← LTMINM - 60;
                                    LTMIND ← LTMIND + 1;
                               END;
                               GO TO FXT;
                          END;
```

```
COMMENT 1849.8=METERS/MINUTE OF LATITUDE FOR CENTRAL VIRGINIA;          CNTR0315
                         FXT: DIFF ← DIFF × 1849.8 / DELY × YSIZE;  CNTR0316
                         TMIN ← DIFF;                                CNTR0317
                    END;                                             CNTR0318
COMMENT 1849.8=METERS/MINUTE OF LATITUDE FOR CENTRAL VIRGINIA;       CNTR0319
                    DT ← DLT × 1849.8 / DELY × YSIZE;                CNTR0320
        %  DRAW LATITUDE SCALE LINE                                  CNTR0321
                    PLOT (SD, 0, 3);                                 CNTR0322
                    PLOT (SD, YSIZE × (JMAX), 2);                    CNTR0323
                    IF TMIN - DT > 0 THEN                            CNTR0324
                    BEGIN                                            CNTR0325
                         TMIN ← TMIN - DT;                           CNTR0326
                         LTMINM ← LTMINM - DLT;                      CNTR0327
                         IF LTMINM < 0 THEN                          CNTR0328
                         BEGIN                                       CNTR0329
                              LTMIND ← LTMIND - 1;                   CNTR0330
                              LTMINM ← LTMINM + 60;                  CNTR0331
                         END;                                        CNTR0332
                    END;                                             CNTR0333
                    NTIME ← 0;                                       CNTR0334
                    FOR LL ← TMIN STEP DT UNTIL YSIZE × (JMAX) DO    CNTR0335
                    BEGIN                                            CNTR0336
        %  DRAW TICK MARKS ON LATITUDE SCALE                         CNTR0337
                         SYMBOL (SD, LL, .1, DUM, 90, - 8);          CNTR0338
                         NTIME ← NTIME + 1;                          CNTR0339
                         LASTA ← LL;                                 CNTR0340
                    END;                                             CNTR0341
                    FM ← LTMINM;                                     CNTR0342
                    FD ← LTMIND;                                     CNTR0343
                    IF SD = 0 THEN SD ← - 1.2 ELSE SD ← SD + .1;     CNTR0344
                    FOR LL ← 0 STEP 1 UNTIL NTIME - 1 DO             CNTR0345
                    BEGIN                                            CNTR0346
                         IF FD = LATMIND AND FM = LATMINM THEN       CNTR0347
                         BEGIN                                       CNTR0348
                              MINLAT ← TRUE;                         CNTR0349
                              LTMN ← TMIN + DT × LL;                 CNTR0350
                         END;                                        CNTR0351
                         IF FD = LATMAXD AND FM = LATMAXM THEN       CNTR0352
                         BEGIN                                       CNTR0353
                              MAXLAT ← TRUE;                         CNTR0354
                              LTMX ← TMIN + DT × LL;                 CNTR0355
                         END;                                        CNTR0356
        %  LABEL LATITUDE SCALE                                      CNTR0357
        %  DEGREES                                                   CNTR0358
                         NUMBER(SD,TMIN+DT×LL-.06,.12,FD,0,0);       CNTR0359
                                                                     CNTR0360
        %  DEGREE SYMBOL                                             CNTR0361
                         SYMBOL(SD+.7,TMIN+DT×LL+.08,.07,DUM,0,-5);CNTR0362
        %  MINUTES                                                   CNTR0363
                         NUMBER(SD+.4,TMIN+DT×LL-.06,.12,FM,0,0);    CNTR0364
        %  MINUTE SYMBOL                                             CNTR0365
                         SYMBOL (SD + 1.1, TMIN + DT × LL + .08,     CNTR0366
                         .07, DUM, 0, - 8);                          CNTR0367
                         FM ← FM + DLT;                              CNTR0368
                         IF FM ≥ 60 THEN                             CNTR0369
                         BEGIN                                       CNTR0370
                              FM ← FM - 60;                          CNTR0371
                              FD ← FD + 1;                           CNTR0372
                         END;                                        CNTR0373
                    END;                                             CNTR0374
                    LAST ← YSIZE × JMAX - LASTA;                     CNTR0375
               END;                                                  CNTR0376
               LTMINM ← FM;                                          CNTR0377
               LTMIND ← FD;                                          CNTR0378
               TMIN ← DT - LAST + 2 × XSIZE;                         CNTR0379
               IF BLOCK=NSUR AND LTMX=0 THEN                         CNTR0380
               BEGIN                                                 CNTR0381
                    LTMX←YSIZE×JMAX;                                 CNTR0382
                    MAXLAT←TRUE;                                     CNTR0383
               END;                                                  CNTR0384
        %  DRAW INNER BOX                                            CNTR0385
               IF MAXLAT AND MINLAT THEN                             CNTR0386
               BEGIN                                                 CNTR0387
                    PLOT (LGMN, LTMN, 3);                            CNTR0388
                    PLOT (LGMN, LTMX, 2);                            CNTR0389
                    PLOT (LGMX, LTMX, 1);                            CNTR0390
                    PLOT (LGMX, LTMN, 1);                            CNTR0391
                    PLOT (LGMN, LTMN, 1);                            CNTR0392
               END ELSE IF MINLAT AND NOT MAXLAT THEN
```

```
                    BEGIN                                                   CNTR0393
                         PLOT (LGMN, 9, 3);                                 CNTR0394
                         PLOT (LGMN, LTMN, 2);                              CNTR0395
                         PLOT (LGMX, LTMN, 1);                              CNTR0396
                         PLOT (LGMX, 9, 1);                                 CNTR0397
                         BOX ← TRUE;                                        CNTR0398
                    END ELSE IF MAXLAT AND NOT MINLAT THEN                  CNTR0399
                    BEGIN                                                   CNTR0400
                         PLOT (LGMN, 0, 3);                                 CNTR0401
                         PLOT (LGMN, LTMX, 2);                              CNTR0402
                         PLOT (LGMX, LTMX, 1);                              CNTR0403
                         PLOT (LGMX, 0, 1);                                 CNTR0404
                         BOX ← FALSE;                                       CNTR0405
                    END ELSE IF BOX THEN                                    CNTR0406
                    BEGIN                                                   CNTR0407
                         PLOT (LGMN, 9, 3);                                 CNTR0408
                         PLOT (LGMN, 0, 2);                                 CNTR0409
