UNITED STATES (DEPARTMENT OF THE INTERIOR)

GEOLOGICAL SURVEY, [Reports - Open file series]

A General Purpose Contouring System

By

Gerald Ian Evenden

Open-file report 75-317

1975

This report is preliminary and has not been edited or reviewed for conformity with U.S. Geological Survey standards.

Contents

Page

Abstract

Three Decsystem-10 FORTRAN IV programs provide a general purpose system for contouring two-dimensional data. The system can provide both quick or final, publication quality contour maps on either interactive or offline plotting devices. Complete user documentation, with examples, and program listings are presented.

DISCLAIMER.

ALTHOUGH THESE PROGRAMS HAVE BEEN SUBJECTED TO MANY TESTS AND CONSIDERABLE USAGE A WARRANTY ON ACCURACY OR PROPER FUNCTIONING IS NEITHER IMPLIED NOR EXPRESSED.

INTRODUCTION.

THIS SYSTEM OF THREE FORTRAN PROGRAMS PROVIDES A SIMPLE YET EFFECTIVE MEANS OF CONTOURING GRIDS OF TWO DIMENSIONAL DATA. THE GRAPHICAL PHASE OF CONTOURING IS PERFORMED BY PROGRAM CONTUR WHICH DETERMINES THE CONTOUR PATH BY LINEAR INTERPOLATION AND PROVIDES FOR CONTOUR LABELING, AXIS LABELING AND DROPPING OF CONTOURS IN HIGH GRADIENT REGIONS.

LACK OF CONTOUR SMOOTHNESS AND ACCURACY OF LINEAR INTERPOLATION OF DATA WITH CONSIDERABLE HIGH FREQUENCY CONTENT MAY REQUIRE A MORE FINELY INTERPOLATED GRID. PROGRAM SPLN2D PROVIDES THIS INTERPOLATION SERVICE WITH BICUBIC SPLINE METHODS. A THIRD PROGRAM, SPLN2X, IS ALSO PROVIDED TO DETERMINE BICUBIC SPLINE COEFFICIENTS FOR DATA GRIDS LARGER THAN CAN BE HANDLED BY SPLN2D.

ALL INPUT AND OUTPUT DATA GRIDS AND SPLINE COEFFICIENT FILES MUST BE IN A STANDARD FORM (SEE APPENDIX A). ALL PROGRAMS ALSO PROVIDE FOR MISSING DATA IN THE GRID PROVIDED SUCH DATA IS FLAGGED WITH THE COMPUTER SYSTEM MAXIMUM FLOATING POINT NUMBER.

THE FOLLOWING SECTIONS CONTAIN COMPLETE USER DOCUMENTATION REQUIRED TO EXECUTE THE CONTOURING SYSTEM. IT IS ASSUMED THAT THE PROSPECTIVE USER IS FAMILIAR WITH THE FACILITIES OF THE DECSYSTEM-10 COMPUTER AND SIMPLE FORMAT CONSTRUCTS AND NAMELIST DATA ENTRY OF THE FORTRAN LANGUAGE. ANY USER SHOULD NOT EXPECT INSTANT SUCCESS ON FIRST USING THE PROGRAMS BECAUSE OF THE MANY DETAILS INVOLVED. IT IS SUGGESTED THAT FAMILIARIZATION OF THE ROUTINES BE MADE BY PLOTTING ON AN INTERACTIVE DEVICE SUCH AS THE TEKTRONIX 4010 AND USING ONLY THE REQUIRED OPTIONS.

THESE PROGRAMS CAN BE TRANSPORTED TO OTHER COMPUTER SYSTEMS WITH A MODERATE AMOUNT OF REVISION BY AN EXPERIENCED PROGRAMMER. EACH PROGRAM (LISTINGS IN APPENDIXES B THROUGH D) ARE WRITTEN IN DECSYSTEM-10 FORTRAN IV. SINCE ATTENTION WAS GIVEN TO BOTH TIMING AND MEMORY PERFORMANCE ALL THREE PROGRAMS CONTAIN NON-STANDARD FORTRAN CONSTRUCTS IN FILE MANIPULATION AND MULTIPLE ENTRY SUBROUTINES. PROGRAM CONTUR ALSO CONTAINS ADDITIONAL NON-STANDARD CONSTRUCTS INVOLVING BIT MANIPULATION OF FLAG ARRAYS BY "LOGICAL MASKING" STATEMENTS.

SEVERAL ROUTINES IN PROGRAMS SPLN2D AND SPLN2X ARE BASED ON ANDERSON'S (1971) ALGORITHMS. THE ONLY BASIC CHANGES WERE TO PROVIDE FOR SIMPLE MONOTONIC INDEPENDENT VARIABLES AND REDUCTION OF WORK FILES AND I/O OPERATIONS IN SPLN2X.

THE LISTINGS CONTAIN ALL CODE EXCEPT BASIC GRAPHICS ROUTINES AND GMPRD (GENERAL MATRIX PRODUCT) SUBROUTINE. APPENDIX E CONTAINS USER DOCUMENTATION OF THE GRAPHICS SYSTEM EMPLOYED AND GMPRD IS COMMONLY AVAILABLE IN

SCIENTIFIC SUBROUTINE PACKAGES. NOTE THAT SUBROUTINES COMMON TO BOTH SPLN2D AND SPLN2X ARE LISTED ONLY UNDER SPLN2D.

GENERAL.

CONTUR IS A GENERAL PURPOSE ROUTINE FOR CONTOURING TWO DIMENSIONAL DATA. THE ROUTINE EMPLOYS THE CONTOUR TRACING METHOD WHERE A CONTOUR IS FOLLOWED THROUGH THE GRID. THIS ALLOWS CONTOUR LABELING AND MINIMIZES PEN UP-DOWN MOTIONS FOR MECHANICAL PLOTTERS. HOWEVER, IT REQUIRES MORE COMPUTER MEMORY AND TIME THAN METHODS COMPLETELY CONTOURING ONE GRID CELL AT A TIME.

COORDINATES OF THE CONTOUR LINE ARE DETERMINED BY LINEAR INTERPOLATION OF THE CONTOUR THROUGH THE GRID CELL "WALLS". THIS WILL OFTEN PRODUCE ANGULAR CONTOURS FOR DATA GRIDS CONTAINING SIGNIFICANT HIGH FREQUENCY COMPONENTS. ALTHOUGH SUCH ANGULARITY DOES NOT MATTER FOR PRELIMINARY PLOTS, SMOOTHED, PUBLICATION QUALITY MAPS CAN BE OBTAINED BY PREVIOUS INTERPOLATION WITH PROGRAM SPLN2D.

CONTUR CAN ALSO CONTOUR QUADRILATERAL GRIDS WHERE INDIVIDUAL CELL INTERIOR ANGLES DO NOT EXCEED 180 DEGREES. THE INPUT OF SUCH A GRID IS SIGNALLED TO THE PROGRAM BY THE NZ FACTOR OF THE STANDARD FILE BEING EQUAL TO 3. EACH ELEMENT OF THE INPUT GRID MUST THEN CONTAIN X, Y AND Z VALUES. THE SPLINE INTERPOLATION ROUTINES, HOWEVER, CANNOT HANDLE THIS TYPE OF DATA.

EXECUTION OF CONTUR.

PRIOR TO RUNNING CONTUR THE LOGICAL DEVICE "PLOT" MUST BE ASSIGNED TO A DEVICE APPROPRIATE TO THE PLOTTER EMPLOYED. THIS CAN BE DONE BY:

```
ASSIGN TTY PLOT
```

FOR THE TEKTRONIX 4010 OR HEWLETT-PACKARD 7202A AND 7203A, OR BY:

```
MOUNT MTA:PLOT/REELID:PLOTAP/VID:7-TRK/WE
```

FOR THE GERBER 622.

FOR TESTING THE PLOT COMMAND FILE:

```
ASSIGN NULL PLOT
```

CAN BE EMPLOYED.

CONTUR IS EXECUTED BY:

```
RUN CONTUR[333,724]
```

AFTER THE PROGRAM IS LOADED AN ASTERISK IS TYPED ON THE TERMINAL. THE USER RESPONDS WITH THE NAME AND EXTENSION OF A CONTUR COMMAND FILE (IF THE EXTENSION IS OMITTED, ".DAT" IS ASSUMED). AFTER THE CURRENT COMMAND FILE IS EXECUTED THE ROUTINE WILL REQUEST ANOTHER COMMAND FILE. THIS SEQUENCE CAN CONTINUE INDEFINITELY. TO EXIT FROM THE PROGRAM THE USER CAN ENTER "EXIT" AFTER THE ASTERISK IS TYPED.

CONTUR COMMAND FILE.

THE COMMAND FILE IS AN ASCII FILE CONTAINING INFORMATION TO CONTROL FORMATTING OF THE CONTOUR PLOT, TITLING DATA AND NAME OF THE STANDARD TWO-DIMENSIONAL FILE TO BE CONTOURED. BASICALLY, THE FILE CONSISTS OF ONE OR MORE "COMMAND SETS". EACH COMMAND SET CONSISTS OF 4 FIXED FORMAT LINES FOLLOWED BY A " &PARMS" NAMELIST SECTION. THE NAMELIST SECTION MAY BE CONTINUED ON SEVERAL LINES (SEE FORTRAN MANUAL FOR DETAILS OF NAMELIST DATA ENTRY). EXCEPT FOR THE FIRST LINE (INPUT DATA FILE NAME) THE PROGRAM CONTROL FUNCTIONS ARE NOT ALTERED BETWEEN COMMAND SETS UNLESS:

A. NON-BLANK FIELDS ARE ENCOUNTERED IN CONTROL SET LINES 2, 3 OR 4 AND
B. THE KEYWORD IDENTIFIER IS ENTERED IN THE NAMELIST SECTION.

A CONSEQUENCE OF THE FEATURE IS THAT THE INITIAL COMMAND SET CREATES DEFAULT VALUES FOR THE REMAINDER OF THE JOB RUN OR UNTIL THEY ARE EXPLICITLY ALTERED AGAIN.

DETAILS OF COMMAND SET.

LINE 1.

CHARACTERS 1 THOUGH 10 CONTAIN FILE NAME AND EXTENSION OF A STANDARD FORMAT TWO DIMENSIONAL FILE TO BE CONTOURED. IF THE EXTENSION IS ABSENT ".DAT" IS ASSUMED. IF THE FIELD IS BLANK ALL PARAMETERS OF THE REMAINDER OF THE COMMAND SET ARE ENTERED BUT EXECUTION OF THE GRID INPUT AND CONTOURING IS BYPASSED.

LINE 2.

SECOND TITLE LINE TO BE PLOTTED IN THE MARGIN. THE FIRST TITLE LINE IS ALWAYS OBTAINED FROM THE STANDARD FORMAT FILE. DEFAULT:BLANKS.

LINE 3.

THIRD TITLE LINE. DEFAULT: BLANKS.

LINE 4.

FORMATS TO BE USED FOR CONTOUR AND NEATLINE ANNOTATION. NOTE THAT FORMAT DATA CONSISTS OF NOT ONLY THESE FIELDS BUT ALSO "NCHAR" AND "SIZE" TYPE VARIABLE IN THE &PARMS SECTION. IMPROPER MIXTURE CAN BREW STRANGE RESULTS.
CHARACTERS 1 THROUGH 20 CONTAIN FORMAT FOR CONTOUR LABEL. DEFAULT:BLANKS.
CHARACTERS 21 THROUGH 40 CONTAIN X-AXIS LABEL FORMAT. DEFAULT: BLANKS.
CHARACTERS 41 THROUGH 60 CONTAIN Y-AXIS LABEL FORMAT. DEFAULT: BLANKS.

LINE 5.

NAMELIST SECTION.

CHARACTERS 2 THROUGH 8 MUST CONTAIN:&PARMS . CHARACTERS 9 ON MAY CONTAIN NAMELIST ITEMS AS WELL A CONTINUATION LINES. SECTION MUST BE TERMINATED BY:&. REMEMBER: CHARACTER 1 ON EACH LINE IS IGNORED. EXCEPT FOR THE VARIABLES ACVAL, CMIN, CMAX AND IDASHS ALL VARIABLES ARE EXPECTED TO BE GREATER THAN OR EQUAL TO ZERO. IF THEY ARE LESS THAN ZERO THEY WILL BE ASSIGNED A ZERO VALUE.

CONTOUR LEVELS.

CONTOURS ARE DETERMINED BY INPUTTING EITHER AN ARRAY OF "SPECIFIED CONTOUR" LEVELS OR "INCREMENTAL CONTOUR" LEVELS. FOR THE SPECIFIED CONTOUR METHOD:

ACVAL=c(1),c(2),...,c(NCVAL) contour levels in monotonically ascending sequence.

NCVAL= number of elements in ACVAL list.

For the incremental contour level method:

NCVAL =0 (default value).

DCVAL = differential contour level. Must be greater than zero. Default = 0 thus creating an error condition if all contour specifications are ignored.

CMIN = lower limit of contour levels.

CMAX = upper limit of contour levels. If CMIN=CMAX=0. (default) then range of contouring determined by data.

NSEC = primary contour level interval. If less than or equal to one then all levels are considered primary. Default = 1.

GRADI = maximum gradient, in contours per inch, before secondary contours in grid cell are not plotted. Default = 30.

IDASHS = secondary-primary contour line dashing.
= 0 all contours plotted as solid lines.
= -1 secondary contours plotted as dashed lines.
= 1 primary contours plotted as dashed lines.

Contour labeling is performed for all specified levels and all primary incremental levels if NCHAR greater than zero. The following parameters affect the form of the contour labeling.

NCHAR = number of characters in format output field. Usually equal to the field width part of the format statement in field 1 of line 4. Default = 0 (no labeling).

SIZE = height, in inches, of labeling characters. Default = 0.06.

Plotter selection.

IPLOTR = plotter type code.
= 0 Gerber 622, default.
= 1 Tektronic 4010
= 2 Hewlett-Packard 7202A
= 3 Hewlett-packard 7203A
All other values will cause error conditions.

GRID SCALING.

XSCALE = X-AXIS DATA UNITS PER INCH. DEFAULT = 0.

YSCALE = Y-AXIS DATA UNITS PER INCH. DEFAULT = 0.
IF BOTH SCALING FACTORS ARE ZERO THEN BOTH AXIS SCALES WILL BE SET EQUAL TO THE SAME VALUE WHICH WILL ALLOW PLOTTING ON THE LESSER OF PLOT DEVICE BOARD SIZE OR 10 BY 8 INCHES. IF ONE SCALE FACTOR IS NOT ZERO AND THE OTHER IS ZERO THE ZERO FACTOR WILL BE ASSIGNED THE NON-ZERO VALUE.

GENERAL PLOT ANNOTATION DETAILS.

SIZEL = HEIGHT OF LABEL LINE CHARACTERS, IN INCHES. DEFAULT = 0.07. IF = ZERO THEN TITLE LINES NOT PLOTTED.

NOTE: THE FOLLOWING X OR Y ([X/Y]) SUFFIXES REFER TO THE RESPECTIVE X,Y AXIS LABELING AND POSITION PARAMETERS.

NCHAR[X/Y] =NUMBER OF CHARACTERS IN RESULTING AXIS LABELING FORMAT FIELD. NORMALLY EQUAL TO "W" FIELD OF RESPECTIVE LINE 4 FORMAT FIELDS. DEFAULT = 0 (IF NO LABELING).

SIZE[X/Y] = HEIGHT, IN INCHES, OF AXIS LABELING CHARACTERS. DEFAULT = 0.

ADEL[X/Y] = INTERVAL OF AXIS TICKS. DEFAULT = 0. IF EQUAL TO ZERO, THEN ONLY MINIMUM AND MAXIMUM VALUES POSTED.

LINT[X/Y] = PRIMARY INTERVAL OF POSTING. DEFAULT =1. IF LESS THAN 2 THEN EVERY INTERVAL LABELLED.