                         PLOT (LGMX, 9, 3);                                 CNTR0410
                         PLOT (LGMX, 0, 2);                                 CNTR0411
                    END;                                                    CNTR0412
                         %********************                             CNTR0413
COMMENT SET UP INDEX PARAMETERS;                                            CNTR0414
                    J ← 1;                                                  CNTR0415
                    JA [0] ← JB [0] ← JA [2] ← JA [3] ← IA [1] ← IA [2]     CNTR0416
                    ← IB [2] ← IA [4] ← IA [0] ← J;                         CNTR0417
                    JA[1]←JB[1]←JB[2]←JB[3]←JB[4]←JMAX-J-1;                CNTR0418
                    IB[0]←IB[1]←IA[3]←IB[3]←IB[4]←IMAX-J-1;                CNTR0419
                    JA [4] ← J + 1;                                         CNTR0420
COMMENT READ IN Z ARRAY;                                                    CNTR0421
                    N ← IMAX + 1;                                           CNTR0422
                    FORJ JMAX DO IF ZFILE=1 THEN                            CNTR0423
                    READ(CR,FMT07,FORI IMAX DO Z[J,I])[EOFC] ELSE           CNTR0424
                    READ(TAPED,FMT07,FORI IMAX DO Z[J,I])[EOFC];            CNTR0425
                    ZMAX ← ZMIN ← Z [JA [0], IA [0]];                       CNTR0426
                    N ← JB [4] + 1;                                         CNTR0427
                    K ← IB [4] + 1;                                         CNTR0428
                    FOR J ← JA [0] STEP 1 UNTIL N DO FOR I ← IA [0]         CNTR0429
                    STEP 1 UNTIL K DO IF Z [J, I] < ZMIN THEN ZMIN ← Z [CNTR0430
                    J, I] ELSE IF Z [J, I] > ZMAX THEN ZMAX ← Z [J, I]; CNTR0431
COMMENT DETERMINE ADJUST, IF NOT SPECIFIED;                                 CNTR0432
                    IF ADJST THEN                                           CNTR0433
                    BEGIN                                                   CNTR0434
                         A ← ABS (ZMAX);                                    CNTR0435
                         B ← ABS (ZMIN);                                    CNTR0436
                         IF A < B THEN A ← B;                               CNTR0437
                         I ← 0.4343 × LN (A);                               CNTR0438
                         ADJUST ← 10 * (I - 9);                             CNTR0439
                         ADJUST2 ← ADJUST × P5                              CNTR0440
                    END ELSE IF ADJUST2 = 0 THEN ADJUST2 ← ADJUST × P5; CNTR0441
                    IF CARRA THEN CVP ← 0 ELSE                              CNTR0442
                    BEGIN                                                   CNTR0443
                         IF ZMIN > MINCV THEN A ← ZMIN ELSE A ← MINCV;      CNTR0444
                         CVAL←ENTIER(ABS(A/DELCV))×DELCV×SIGN(A);           CNTR0445
                         IF ABS (CVAL - A) ≠ 0 AND ABS (CVAL - A) <         CNTR0446
                         ADJUST THEN CVAL ← CVAL + DELCV;                   CNTR0447
                         IF ZMAX<MAXCV THEN ENDCV←ZMAX ELSE ENDCV←MAXCV CNTR0448
                    END;                                                    CNTR0449
                    EOF: RESETT (CODE [0]);                                 CNTR0450
                    GO TO QUIT;                                             CNTR0451
                    EOFE: WRITE(LP,<"TRYING TO READ ZDATA CARD, BUT NO "CNTR0452
                              "MORE CARDS IN FILE">);                       CNTR0453
                    GO TO EOJE;                                             CNTR0454
                    EOFA: WRITE(LP,<"TRYING TO READ IN CONTOUR LEVELS, "CNTR0455
                              "BUT NO MORE CARDS IN FILE">);                CNTR0456
                    GO TO EOJE;                                             CNTR0457
                    EOFC: WRITE(LP,<"END-OF-FILE OCCURRED WHILE READING"CNTR0458
                              " GRID">);                                    CNTR0459
                    GO TO EOJE;                                             CNTR0460
        COMMENT END OF Z DATA INPUT BLOCK;                                  CNTR0461
                    QUIT:                                                   CNTR0462
               END;                                                         CNTR0463
                         %********************                             CNTR0464
               FORI 127 DO PCODE [I] ← 0;                                   CNTR0465
               IF CPI > 0 THEN                                              CNTR0466
               BEGIN                                                        CNTR0467
                    INTEGER I, J, I1, J1, JQ2;                              CNTR0468
                    LABEL POLEIT, ENDLOOP;                                  CNTR0469
% SETS PCODE TO SUPPRESS SECONDARY CONTOURS IN HIGH GRADIENT BLOCKS         CNTR0470
                         %********************                             CNTR0471
```

```
                    STREAM PROCEDURE SETP (A, B);
                    VALUE B;
                    BEGIN
                         DI ← A;
                         SKIP B DB;
                         DS ← SET
                    END;
                         %********************
%% NOTE J AND I REVERSED FROM NORMAL INDEXING CONVENTION
                    FOR I ← JA [0] STEP 1 UNTIL JB [4] DO
                    BEGIN
                         I1 ← I + 1;
                         JQ2 ← I × 2;
                         FOR J ← IA [0] STEP 1 UNTIL IB [4] DO
                         BEGIN
                              J1 ← J + 1;
                              IF ABS ((Z [I, J] - Z [I1, J]) / LX) >
                              CPI THEN GO TO POLEIT;
                              IF ABS ((Z [I, J1] - Z [I1, J1]) / LX) >
                              CPI THEN GO TO POLEIT;
                              IF ABS ((Z [I, J] - Z [I, J1]) / LY) >
                              CPI THEN GO TO POLEIT;
                              IF ABS ((Z [I1, J] - Z [I1, J1]) / LY) >
                              CPI THEN GO TO POLEIT;
                              IF ABS ((Z [I, J] - Z [I1, J1]) / LD) >
                              CPI THEN GO TO POLEIT;
                              IF ABS ((Z [I, J1] - Z [I1, J]) / LD) >
                              CPI THEN GO TO POLEIT;
                              GO TO ENDLOOP;
                              POLEIT: SETP (PCODE [JQ2], I);
                              ENDLOOP:
                         END
                    END;
               END;
                         %********************
COMMENT ITERATIVE CONTROL FOR EACH CONTOUR VALUE;
               PRIMF ← TRUE;
               WHILE (CARRA AND CVP≤NCVAL) OR (NOT CARRA AND CVAL≤ENDCV
               AND (CVAL=ENDCV OR ABS(ENDCV-CVAL)>ADJUST)) DO
               BEGIN
                    IF CARRA THEN
                    BEGIN
                         CVAL ← CVALA [CVP];
                         IF CVAL<ZMIN OR CVAL>ZMAX THEN GO TO SKIPCON
                    END ELSE IF PRIMC > 0 THEN
                    BEGIN
                         TEMP ← ABS (CVAL / PRIMC);
                         TEMP ← ENTIER (ABS (ENTIER (TEMP + 0.0005) -
                         TEMP) × 1000);
                         IF TEMP=0 THEN PRIMF←TRUE ELSE PRIMF←FALSE
                    END;
COMMENT CONTOUR TRACING AND DRAWING SUB-BLOCK, LEVEL 3;
                         %********************
                    BEGIN
COMMENT PARAMETER DESCRIPTIONS.
        Z[,]: GRID VALUES.
        IQ,JQ: GRID BLOCK INDICES.
        SIDE: (=0 TO 3) ENTRY SIDE OF CONTOUR:
                              SIDE=2
             Z[JQ+1,IQ]+  Z[JQ+1,IQ+1]+
        SIDE=1                          SIDE=3
              Z[JQ,IQ]+   Z[JQ,IQ+1] +
                              SIDE=0
        SGRID: (=TRUE) START OF CONTOUR SCAN
        CVAL: CURRENT CONTOUR VALUE.
        CI,CJ[]:  ARRAY OF X-Y POINTS DEFINING CONTOUR LINE.
        POINTS: NUMBER OF POINTS IN CI,CJ.  POINTS TO NEXT ELEMENT
             TO BE FILLED.
        NUMX,NUMY: NUMBER OF SUB-GRID BLOCKS WHEN INTERPOLATING IN X & Y
        DX: SUB-GRID X INCREMENT (=1/NUMX)
        DY: SUB-GRID Y INCREMENT (=1/NUMY)      ;
                         BOOLEAN TERM,FTERM,TRM,SGRID;
                         LABEL BUSTED, RECASE;
                         ARRAY W[0:3],CJ,CI[0:1022];
                         INTEGER IQ,JQ,JQB,MODE,JS,IS,SS,SIDE,I1,I2,J1,
                                 J2,CONL,SL,JQ2;
                         DEFINE Z1 = W [1] #,
                         Z2 = W [0] #,
                         Z3 = W [2] #,
                         Z4 = W [3] #;
```

```
%*********************
PROCEDURE PLOTIT;
BEGIN
     INTEGER I,K,CPT,PT,IMIN,JMIN;
     REAL YMIN,DCI,DCJ,MINI,MAXI,MINJ,MAXJ;
     LABEL NOLABLE,LBB1,LBB;
%*********************
     PROCEDURE LABLE (CPT, PT, LBB1);
     INTEGER CPT, PT;
     LABEL LBB1;
     BEGIN
          LABEL LB,LB1,LB2;
          REAL DIF,THETA,SP,LSP,CJPT;
          BOOLEAN CHK;
          INTEGER DIGIT, I;
          DIGIT ← 0;
          CHK ← FALSE;
          FOR I ← 1 STEP 1 WHILE DIGIT = 0 DO
          IF ENTIER (ABS (CVAL) / 10 * I) = 0
          THEN DIGIT ← I + 1;
          IF CVAL < 0 THEN DIGIT ← DIGIT + 1;
          IF DIGIT<6 THEN LSP←(6-DIGIT)/10;
          IF NDEC>0 THEN DIGIT←DIGIT+NDEC+1;
          SP ← DIGIT × .1;
LB:       FOR I←1 STEP 1 WHILE DIF≤SP DO
          BEGIN
               IF CPT+I≥1022 THEN GO TO LB1;
               DIF←ABS(CI[CPT]-CI[CPT+I]);
               PT ← I;
          END;
          PT ← CPT + PT;
          IF ABS (CJ [CPT] - CJ [PT]) > .3
          THEN GO TO LB1;
          THETA ← 0;
          IF PT>POINTS THEN
LB1:      BEGIN
               DIF ← 0;
               FOR I←1 STEP 1 WHILE DIF≤.3 DO
               BEGIN
                    IF CPT + I ≥ 1022 THEN GO
                    TO LB2;
                    DIF ← ABS (CJ [CPT] - CJ [
                    CPT + I]);
                    PT ← I;
               END;
               PT ← CPT + PT;
               IF PT>POINTS THEN
     LB2:      BEGIN
                    IF CHK THEN GO TO LBB1;
                    CPT ← 1;
                    DIF ← 0;
                    CHK ← TRUE;
                    GO TO LB;
               END;
               IF CJ [CPT] > CJ [PT] THEN CJPT
               ← CJ [PT] ELSE CJPT ← CJ [CPT];
          END ELSE CJPT←(CJ[CPT]+CJ[PT])/2;
% LABEL CONTOUR LEVEL
          IF CI [CPT] < CI [PT] THEN NUMBER (
          CI [CPT] - LSP, CJPT, .10, CVAL,
          THETA, NDEC) ELSE NUMBER (CI [PT] -
          LSP, CJPT, .10, CVAL, THETA, NDEC);
     END;
%*********************
     PROCEDURE MINMAX (ARR, NPT, MIN, MAX, K);
     ARRAY ARR [0];
     REAL MIN, MAX;
     INTEGER NPT, K;
     BEGIN
          INTEGER I;
          MIN ← MAX ← ARR [1];
          FOR I ← 2 STEP 1 UNTIL NPT DO
          BEGIN
               IF MIN > ARR [I] THEN
               BEGIN
                    MIN ← ARR [I];
                    K ← I;
               END ELSE IF MAX < ARR [I] THEN
               MAX ← ARR [I];
          END;
```

```
                    END;
                %*********************
                    FOR I ← POINTS STEP - 1 UNTIL 1 DO
                    BEGIN
                         CI [I] ← CI [I - 1];
                         CJ [I] ← CJ [I - 1];
                    END;
                    DCI ← 1.0 / XSIZE;
                    DCJ ← 1.0 / YSIZE;
%  SCALE ARRAYS
                    SCALES (CI, POINTS, 0, DCI, 1);
                    SCALES (CJ, POINTS, 0, DCJ, 1);
                    IF CONL≠1 THEN
          LBB1:     BEGIN
                         IF ENTIER (CVAL / 10) = CVAL / 10
                         THEN FOR I ← 1, 2 DO LYNE (CI, CJ,
                         POINTS, 1);
                         LYNE (CI, CJ, POINTS, 1);
                         GO TO NOLABLE;
                    END;
                    MINMAX (CI, POINTS, MINI, MAXI, IMIN);
                    MINMAX (CJ, POINTS, MINJ, MAXJ, JMIN);
                    IF (CI [1] ≠ CI [POINTS] OR CJ [1] ≠ CJ [
                    POINTS]) THEN IF POINTS ≠ 1022 AND (MAXI
                    - MINI) < .5 AND (MAXJ - MINJ) < .5 THEN
                    GO TO LBB1 ELSE CPT ← ENTIER ((POINTS + 1
                    ) / 2) ELSE
                    BEGIN
                         IF CI [1] = CI [POINTS] AND CJ [1] =
                         CJ [POINTS] THEN CPT ← JMIN;
                         IF CPT ≥ 1000 THEN CPT ← 900;
                    END;
                    LABLE (CPT, PT, LBB1);
                    IF CPT = 1 THEN GO TO LBB;
%  DRAW  CONTOUR TO ONE SIDE OF LABEL
                    IF ENTIER (CVAL / 10) = CVAL / 10 THEN
                    FOR I ← 1, 2 DO LYNE (CI, CJ, CPT, 1);
                    LYNE (CI, CJ, CPT, 1);
                    LBB: CPT ← POINTS - PT + 1;