PLL[X/Y] = POSITION OF LOWER-LEFT CORNER OF GRID AREA RELATIVE TO PLOT EDGE, IN INCHES. IF EQUAL TO ZERO, THEN RESPECTIVE POSITION DETERMINED AUTOMATICALLY. PRINCIPLY AVAILABLE FOR SPECIAL PURPOSES.

EXAMPLE OF CONTUR COMMAND FILE.

THE FOLLOWING ASCII COMMAND FILE NAMED SAMPLE.CNT CREATED BY EITHER TECO OR $DECK PERFORMED THE PLOTS ON THE FOLLOWING PAGES. THE STANDARD TWO-DIMENSIONAL FILE LVGRAV WAS CREATED BY UTILTY FROM PUNCH CARDS OF HAND DIGITIZED DATA. THE RESULTS OF EACH COMMAND SET ARE SHOWN IN FIGS. 1, 2 AND 3.

```
LVGRAV                    (COMMAND SET 1)
PLOT SAMPLE 1


 &PARMS
  DCVAL=100.,IPLOTR=1,
 &END
LVGRAV                    (COMMAND SET 2)
PLOT SAMPLE 2
MORE COMPLEX CALL
(F6.0)
 &PARMS
  NSEC=5,NCHAR=5,SIZE=.07,
 &END
LVGRAV                    (COMMAND SET 3)
PLOT SAMPLE 3

(F6.0)                   (F8.0)                   (F8.0)
 &PARMS
  NCHARX=7,NCHARY=7,SIZEX=.08,SIZEY=.08,
  ADELX=1000.,ADELY=1000.,LINTX=10,LINTY=10,
 &END
```

CONTUR WAS EXECUTED AS:

```
.RUN CONTUR[333,724]

*SAMPLE.CNT

<<TK4010[X/Y:6.9/5.0-IN[CH I/O:3/3[LN I/O:3871/3846
<<  DEV:NULL[PLT000.DAT[BLKS:19

<<TK4010[X/Y:6.9/5.0-IN[CH I/O:12/12[LN I/O:2642/2617
<<  DEV:NULL[PLT000.DAT[BLKS:15

<<TK4010[X/Y:6.9/5.0-IN[CH I/O:192/192[LN I/O:2402/2402
<<  DEV:NULL[PLT000.DAT[BLKS:18
*EXIT
STOP

END OF EXECUTION
CPU TIME: 25.18 ELAPSED TIME: 1:17.12
EXIT

.
```

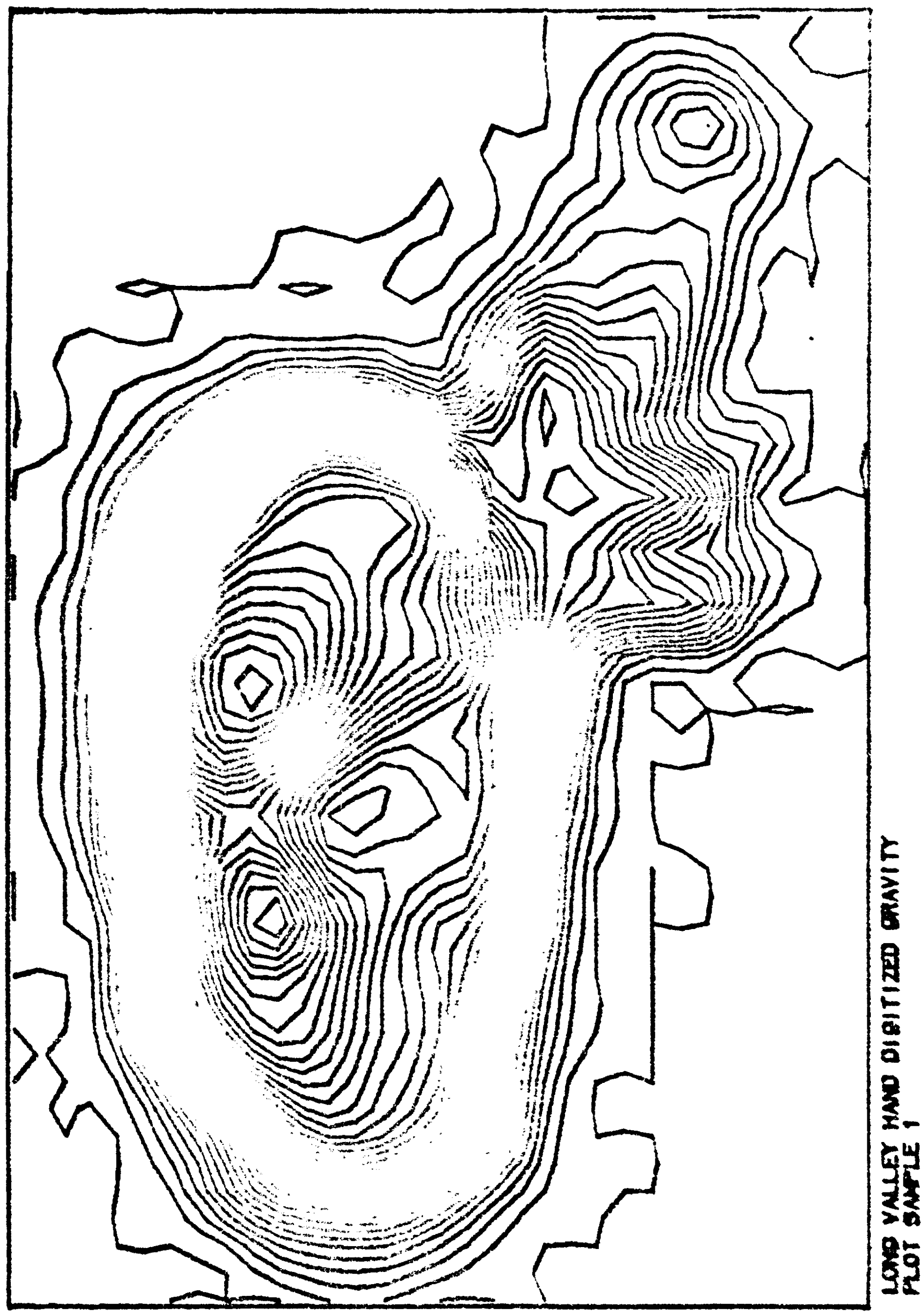


FIGURE 1. RESULTING CONTOUR MAP OF COMMAND SET 1.

Best copy available

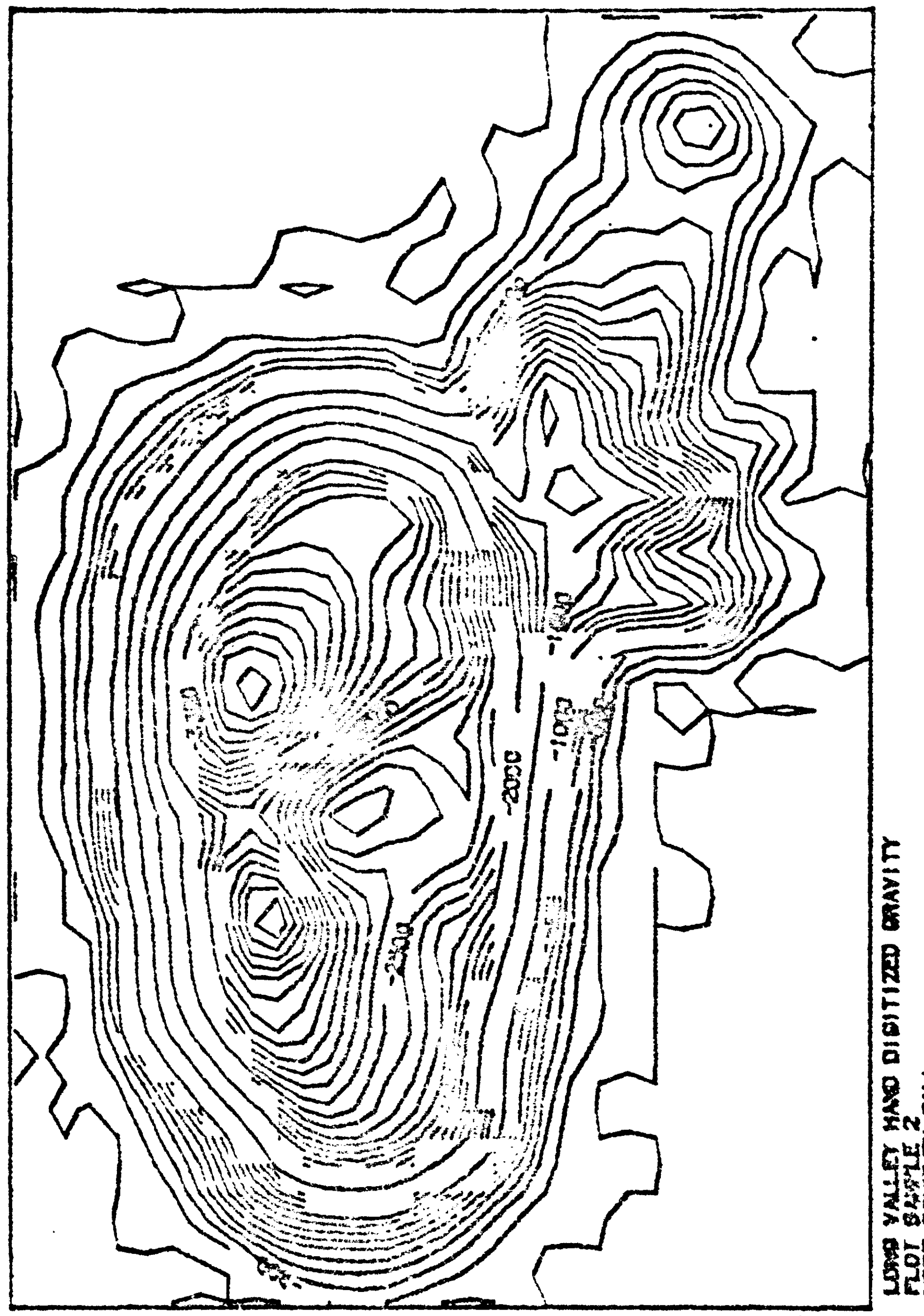


FIGURE 2. RESULTING CONTOUR MAP OF COMMAND SET 2.

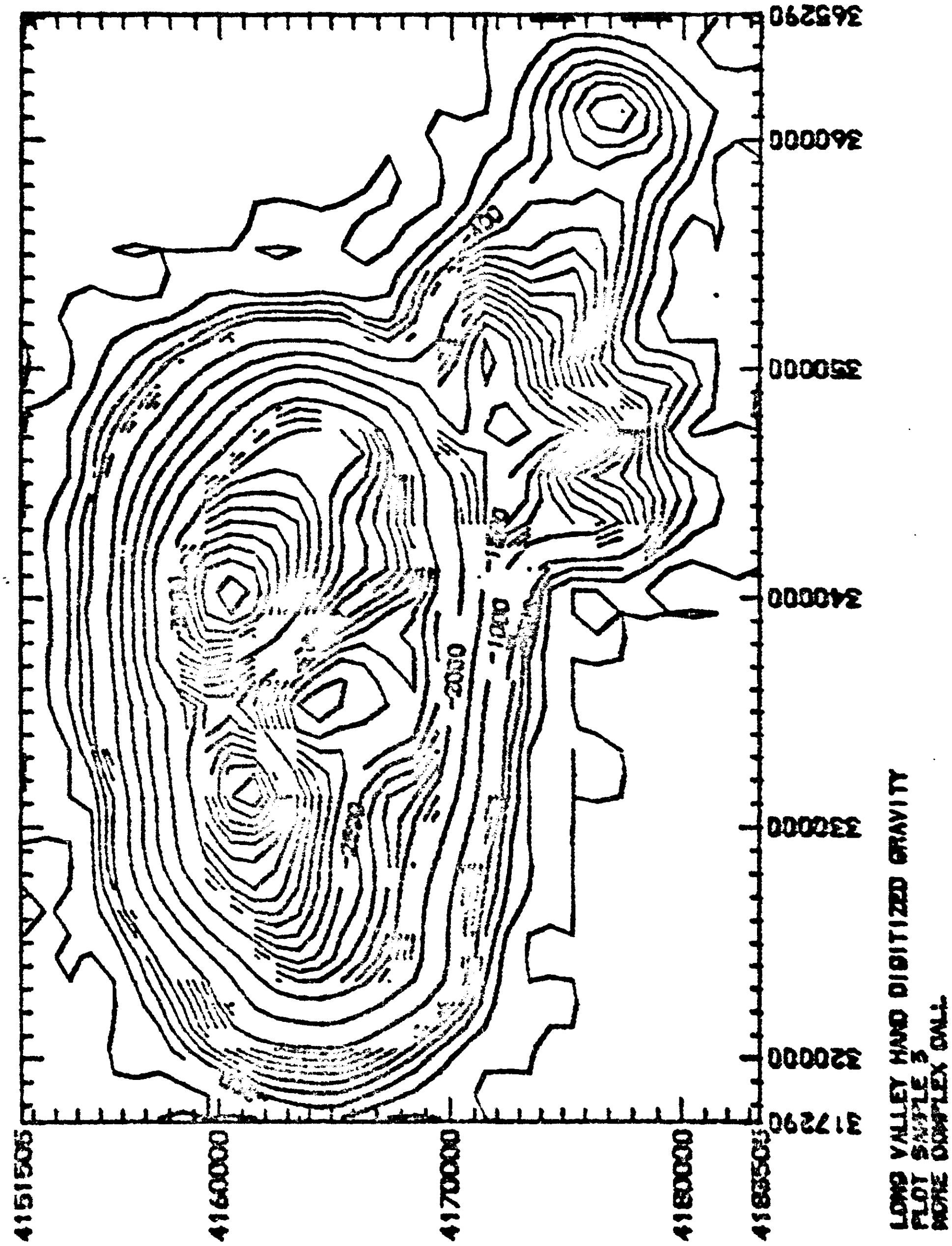


Figure 3. Resulting contour map of command set 3.

GENERAL.

SPLN2D PROVIDES TWO FUNCTIONS: 1) DETERMINE SPLINE COEFFICIENTS OF A TWO DIMENSIONAL GRID AND 2) TWO DIMENSIONAL INTERPOLATION. SPLINE POLYNOMIAL PROPERTIES AND ALGORITHMS FOR THEIR DETERMINATION ARE DESCRIBED BY ANDERSON (1971) AND HIS SUBROUTINES, WITH MINOR MODIFICATIONS, ARE EMPLOYED IN THIS PROGRAM. IN THE INTERPOLATION FUNCTION OF SPLN2D TWO BASIC MODES ARE PROVIDED: 1) SIMPLE INTEGRAL SUBDIVISION OF THE ORIGINAL GRID AND 2) DETERMINATION OF A NEW MESH AT EQUALLY SPACED COORDINATES.

THE SINGULAR DISADVANTAGE OF THE SPLINE METHOD IS MEMORY REQUIREMENTS. SPLN2D, AS CURRENTLY COMPILED, IS RESTRICTED TO DETERMINING SPLINE COEFFICIENTS OF GRIDS OF I COLUMNS AND J ROWS THAT MEET THE FOLLOWING RESTICTION:

4*I*J+17*I+J+6*MAXIMUM OF(I,J)+I2 .LE.20000

WHERE I2 IS THE NUMBER OF INTERPOLATED OUTPUT COLUMNS (I2=0 IF SPLINE POLYNOMIALS ARE OUTPUT). THIS REPRESENTS THE EQUIVALENT OF A 67 BY 67 GRID LIMIT WHEN I2=4*I. IF PRE-DETERMINED POLYNOMIAL COEFFICIENTS ARE INPUT, THE INTERPOLATION PHASE REQUIRES THAT:

33*I+2+I2.LE.20000.