                    FOR I ← 1 STEP 1 UNTIL POINTS - PT + 1 DO
                    BEGIN
                         CI [I] ← CI [I + PT - 1];
                         CJ [I] ← CJ [I + PT - 1];
                    END;
%  DRAW CONTOUR TO THE OTHER SIDE OF LABEL
                    IF ENTIER (CVAL / 10) = CVAL / 10 THEN
                    FOR I ← 1, 2 DO LYNE (CI, CJ, CPT, 1);
                    LYNE (CI, CJ, CPT, 1);
                    NOLABLE: POINTS ← 0;
                END;
                %*********************
                STREAM PROCEDURE SCLEAR (A);
                BEGIN
                    DI ← A;
                    4 (32 (32 (SKIP DB;
                    DS ← 5 RESET)))
                END;
                %*********************
                INTEGER STREAM PROCEDURE GETCHAR (A, C);
                VALUE C;
                BEGIN
                    SI ← A;
                    SI ← SI + C;
                    DI ← LOC GETCHAR;
                    DS ← 7 LIT "0";
                    DS ← CHR
                END;
                %*********************
                STREAM PROCEDURE LOADCHR (A, C, V);
                VALUE C, V;
                BEGIN
                    DI ← A;
                    DI ← DI + C;
                    SI ← LOC V;
                    SI ← SI + 7;
                    DS ← CHR
                END;
                %*********************
                PROCEDURE RIGHT (LL, M, NN);
                INTEGER LL, M, NN;
```

```
BEGIN                                                          CNTR0712
     BOOLEAN RIGH;                                             CNTR0713
     CASE SIDE OF                                              CNTR0714
     BEGIN                                                     CNTR0715
          RIGH ← NOT BOOLEAN (GETCHAR (CODE [                  CNTR0716
          JQ8], IQ) . [47 : 1]);                               CNTR0717
          RIGH ← NOT BOOLEAN (GETCHAR (CODE [                  CNTR0718
          JQ8], IQ) . [44 : 1]);                               CNTR0719
          RIGH ← NOT BOOLEAN (GETCHAR (CODE [                  CNTR0720
          JQ8], IQ) . [45 : 1]);                               CNTR0721
          RIGH ← NOT BOOLEAN (GETCHAR (CODE [                  CNTR0722
          JQ8], IQ) . [46 : 1]);                               CNTR0723
     END;                                                      CNTR0724
     IF RIGH THEN                                              CNTR0725
     BEGIN                                                     CNTR0726
          LL ← 3;                                              CNTR0727
          M ← - 1;                                             CNTR0728
          NN ← 1                                               CNTR0729
     END ELSE                                                  CNTR0730
     BEGIN                                                     CNTR0731
          LL ← 1;                                              CNTR0732
          M ← 1;                                               CNTR0733
          NN ← 3                                               CNTR0734
     END;                                                      CNTR0735
END;                                                           CNTR0736
%********************                                          CNTR0737
PROCEDURE SET;                                                 CNTR0738
BEGIN                                                          CNTR0739
     INTEGER J, I, K;                                          CNTR0740
     K ← GETCHAR (CODE [JQ8], IQ);                             CNTR0741
     IF ABS (SL - SIDE) = 2 OR K . [43 : 5] ≠                  CNTR0742
     0 THEN J ← 16 ELSE                                        CNTR0743
     BEGIN                                                     CNTR0744
          J ← 0;                                               CNTR0745
          FOR I←0 STEP 1 UNTIL 3 DO IF W[I]>                   CNTR0746
          CVAL THEN J←J+1 ELSE J←J-1;                          CNTR0747
          IF J ≠ 0 THEN J ← 16 ELSE                            CNTR0748
          BEGIN                                                CNTR0749
               FOR I ← SL, SIDE DO CASE I OF                   CNTR0750
               BEGIN                                           CNTR0751
                    J ← J + 8;                                 CNTR0752
                    J ← J + 4;                                 CNTR0753
                    J ← J + 2;                                 CNTR0754
                    J ← J + 1;                                 CNTR0755
               END                                             CNTR0756
          END                                                  CNTR0757
     END;                                                      CNTR0758
     J ← K & J [43 : 43 : 5];                                  CNTR0759
     LOADCHR (CODE [JQ8], IQ, J);                              CNTR0760
END;                                                           CNTR0761
%********************                                          CNTR0762
BOOLEAN STREAM PROCEDURE CHK2 (A, B);                          CNTR0763
VALUE B;                                                       CNTR0764
BEGIN                                                          CNTR0765