INPUT, COEFFICIENT AND INTERPOLATED FILES ARE ALL STANDARD FORMAT FILES. THE INPUT FILE MUST HAVE NZ=1 (SEE APPENDIX A) AND, OBVIOUSLY, THE INTERPOLATED FILE WILL ALSO HAVE NZ=1 WITH THE PGM VARIABLE SET TO "SPLINE-INT". THE COEFFICIENT FILE WILL HAVE NZ=16 AND PGM="SPLNCOEFF**". ALSO, THE COEFFICIENT FILE WILL HAVE NCOL BY NROW DATA EVEN THOUGH NCOL-1 BY NROW-1 COEFFICIENT SETS ARE EMPLOYED (FIRST COEFFICIENT OF NCOL OR NROW COORDINATES ARE FLAGGED) AND THE Y VALUE OF EACH ROW RECORD CONTAINS THE ACTUAL VALUE REGARDLESS OF DY SETTING.

OPTIONALLY, THE DIRECTIONAL-DERIVATIVE AND SECOND VERTICAL DERIVATIVE CAN BE GENERATED AS THE INTERPOLATED OUTPUT. THESE ARE RESPECTIVELY DEFINED AS:

$$|\Delta u| = \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2\right)^{1/2}$$

AND

$$\frac{\partial^2 u}{\partial z^2} = -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

WHERE U IS THE BICUBIC SPLINE COEFFICIENTS. NOTE THAT FOR SECOND DERIVATIVE OUTPUT LINEAR SEGMENTS WILL RESULT. TO OBTAIN SMOOTH SECOND DERIVATIVE PLOTS RE-SPLINE AND INTERPOLATE WITH A SECOND PASS.

Execution of SPLN2D.

SPLN2D is executed by:

RUN SPLN2D[333,724]

The program responds by requesting:

ENTER INPUT FILE NAME.EXT:

If spline coefficients need to be determined the user enters the appropriate file name. If "EXIT" is entered the routine stops execution. Note that if a ".extension" is omitted, ".DAT" extension is assumed for all file name requests. If spline coefficients are already determined enter a blank line. The program then responds, in both cases, with

ENTER COEFFICIENT FILE NAME.EXT:

If the input file name is blank, then a name must be entered. If the input file name is not blank and it is desired to save the spline coefficients, then a name is entered. When coefficients are saved, the program starts over and asks for a new input file.

When coefficients are input or not saved the program proceeds to the interpolation phase and makes the following request for interpolated output file name:

ENTER INTERPOLATED FILE NAME.EXT:

If no file name is entered, the program starts over. After the name is entered a request for interpolation factors is asked for by:

ENTER PARAMETERS:

The user responds with:

&&PARMS selected namelist parameter values &

After the output file is generated the routine starts over.

Interpolation parameters.

For simple subdivision of grid cells:

IDELX=,IDELY= integral fractional parts that the respective X and Y axis of each grid cell are to be subdivided. One or both must be greater than zero. If one parameter is less than or equal to zero it is assigned the value of the other parameter.

This method is most frequently used for "smoothing" of grids for contouring.

FOR GENERATION OF AN EQUALLY SPACED GRID FROM SPLINE COEFFICIENTS:

XDEL=, YDEL= RESPECTIVE X AND Y INTERVAL OF GRID SPACING. IF ONE PARAMETER IS LESS THAN OR EQUAL TO ZERO IT IS ASSIGNED THE VALUE OF THE OTHER PARAMETER.

XOFF=,YOFF= RESPECTIVE X AND Y OFFSETS ALGEBRAICALLY ADDED TO THE "X0", "Y0" PARAMETERS OF THE INPUT DATA OR COEFFICIENT FILE (SEE APPENDIX A). DEFAULT: XOFF=YOFF=0.

NCOLO=,NROWO= NUMBER OF OUTPUT COLUMNS AND ROWS OF NEW, INTERPOLATED GRID.

IF ANY PORTION OF THE NEW GRID COORDINATES LAY OUTSIDE THE OLD GRID SYSTEM AUTOMATIC FLAGGING OF EMPTY CELLS OCCURS (I.E. NO EXTRAPOLATION). IT IS POSSIBLE TO GENERATE AN OUTPUT GRID COMPLETELY FLAGGED IF POOR VALUES ARE SELECTED.

FOR OBTAINING DERIVATIVE OUTPUT IN EITHER OF THE ABOVE MODES:

IDER.LT.0 FOR DIRECTIONAL·FIRST DERIVATIVE
IDER.GT.0 FOR SECOND VERTICAL DERIVATIVE
IDER.EQ.0 (DEFAULT) FOR NORMAL INTERPOLATION.

EXAMPLES.

FIGURE 4 SHOWS A TYPICAL CONTOUR MAP OF AN "UNSMOOTHED" GRID. TO OBTAIN A SMOOTHER PLOT (UNDERLINED ARE USER ENTRIES):

```
.RUN SPLN2D[333,724]

ENTER INPUT FILE NAME.EXT:LV
ENTER COEFFICIENT FILE NAME.EXT:
ENTER INTERPOLATED FILE NAME.EXT:A
ENTER PARAMETERS:
&&PARMS IDELX=4,&
ENTER INPUT FILE NAME.EXT:EXIT
STOP

END OF EXECUTION
CPU TIME: 3.40  ELAPSED TIME: 39.25
EXIT
```

THE RECONTOURED RESULT IS SHOWN IN FIGURE 5. TYPICALLY, SUBDIVIDING GRID BLOCKS BY 4 PRODUCES AN ADEQUATE CONTOUR MAP. OCCASIONALLY, A 6 TO 8 SUBDIVISION FACTOR IS NEEDED FOR EXTREMELY ROUGH DATA SETS.

FOR THE SECOND EXAMPLE THE SAME DATA SET IS INPUT, THE COEFFICIENTS SAVED AND AN EQUAL GRID SUB-AREA IS GENERATED (NOTE: ORIGINAL DATA HAS 25 COLUMNS BY 17 ROWS AT 2000 UNIT INTERVALS).

```
.RUN SPLN2D[333,724]

ENTER INPUT FILE NAME.EXT:LV
ENTER COEFFICIENT FILE NAME.EXT:LVCOEF
ENTER INPUT FILE NAME.EXT:
ENTER COEFFICIENT FILE NAME.EXT:LVCOEF
ENTER INTERPOLATED FILE NAME.EXT:B
ENTER PARAMETERS:
&&PARMS XDEL=500,XOFF=32710,NCOLO=20,
  YDEL=-500,NROWO=20,&
ENTER INPUT FILE NAME.EXT:EXIT
STOP

END OF EXECUTION
CPU TIME: 3.24  ELAPSED TIME: 1:18.12
EXIT
```

FIGURE 6 SHOWS A CONTOUR PLOT OF THE INTERPOLATED GRID.