     DI ← LOC CHK2;                                            CNTR0766
     DS ← 7 LIT "0";                                           CNTR0767
     SI ← A;                                                   CNTR0768
     SI ← SI + B;                                              CNTR0769
     IF SB THEN DS←LIT "1" ELSE DS←LIT "0";                    CNTR0770
END;                                                           CNTR0771
%********************                                          CNTR0772
BOOLEAN PROCEDURE CHECK;                                       CNTR0773
BEGIN                                                          CNTR0774
     INTEGER I;                                                CNTR0775
     I ← GETCHAR (CODE [JQ8], IS);                             CNTR0776
     IF I>15 THEN CHECK←FALSE ELSE CASE SIDE OF                CNTR0777
     BEGIN                                                     CNTR0778
          CHECK ← NOT BOOLEAN (I . [44 : 1]);                  CNTR0779
          CHECK ← NOT BOOLEAN (I . [45 : 1]);                  CNTR0780
          CHECK ← NOT BOOLEAN (I . [46 : 1]);                  CNTR0781
          CHECK ← NOT BOOLEAN (I . [47 : 1]);                  CNTR0782
     END;                                                      CNTR0783
END;                                                           CNTR0784
%********************                                          CNTR0785
PROCEDURE ZSHIFT;                                              CNTR0786
BEGIN                                                          CNTR0787
     CASE SIDE OF                                              CNTR0788
     BEGIN                                                     CNTR0789
          BEGIN                                                CNTR0790
               Z1 ← Z3;                                        CNTR0791
```

```
                                        Z2 ← Z4;
                                        Z3 ← AJUST (Z [JQ + 1, IQ]);
                                        Z4 ← AJUST (Z [JQ + 1, IQ + 1])
                                   END;
                                   BEGIN
                                        Z1 ← Z2;
                                        Z3 ← Z4;
                                        Z2 ← AJUST (Z [JQ, IQ + 1]);
                                        Z4 ← AJUST (Z [JQ + 1, IQ + 1])
                                   END;
                                   BEGIN
                                        Z3 ← Z1;
                                        Z4 ← Z2;
                                        Z1 ← AJUST (Z [JQ, IQ]);
                                        Z2 ← AJUST (Z [JQ, IQ + 1])
                                   END;
                                   BEGIN
                                        Z2 ← Z1;
                                        Z4 ← Z3;
                                        Z1 ← AJUST (Z [JQ, IQ]);
                                        Z3 ← AJUST (Z [JQ + 1, IQ])
                                   END;
                              END
                         END;
                         %********************
                         PROCEDURE TRACE;
                         BEGIN
% DETERMINES EXIT SIDE FOR FOLLOWING SECONDARY CONTOURS THROUGH
% POLED BLOCKS
                              INTEGER LL, M, NN, I, K, L, N;
                              LABEL FND2;
                              DEFINE J = SIDE #;
                              IF NOT SGRID THEN ZSHIFT ELSE SGRID←FALSE;
                              RIGHT (LL, M, NN);
                              L ← SIDE + LL;
                              N ← SIDE + NN;
                              FOR I ← L STEP M UNTIL N DO
                              BEGIN
                                   J ← I . [46 : 2];
                                   K ← (I + 1) . [46 : 2];
                                   IF (W [J] - CVAL) . [1 : 1] ≠ (W [K]
                                   - CVAL) . [1 : 1] THEN GO TO FND2
                              END;
                              FND2:
                         END;
                         %********************
                         BOOLEAN STREAM PROCEDURE POLED (A, B);
                         VALUE B;
                         BEGIN
                              DI ← LOC POLED;
                              DS ← 7 LIT "0";
                              SI ← A;
                              SKIP B SB;
                              IF SB THEN DS←LIT "1" ELSE DS←LIT "0";
                         END;
                         %********************
                         PROCEDURE SUBGRID;
                         BEGIN
COMMENT THIS ROUTINE INTERPOLATES AND TRACES COUTOUR THROUGH A GRID BOX.
        THE BASIC GRID IS DIVIDED INTO A SUB-GRID WHOSE Z VALUES ARE
        DETERMINED (AS NEEDED) BY PSEUDO-SPLINE INTERPOLATION.
        THE CONTOUR IS THEN TRACED THROUGH THE SUB-GRID SIDES WITH
        X-Y VALUES DETERMINED BY LINEAR INTERPOLATION OF THE SUB-GRID
        VALUES.
        THE EXIT SIDE IS RETURNED TO THE MAIN TRACING ROUTINE ALONG
        WITH RE-ENTRY DATA.
        LOCAL PARAMETER DESCRIPTIONS:
        FIELD[,]: IMAGE OF BASIC GRID BLOCK.
        IS,JS:  SUB-GRID INDICES.
        A[]: PSEUDO-SPLINE COEFFICIENTS.
        POINTA,POINTB,Z1,Z2,Z3,Z4: TEMPORARY Z SUB-GRID DATA.