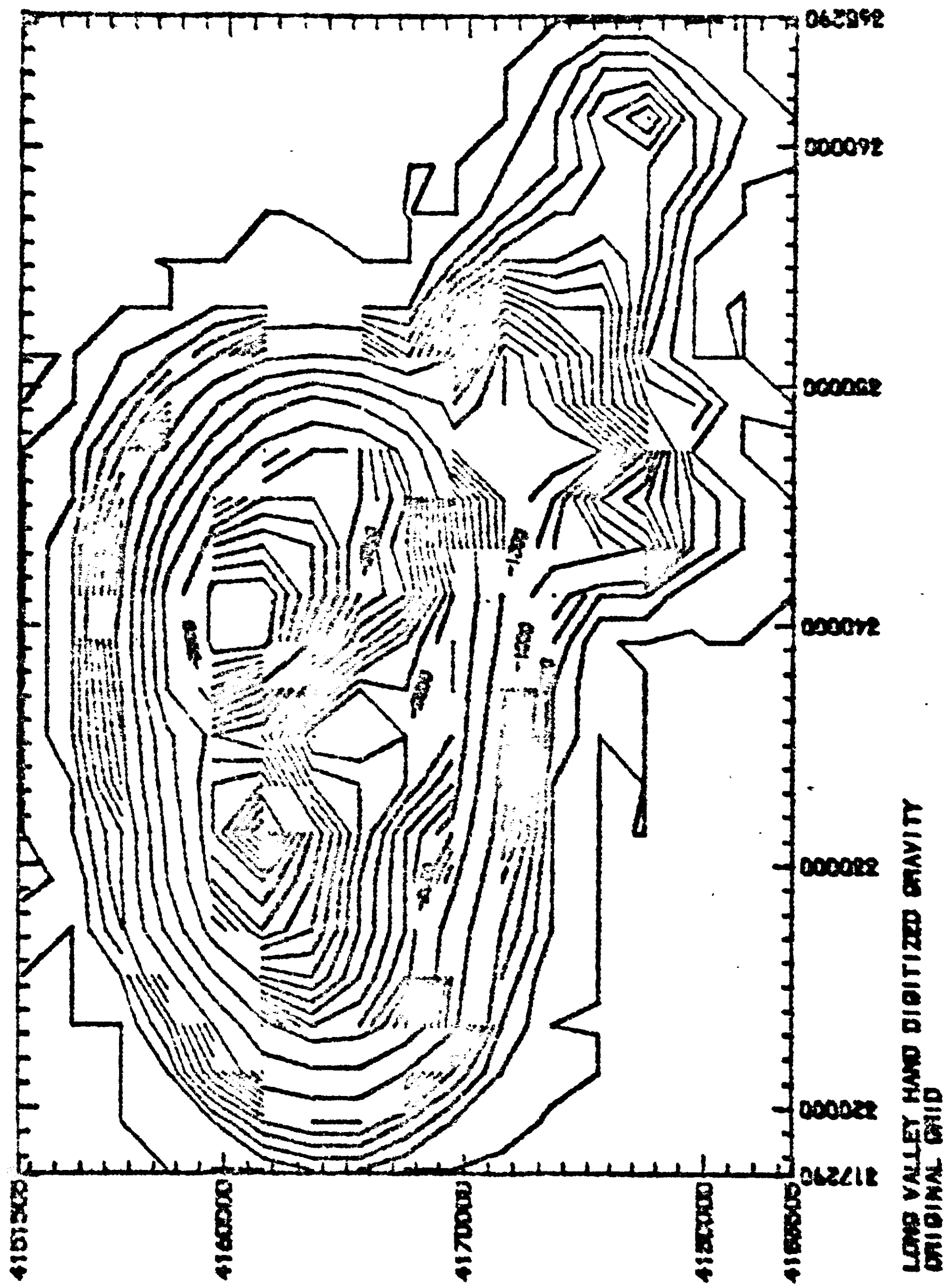


FIGURE 4. CONTOUR MAP OF ORIGINAL UNSMOOTHED GRID.

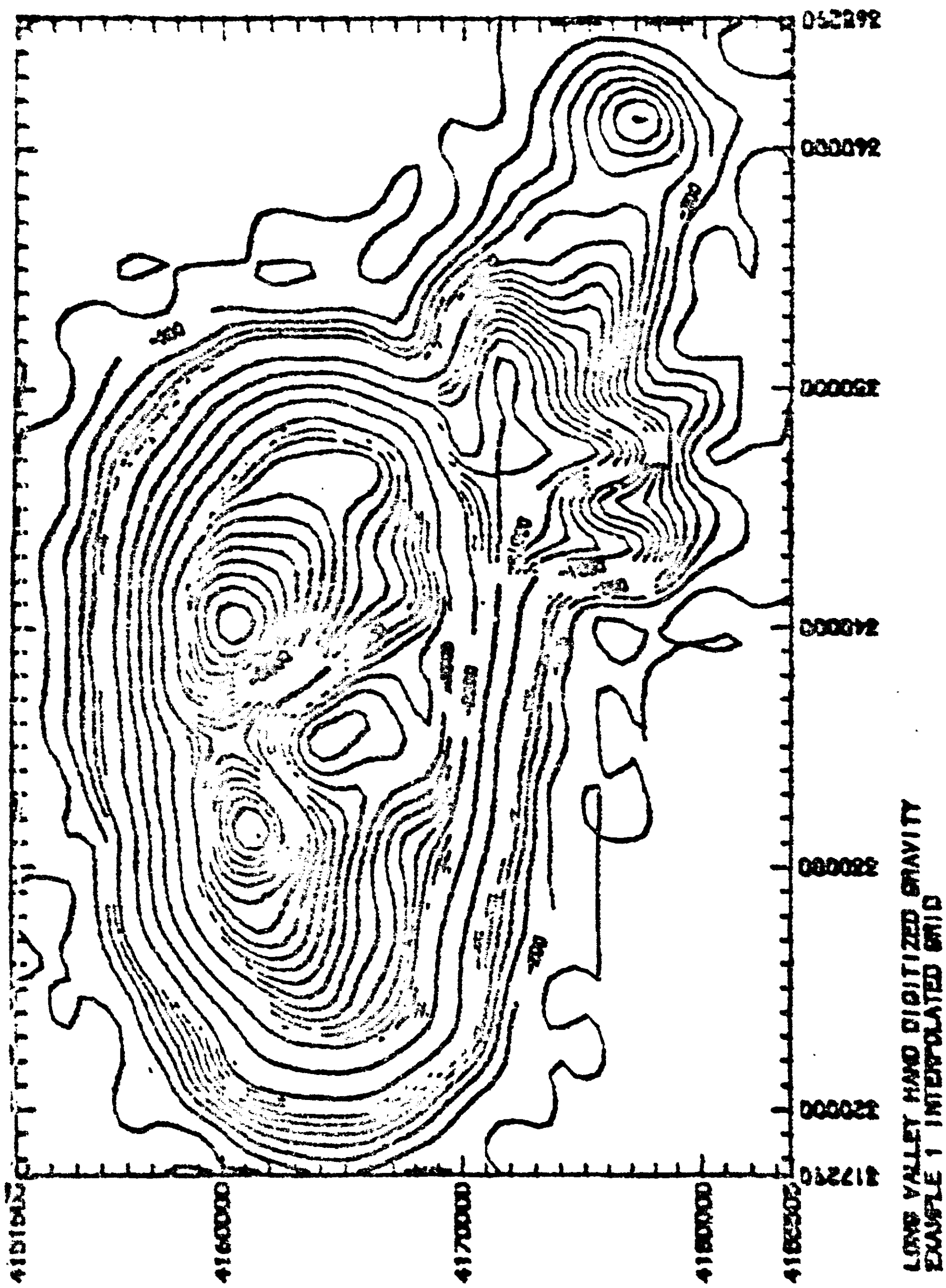


FIGURE 5. CONTOUR MAP OF SMOOTHED GRID.

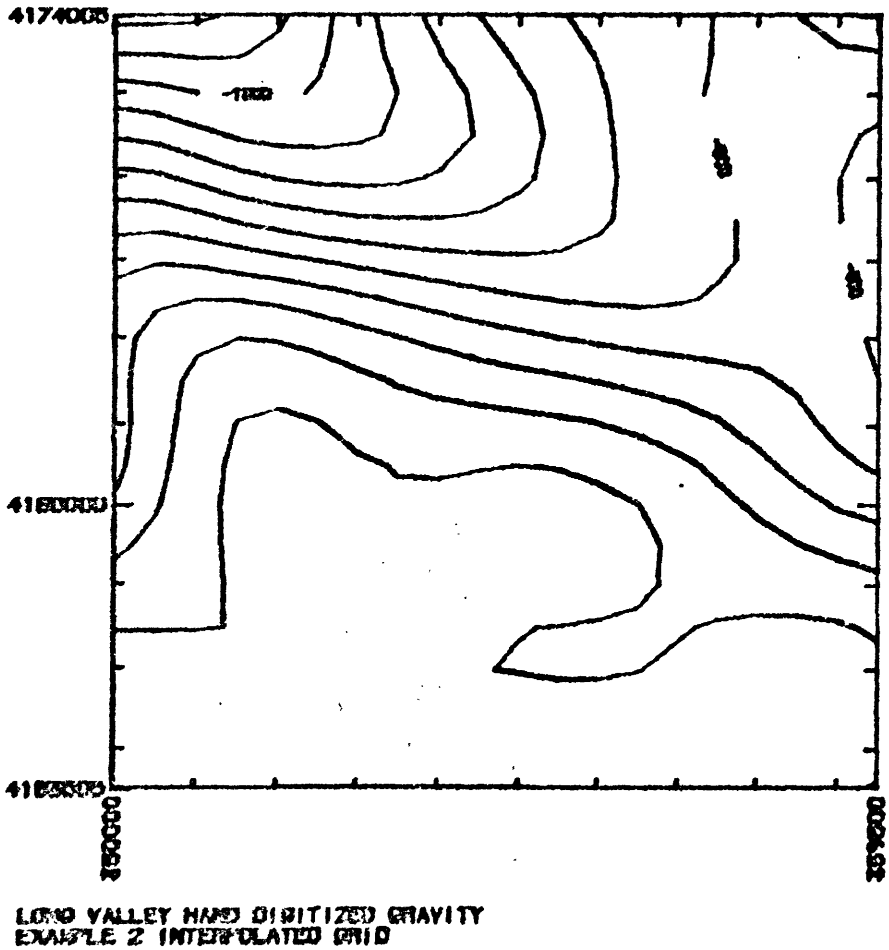


FIGURE 6. CONTOUR MAP OF EQUAL SPACED SUB-AREA.

GENERAL

SPLN2X IS A ROUTINE TO DETERMINE SPLINE COEFFICIENTS FOR LARGE TWO DIMENSIONAL RECTANGULAR GRIDS OF UP TO 500 ROWS BY 500 COLUMNS. THIS IS AN IMPROVED ALOGORITHM BASED UPON THE ORIGINAL PROGRAM BIGGRID (ANDERSON, 1971) WHICH OVERLAYS INTERMEDIATE RESULTS ON DISK. OPERATION OF THE PROGRAM IS NEARLY IDENTICAL TO SPLN2X EXCEPT INTERPOLATION IS NOT PROVIDED. IF INTERPOLATION IS DESIRED, SPLN2D CAN BE EXECUTED WITH SPLINE COEFFICIENTS GENERATED BY THIS PROGRAM AS INPUT.

GRID INPUT AND SPLINE COEFFICIENT OUTPUT ARE IN STANDARD TWO DIMENSIONAL FORM. THREE SEQUENTIAL WORK FILES ARE REQUIRED OF APPROXIMATELY THE SAME SIZE AS THE INPUT FILE. THE DISK NAMES OF THESE FILES ARE LPS991.TMP, LPS992.TMP AND LPS993.TMP. THEY ARE DELETED UPON NORMAL TERMINATION OF THE JOB. THE SIZE OF THE OUTPUT COEFFICIENT FILE WILL BE APPROXIMATELY 16 TIMES THE INPUT FILE.

EXECUTION OF SPLN2X

SPLN2X IS EXECUTED BY:

RUN SPLN2X[333,724]

THE PROGRAM RESPONDS BY REQUESTING:

ENTER INPUT FILE NAME.EXT:

A NON-BLANK NAME MUST BE ENTERED. IF THE FIELD IS BLANK A RE-REQUEST IS MADE. TO EXIT FROM SPLN2X ENTER "EXIT". IF THE INPUT FILE APPEARS TO BE ACCEPTABLE, A REQUEST FOR THE NAME OF THE SPLINE COEFFICIENT FILE IS MADE:

ENTER COEFFICIENT FILE NAME.EXT:

AFTER THE COEFFICIENT FILE HAS BEEN WRITTEN THE PROGRAM RESTARTS AND REQUESTS A NEW INPUT FILE.

BECAUSE OF THE SIMPLICITY OF EXECUTION AND SIMILARITIES TO SPLN2D, NO EXAMPLES ARE PRESENTED.

REFERENCES.

ANDERSON, W. L., 1971, APPLICATION OF BICUBIC SPLINE FUNCTIONS TO TWO-DIMENSIONAL GRIDDED DATA:NATIONAL TECHNICAL INFORMATION SERVICE REPORT PB-203579, 87P.

APPENDIX A.

STANDARD FILE.

THE ESTABLISHED ONE OR TWO DIMENSIONAL FILE FACILITATES DATA LINKAGE BETWEEN COMMONLY USED PROGRAMS. IT CONSISTS OF TWO BASIC PARTS: 1) A HEADER RECORD PROVIDING CONTROL PARAMETERS AND, OPTIONALLY, A FOLLOWING RECORD OF COLUMN COORDINATES AND 2) THE PROFILE OR GRID DATA. RECORDING MODE IS FORTRAN 'BINARY'.

A. HEADER RECORD (23 WORDS LONG)
   - ID: 70 ASCII CHARACTERS OF IDENTIFICATION (14 WORDS).
   - PGM: 10 ASCII CHARACTERS OF CREATION PROGRAM IDENTIFICATION (2 WORDS).
   - NCOL: NUMBER OF COLUMNS OF DATA (INTEGER, 1 WORD).
   - NROW: NUMBER OF ROWS OF DATA (INTEGER, 1 WORD). FOR ONE DIMENSIONAL DATA: SET TO 1.
   - NZ: NUMBER OF WORDS PER DATA ELEMENT (INTEGER, 1 WORD). FOR SINGLE PRECISION, SINGLE VALUED DATA USE 1, DOUBLE PRECISION OR COMPLEX USE 2, DOUBLE PRECISION COMPLEX USE 4, ... ETC.
   - X0: POSITION OF FIRST COLUMN OF DATA (REAL, 1 WORD).
   - DX: EQUAL SPACING INTERVAL OF COLUMNS (REAL, 1 WORD). IF EQUAL TO ZERO, THEN COORDINATE FOR EACH COLUMN IS IN THE DATA RECORD; OTHERWISE THE FOLLOWING RECORD CONSISTS OF DATA.
   - Y0: POSITION OF FIRST ROW (REAL, 1 WORD).
   - DY: EQUAL SPACING INTERVAL OF ROWS (REAL, 1 WORD). IF EQUAL TO ZERO, THE THE COORDINATE FOR EACH ROW IS THE FIRST WORD OF EACH DATA RECORD ROW.

B. COLUMN COORDINATE RECORD, PRESENT ONLY IF DX OF HEADER RECORD EQUAL TO ZERO. RECORD CONSISTS OF NCOL REAL WORDS SPECIFYING THE COORDINATES OF EACH DATA COLUMN. MOST PROGRAMS REQUIRE THE COORDINATES TO BE IN MONOTONIC ORDER.

C. DATA RECORD. EACH DATA RECORD CONTAINS ONE ROW OF REAL DATA ITEMS. THE TOTAL RECORD LENGTH IS NCOL TIMES NZ PLUS 1 WORDS. THE FIRST WORD CONTAINS THE ROW COORDINATE IF DY ZERO. WHEN DY NOT ZERO, THE FIRST WORD IS IGNORED. WHEN SPECIFIED, THE ROW COORDINATES SHOULD BE IN MONOTONIC SEQUENCE. MANY PROGRAMS ACCEPT INCOMPLETE DATA WHERE THE MISSING DATA IS FLAGGED BY THE MAXIMUM FLOATING POINT NUMBER (OCTAL 37777777777).

IN GENERAL, I/O FOR THIS STANDARD FILE CAN BE STATED IN FORTRAN AS:

```
        DIMENSION G(IZ,IX,IY),ID(14),PGM(2),X(IX),Y(IY)
         ..
         ..
        READ OR WRITE (..) ID,PGM,NCOL,NROW,NZ,XO,DX,YO,DY
        IF (DX.EQ.0.) READ OR WRITE (..) (X(I),I=1,NCOL)
        IF (DY.NE.0.) GO TO 15
        DO 10 J=1,NROW
 10     READ OR WRITE(..) Y(J),((G(K,I,J),K=1,NZ),I=1,NCOL)
        GO TO 20
 15     DO 19 J=1,NROW
 19     READ OR WRITE (..) DUM,((G(K,I,J),K=1,NZ),I=1,NCOL)
 20     CONTINUE
         ..
         ..
```

THE ABOVE IS A SIMPLISTIC EXAMPLE AND GENERAL USAGE IS NOT WARRANTED NOR SUGGESTED. IT IS ONLY PROVIDED AS A GUIDE.

APPENDIX B.