        SSIDE:  SUB-GRID SIDE INDEX. SAME CONVENTION AS SIDE
                              OWN ARRAY A[0:3,0:8],W[0:3];
                              REAL POINTA, POINTB, DELTA, X0, Y;
                              INTEGER I,LL,M,NN,K,SSIDE,NUMBER,L,N;
                              OWN INTEGER IS, JS;
                              DEFINE Z1 = W [1] #,
                              Z2 = W [0] #,
                              Z3 = W [2] #,
                              Z4 = W [3] #,
```

```
                    X = XO #,
                    J = SIDE #;
                    FORMAT FMT01("SUBGRID: NO ENTRY"),
                           FMT02("SUBGRID: NO EXIT");
                    LABEL FND1,FND2;
                %********************
                    PROCEDURE COFFFC (IQ, JQ);
                    INTEGER IQ, JQ;
                    BEGIN
                         INTEGER JJ, J, II, I, K;
                         DEFINE W11 = Z [JJ, II] #,
                         W12 = Z [JJ, II + 1] #,
                         W13 = Z [JJ, II + 2] #,
                         W21 = Z [JJ + 1, II] #,
                         W22 = Z [JJ + 1, II + 1] #,
                         W23 = Z [JJ + 1, II + 2] #,
                         W31 = Z [JJ + 2, II] #,
                         W32 = Z [JJ + 2, II + 1] #,
                         W33 = Z [JJ + 2, II + 2] #,
                         P5 = 0.5 #,
                         P4 = 0.25 #;
                         JJ ← JQ - 1;
                         FOR J ← 0, 2 DO
                         BEGIN
                              II ← IQ - 1;
                              FOR I ← 0, 1 DO
                              BEGIN
                                   K ← J + I;
                                   A [K, 8] ← W22;
                                   A[K,7]←P5×(W32-W12);
                                   A[K,6]←P5×(W12+W32)-A[K,8];
                                   A[K,5]←P5×(W23-W21);
                                   A[K,4]←P4×(W11-W13-W31+
                                   W33);
                                   A[K,3]←P4×(W13+W33-W11-
                                   W31)-A[K,5];
                                   A[K,2]←P5×(W21+W23)-A[K,8];
                                   A[K,1]←P4×(W31+W33-W11-W13)
                                   -A[K,7];
                                   A[K,0]←P4×(W31+W33+W11+W13)
                                   -A[K,8]-A[K,6]-A[K,2];
                                   II ← IQ
                              END;
                              JJ ← JQ
                         END;
                    END;
                %********************
                    REAL PROCEDURE ZVAL (X, Y);
                    VALUE X, Y;
                    REAL X, Y;
                    BEGIN
                %********************
                         REAL PROCEDURE ZIT (X, Y, I);
                         VALUE X, Y, I;
                         REAL X, Y;
                         INTEGER I;
                         BEGIN
                              ZIT ← (((A [I, 0] × Y + A [I, 1
                              ]) × Y + A [I, 2]) × X + (A [I,
                              3] × Y + A [I, 4]) × Y + A [I,
                              5]) × X + (A [I, 6] × Y + A [I,
                              7]) × Y + A [I, 8];
                         END;
                %********************
                         REAL DX, DY;
                         DX ← X - 1;
                         DY ← Y - 1;
                         ZVAL ← DX × DY × ZIT (X, Y, 0) - X ×
                         DY × ZIT (DX, Y, 1) - DX × Y × ZIT (
                         X, DY, 2) + X × Y × ZIT (DX, DY, 3);
                    END;
                %********************
                    COEFFC (IQ, JQ);
                    IF SGRID THEN
                    BEGIN
COMMENT THIS SECTION DETERMINES ENTRY PT. AT BEGINNING OF CONTOUR SCAN;
                         CASE SIDE OF
                         BEGIN
                              BEGIN
                                   CJ [0] ← JQ;
                                   POINTA ← Z [JQ, IQ];
```

```
                    NUMBER ← NUMX                    CNTR0953
               END;                                  CNTR0954
               BEGIN                                 CNTR0955
                    CI [0] ← IQ;                     CNTR0956
                    POINTA ← Z [JQ, IQ];             CNTR0957
                    NUMBER ← NUMY                    CNTR0958
               END;                                  CNTR0959
               BEGIN                                 CNTR0960
                    CJ [0] ← JQ + 1;                 CNTR0961
                    POINTA ← Z [JQ + 1, IQ];         CNTR0962
                    NUMBER ← NUMX                    CNTR0963
               END;                                  CNTR0964
               BEGIN                                 CNTR0965
                    CI [0] ← IQ + 1;                 CNTR0966
                    POINTA ← Z [JQ, IQ + 1];         CNTR0967
                    NUMBER ← NUMY                    CNTR0968
               END;                                  CNTR0969
          END;                                       CNTR0970
          POINTA ← AJUST (POINTA);                   CNTR0971
          FOR I ← 1 STEP 1 UNTIL NUMBER DO           CNTR0972
          BEGIN                                      CNTR0973
               IS ← I - 1;                           CNTR0974
               CASE SIDE OF                          CNTR0975
               BEGIN                                 CNTR0976
                    POINTB ← ZVAL (I × DX, 0);       CNTR0977
                    POINTB ← ZVAL (0, I × DY);       CNTR0978
                    POINTB ← ZVAL (I × DX, 1);       CNTR0979
                    POINTB ← ZVAL (1, I × DY);       CNTR0980
               END;                                  CNTR0981
               POINTB ← AJUST (POINTB);              CNTR0982
               IF (POINTA - CVAL) . [1 : 1] ≠ (CNTR0983
               POINTB - CVAL) . [1 : 1] THEN         CNTR0984
               GO TO FND1;                           CNTR0985
               POINTA ← POINTB                       CNTR0986
          END;                                       CNTR0987
COMMENT IF FALL OUT OF LOOP THEN BUST OCCURRED;      CNTR0988
          WRITE (LP, FMT01);                         CNTR0989
          GO TO BUSTED;                              CNTR0990
          FND1: DELTA ← ((CVAL - POINTA) / (         CNTR0991
          POINTB - POINTA) + IS);                    CNTR0992
          CASE SIDE OF                               CNTR0993
          BEGIN                                      CNTR0994
               BEGIN                                 CNTR0995
                    Z1 ← POINTA;                     CNTR0996
                    Z2 ← POINTB                      CNTR0997
               END;                                  CNTR0998
               BEGIN                                 CNTR0999
                    Z1 ← POINTA;                     CNTR1000
                    Z3 ← POINTB                      CNTR1001
               END;                                  CNTR1002
               BEGIN                                 CNTR1003
                    Z3 ← POINTA;                     CNTR1004
                    Z4 ← POINTB                      CNTR1005
               END;                                  CNTR1006
               BEGIN                                 CNTR1007
                    Z2 ← POINTA;                     CNTR1008
                    Z4 ← POINTB                      CNTR1009
               END;                                  CNTR1010
          END;                                       CNTR1011
          CASE SIDE . [47 : 1] OF                    CNTR1012
          BEGIN                                      CNTR1013
               BEGIN                                 CNTR1014
                    CI [0] ← IQ + DELTA × DX;        CNTR1015
                    JS ← 0                           CNTR1016
               END;                                  CNTR1017
               BEGIN                                 CNTR1018
                    CJ [0] ← JQ + DELTA × DY;        CNTR1019
                    JS ← IS;                         CNTR1020
               END;                                  CNTR1021
          END;                                       CNTR1022
          POINTS ← 1;                                CNTR1023
          SGRID ← FALSE;                             CNTR1024
     END;                                            CNTR1025
     CASE SIDE OF                                    CNTR1026
     BEGIN                                           CNTR1027
          JS ← 0;                                    CNTR1028
          IS ← 0;                                    CNTR1029
          JS ← NUMY - 1;                             CNTR1030
          IS ← NUMX - 1                              CNTR1031
     END;                                            CNTR1032
```

```
                    SSIDE ← SIDE;                                   CNTR1033
COMMENT SET RIGHT OR LEFT TREND THROUGHOUT BLOCKS;                  CNTR1034
                    RIGHT (LL, M, NN);                              CNTR1035
COMMENT START SUB-GRID TRACE ITERATION;                             CNTR1036
                    WHILE IS ≥ 0 AND JS ≥ 0 AND IS < NUMX AND       CNTR1037
                    JS < NUMY DO                                    CNTR1038
                    BEGIN                                           CNTR1039
COMMENT COMPUTE REMAINING Z VALUES OF SUB-GRID BLOCK;               CNTR1040
                         XO ← IS × DX;                              CNTR1041
                         Y ← JS × DY;                               CNTR1042
                         CASE SSIDE OF                              CNTR1043
                         BEGIN                                      CNTR1044
                              BEGIN                                 CNTR1045
                                   Z3←AJUST(ZVAL(XO,Y+DY));         CNTR1046
                                   Z4←AJUST(ZVAL(XO+DX,Y+DY));      CNTR1047
                              END;                                  CNTR1048
                              BEGIN                                 CNTR1049
                                   Z2←AJUST(ZVAL(XO+DX,Y));         CNTR1050
                                   Z4←AJUST(ZVAL(XO+DX,Y+DY));      CNTR1051
                              END;                                  CNTR1052
                              BEGIN                                 CNTR1053
                                   Z1 ← AJUST (ZVAL (XO, Y));       CNTR1054
                                   Z2←AJUST(ZVAL(XO+DX,Y));         CNTR1055
                              END;                                  CNTR1056
                              BEGIN                                 CNTR1057
                                   Z1 ← AJUST (ZVAL (XO, Y));       CNTR1058
                                   Z3←AJUST(ZVAL(XO,Y+DY));         CNTR1059
                              END;                                  CNTR1060
                         END;                                       CNTR1061
                         L ← SSIDE + LL;                            CNTR1062
                         N ← SSIDE + NN;                            CNTR1063
                         FOR I ← L STEP M UNTIL N DO                CNTR1064
                         BEGIN                                      CNTR1065
                              J ← I . [46 : 2];                     CNTR1066
                              K ← (I + 1) . [46 : 2];               CNTR1067
                              IF (W [J] - CVAL) . [1 : 1] ≠ (       CNTR1068
                              W [K] - CVAL) . [1 : 1] THEN GO       CNTR1069
                              TO FND2                               CNTR1070
                         END;                                       CNTR1071
COMMENT IF FALL OUT THEN FAILURE TO FIND EXIT;                      CNTR1072
                         WRITE (LP, FMT02);                         CNTR1073
                         GO TO BUSTED;                              CNTR1074
                         FND2: IF J = 0 OR J = 3 THEN DELTA ←       CNTR1075
                         (W [K] - CVAL) / (W [K] - W [J])           CNTR1076
                         ELSE DELTA ← (W [J] - CVAL) / (W [J]       CNTR1077
                         - W [K]);                                  CNTR1078
                         CASE J . [47 : 1] OF                       CNTR1079
                         BEGIN                                      CNTR1080
                              CI[POINTS]←IQ+(DELTA+IS)×DX;          CNTR1081
                              CJ[POINTS]←JQ+(DELTA+JS)×DY           CNTR1082
                         END;                                       CNTR1083
                         CASE J OF                                  CNTR1084
                         BEGIN                                      CNTR1085
                              BEGIN                                 CNTR1086
                                   CJ[POINTS]←JQ+JS×DY;             CNTR1087
                                   JS ← JS - 1;                     CNTR1088
                                   Z3 ← Z1;                         CNTR1089
                                   Z4 ← Z2                          CNTR1090
                              END;                                  CNTR1091
                              BEGIN                                 CNTR1092
                                   CI[POINTS]←IQ+IS×DX;             CNTR1093
                                   IS ← IS - 1;                     CNTR1094
                                   Z2 ← Z1;                         CNTR1095
                                   Z4 ← Z3                          CNTR1096
                              END;                                  CNTR1097
                              BEGIN                                 CNTR1098
                                   CJ[POINTS]←JQ+(JS+1)×DY;         CNTR1099
                                   JS ← JS + 1;                     CNTR1100
                                   Z1 ← Z3;                         CNTR1101
                                   Z2 ← Z4                          CNTR1102
                              END;                                  CNTR1103
                              BEGIN                                 CNTR1104
                                   CI[POINTS]←IQ+(IS+1)×DX;         CNTR1105
                                   IS ← IS + 1;                     CNTR1106
                                   Z1 ← Z2;                         CNTR1107
                                   Z3 ← Z4                          CNTR1108
                              END;                                  CNTR1109
                         END;                                       CNTR1110
```

```
                                SSIDE ← (J + 2) . [46 : 2];
                                POINTS ← POINTS + 1;
                                IF POINTS ≥ 1022 THEN
                                BEGIN
                                     PLOTIT;
                                     CONL ← 2
                                END;
                           END;
                      END;
                      %********************
COMMENT CONTOUR SEARCH LOOPS.
        MODE=0 BOTTOM BORDER
        MODE=1 TOP
            =2 LEFT
            =3 RIGHT
            =4 INTERIOR;
                      FOR MODE ← 0 STEP 1 UNTIL 4 DO
                      BEGIN
                           J1 ← IJA [MODE];
                           J2 ← IJA [MODE + 5];
                           I1 ← IJA [MODE + 10];
                           I2 ← IJA [MODE + 15];
                           FOR JS←JA[MODE] STEP 1 UNTIL JB[MODE] DO
                           FOR IS←IA[MODE] STEP 1 UNTIL IB[MODE] DO
                           BEGIN
                                SIDE ← SDE [MODE];
                                JQ8 ← JS × 8;
                                JQ2 ← JS × 2;
                                IF CHECK AND (AJUST (Z [JS + J1, IS
                                + I1]) - CVAL) . [1 : 1] ≠ (AJUST (Z
                                [JS + J2, IS + I2]) - CVAL) . [1 : 1
                                ] THEN
                                BEGIN
                                     SGRID ← FTERM ← TRUE;
                                     TERM ← TRM ← FALSE;
                                     SS ← SIDE;
                                     IQ ← IS;
                                     JQ ← JS;
                                     CONL ← 1;
                                     POINTS ← 0;
                                     DO
                                     BEGIN
                                          SL ← SIDE;
                                          IF SGRID THEN
                                          BEGIN
                                               Z1←AJUST(Z[JQ,IQ]);
                                               Z2←AJUST(Z[JQ,IQ+1]);
                                               Z3←AJUST(Z[JQ+1,IQ]);
                                               Z4←AJUST(Z[JQ+1,IQ+1])
                                          END ELSE ZSHIFT;
                                          IF NOT PRIMF AND POLED (
                                          PCODE [JQ2], IQ) THEN
                                          BEGIN
                                               IF POINTS > 0 THEN
                                               BEGIN
                                                    PLOTIT;
                                                    CONL ← 1
                                               END;
                                               TRACE
                                          END ELSE SUBGRID;
                                          SET;
                                          SIDE←(SIDE+2).[46:2];
                                          RECASE: CASE SIDE OF
                                          BEGIN
                                               BEGIN
                                                    JQ ← JQ + 1;
                                                    JQ8 ← JQ × 8;
                                                    JQ2 ← JQ × 2
                                               END;
                                               IQ ← IQ + 1;
                                               BEGIN
                                                    JQ ← JQ - 1;
                                                    JQ8 ← JQ × 8;
                                                    JQ2 ← JQ × 2
                                               END;
                                               IQ ← IQ - 1;
                                          END;
                                          IF IQ < IA [0] OR IQ > IB [
                                          4] OR JQ < JA [0] OR JQ >
```

```
                                        JB [4] THEN TRM + TRUE          CNTR1190
                                        ELSE TERM + CHK2 (CODE [        CNTR1191
                                        JQ8], IQ)}                      CNTR1192
                                   END UNTIL TRM OR TERM OR (IQ =       CNTR1193
                                   IS AND JQ = JS AND SS = SIDE)}       CNTR1194
                                   IF POINTS > 0 THEN PLOTIT}           CNTR1195
                                   IF (TERM OR TRM) AND FTERM AND       CNTR1196
                                   MODE = 4 THEN                        CNTR1197
                                   BEGIN                                CNTR1198
                                        SIDE+(SS+2).[46:2]}             CNTR1199
                                        IQ + IS}                        CNTR1200
                                        JQ + JS}                        CNTR1201
                                        IF SIDE . [47 : 1] ≠ 0          CNTR1202
                                        THEN                            CNTR1203
                                        BEGIN                           CNTR1204
                                             JQ8 + JQ × 8}              CNTR1205
                                             JQ2 + JQ × 2               CNTR1206
                                        END}                            CNTR1207
                                        CONL + 1}                       CNTR1208
                                        TRM + FTERM + FALSE}            CNTR1209
                                        SGRID + TRUE}                   CNTR1210
                                        GO TO RECASE                    CNTR1211
                                   END}                                 CNTR1212
                              END}                                      CNTR1213
                         END}                                           CNTR1214
                    END}                                                CNTR1215
                    BUSTED: SCLEAR (CODE [0])}                          CNTR1216
               END}                                                     CNTR1217
                    %*******************                                CNTR1218
COMMENT END TRACING AND DRAWING BLOCK}                                  CNTR1219
                    %*******************                                CNTR1220
COMMENT THIS COMPLETES ART WORK ON SUB-GRID. RETURN FOR NEXT GRID}      CNTR1221
               IF CARRA THEN                                            CNTR1222
               BEGIN                                                    CNTR1223
                    SKIPCON: CVP + CVP + 1                              CNTR1224
               END ELSE CVAL + CVAL + DELCV}                            CNTR1225
          END}                                                          CNTR1226
COMMENT  PRINT LEGEND IN RIGHT HAND CORNER OF BOTTOM SECTION}           CNTR1227
          IF BLOCK = 0 THEN                                             CNTR1228
          BEGIN                                                         CNTR1229
               KY + XSIZE × IMAX + 2.0}                                 CNTR1230
               PTSGEN + ISIZE × JSIZE}                                  CNTR1231
               DUM [0] + "LEGEND"}                                      CNTR1232
               SYMBOL (KY, 2.0, .15, DUM, 0, 6)}                        CNTR1233
               PLOT (KY, 1.9, 3)}                                       CNTR1234
               PLOT (KY + .8, 1.9, 2)}                                  CNTR1235
               FILL DUM [*] WITH "SCALE ", "=      "}                   CNTR1236
               SYMBOL (KY, 1.5, .15, DUM, 0, 7)}                        CNTR1237
               NUMBER (KY + .75, 1.5, .15, SCAL, 0, 2)}                 CNTR1238
               FILL DUM [*] WITH " MI./I", "N.     "}                   CNTR1239
               SYMBOL (KY + 1.9, 1.5, .15, DUM, 0, 8)}                  CNTR1240
               FILL DUM [*] WITH "CONTOU", "R INTE", "RVAL ="}          CNTR1241
               SYMBOL (KY, 1.0, .15, DUM, 0, 18)}                       CNTR1242
               NUMBER (KY + 2., 1.0, .15, DELCV, 0, NDEC)}              CNTR1243
               NUMBER (KY, .5, .15, NORPT, 0, 0)}                       CNTR1244
               FILL DUM [*] WITH " ORIGI", "NAL PO", "INTS  "}          CNTR1245
               SYMBOL (KY + .8, .5, .15, DUM, 0, 16)}                   CNTR1246
               NUMBER (KY, 0, .15, PTSGEN, 0, 0)}                       CNTR1247
               FILL DUM [*] WITH " POINT", "S GENE", "RATED "}          CNTR1248
               SYMBOL (KY + .8, 0, .15, DUM, 0, 17)}                    CNTR1249
          END}                                                          CNTR1250
                    %*******************                                CNTR1251
     END}                                                               CNTR1252
END}                                                                    CNTR1253
                    %*******************                                CNTR1254
COMMENT END OF SUB-BLOCK SEGMENT}                                       CNTR1255
     GO TO EOJ}                                                         CNTR1256
     EOJE: WRITE (LP, FMTED)}                                           CNTR1257
     GO TO QUIT}                                                        CNTR1258
     EOJ: WRITE (LP, FMTEN)}                                            CNTR1259
     QUIT: PLOT (XSIZE × JMAX + 5, 0, - 3)}                             CNTR1260
END .                                                                   CNTR1261
```