LISTING OF PROGRAM CONTUR

```
      PROGRAM CONTUR
C
C  GENERAL CONTOURING PROGRAM
C
C  DEVELOPED AND CODED BY..
C    GERALD IAN EVENDEN
C    U. S. GEOLOGICAL SURVEY
C    DENVER FEDERAL CENTER
C    DENVER, COLORADO  80225
C
      DIMENSION WORK(10000)
C
C  VERSION IV-B
C
C  NOTE.. THIS PROGRAM CONTAINS SEVERAL NON-ANSI FORTRAN
C    STATEMENTS.  ALTHOUGH SEVERAL OF THESE CONSTRUCTS
C    MAY BE AVAILABLE ON OTHER COMPUTER SYSTEMS CARE MUST
C    BE EXERCISED IN THE TRANSPORTATION OF THIS PROGRAM.
C
C   A WARRANTEE ON PROPER FUNCTIONING OF THIS
C   PROGRAM IS NEITHER IMPLIED NOR EXPRESSED.
C
      DOUBLE PRECISION TITLE1,TITLE2,TITLE3,FMTX,FMTY
      NAMELIST /PARMS/
     1   NCVAL,CMIN,CMAX,NSEC,DCVAL,XSCALE,YSCALE
     2 ,ACVAL,GRADI,IPLOTR,NCHAR,SIZE,IDASHS
     3 ,SIZEL,NCHARX,NCHARY,SIZEX,SIZEY,ADELX,ADELY,
     4 PLLX,PLLY,LINTX,LINTY
      COMMON /SEXY/
     1  TITLE1(7),TITLE2(8),TITLE3(8),FMTX(2),FMTY(2),
     2 XX(2),YY(2),XP(4),YP(4),IPLOTR,SIZEL,NCHARX,NCHARY,
     3 SIZEX,SIZEY,ADELX,ADELY,PLLX,PLLY,LINTX,LINTY,
     4 XSCALE,YSCALE,XXSCAL,YYSCAL
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
C
      COMMON/CONTRC/ CMIN,CMAX,DCVAL,NCVAL,NSEC,GRADI
      DOUBLE PRECISION CONAME,FNAME,FMTC(2),BLANK
      DIMENSION DUMDUM(6)
      EQUIVALENCE (DUMDUM(1),CONAME),(DUMDUM(4),FNAME)
      LOGICAL NOMORE
      DIMENSION ACVAL(1000)
      EQUIVALENCE (ACVAL(1),WORK(1))
      DATA NWORK/10000/
      DATA DUMDUM/6*0./
      DATA BLANK/'          '/
C
```

```
      .
      DATA ICMD,IFILE/5,10/
C
C  GET COMMAND FILE NAME.
C
   10 TYPE 20
   20 FORMAT(' *',$)
      ACCEPT 30,CONAME
   30 FORMAT(A10)
      IF (CONAME.EQ.'EXIT     ') STOP
      OPEN (UNIT=ICMD,MODE='ASCII',ACCESS='SEQIN',
     1 FILE=CONAME,DEVICE='DSK')
      NWKRES=NWORK
C
C  GET CONTROL
C  INPUT DATA FILE NAME AND FORMATS.
   40 CALL HEADS(FNAME,TITLE2,TITLE3,FMTC,FMTX,FMTY,ICMD,I)
      IF (I.NE.0) GO TO 240
C
C  GET NUMERIC CONTROL VALUES.
      READ(ICMD,PARMS,END=220)
      IF (NSEC.LT.0) NSEC=0
      IF (GRADI.LT.0.) GRADI=0.
      IF (NCHAR.LT.0) NCHAR=0
      IF (SIZE.LT.0.) SIZE=0.
C
C  START CHECK-OUT.
C  CONTOUR VALUES...
      IF (NCVAL.GT.0.) GO TO 70
C
C  INCRIMENTAL CONTOURS MODE.
      NCVAL=0
      IF (DCVAL.GT.0.) GO TO 60
      IF (FNAME.EQ.BLANK) GO TO 40
      TYPE 50
   50 FORMAT(' %DCVAL LESS THAN OR EQUAL TO ZERO')
      GO TO 40
   60 IXAD=1
      GO TO 110
C
C  SPECIFIED CONTOURS MODE.
   70 IF (NCVAL.EQ.1) GO TO 100
C
C  CHECK ASCENDENCY.
      DO 90 I=2,NCVAL
      IF (WORK(I).GT.WORK(I-1)) GO TO 90
      TYPE 80
   80 FORMAT(' %NON-ASCENDING SPECIFIED CONTOURS')
      GO TO 40
   90 CONTINUE
  100 IXAD=NCVAL+1
C
C  ALL THAT CAN BE DONE WITHOUT LOOKING AT DATA FILE.
C
C  OPEN FILE AND PRELIM CHECK.
```

```
  110 NWKRES=NWKRES-NCVAL
      IF (FNAME.EQ.BLANK) GO TO 40
      CALL OPENCK(IFILE,FNAME,X0,Y0,DELX,DELY,
     1 WORK(IXAD),NWKRES,IQUAD)
      IF (NCOL.EQ.0) GO TO 40
C
C  SLICE UP CORE
      IF (IQUAD.EQ.3) GO TO 150
      NROWT=4*(NWKRES-NCOL)/(5*NCOL+4)
      IF (NROWT.GE.2) GO TO 140
C
  120 TYPE 130
  130 FORMAT(' INSUFFICIENT MEMORY')
      GO TO 210
C
  140 IDAD=IXAD+NCOL
      IYAD=NWORK-NROWT+1
      IFAD=IDAD+NROWT*NCOL
      GO TO 160
C
  150 NROWT=4*NWKRES/(13*NCOL)
      IF (NROWT.LT.2) GO TO 120
      K=NROWT*NCOL
      IDAD=IXAD+K
      IYAD=NWORK-K+1
      IFAD=IDAD+K
C
C  OK SO FAR, SCALE AND ANNOTATE
  160 CALL SEXUAL(I)
      IF (I.NE.0) GO TO 210
      IF (IQUAD.EQ.3) GO TO 200
      K=IXAD+NCOL-1
      IF (DELX.NE.0.) GO TO 180
      DO 170 I=IXAD,K
  170 WORK(I)=WORK(I)*XXSCAL
      GO TO 200
  180 X0=X0*XXSCAL
      DELX=DELX*XXSCAL
      DO 190 I=IXAD,K
      WORK(I)=X0
  190 X0=X0+DELX
C
  200 NPASS=(NROW+NROWT-3)/(NROWT-1)
      JRES=1+NROW-NROWT-(NPASS-2)*(NROWT-1)
C
      CALL SETCON(FMTC,SIZE,NCHAR)
      CALL CONTTR(WORK(IXAD),WORK(IYAD),WORK(IDAD),
     1 WORK(IFAD),WORK,NROWT,NCOL,XXSCAL,YYSCAL,GFLG,
     2 Y0*YYSCAL,DELY*YYSCAL,IQUAD,NPASS,JRES,IFILE)
C
C  CLOSE PLOT
      CALL ENDPLT(0)
C
C  CLOSE DATA FILE
```

```
      .
  210 CLOSE(UNIT=IFILE)
      GO TO 40
C
C  COMMAND FILE END
  220 TYPE 230
  230 FORMAT(' %ODD EOF ON COMMAND FILE')
  240 CLOSE(UNIT=ICMD)
      GO TO 10
      END
```

```
      SUBROUTINE HEADS(FNAME,T1,T2,FMTA,FMTB,FMTC,
     1  IFILE,IER)
C
C  SET FIRST 4 LINES OF CONTROL DATA.
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION T1(8),T2(8),FMTA(2),FMTB(2),FMTC(2),STRL(8)
C
      READ(IFILE,10,END=20) FNAME
   10 FORMAT(8A10)
      READ(IFILE,10,END=20) STRL
      CALL UPDTE(T1,STRL,8)
      READ(IFILE,10,END=20) STRL
      CALL UPDTE(T2,STRL,8)
      READ(IFILE,10,END=20) STRL
      CALL UPDTE(FMTA,STRL,2)
      CALL UPDTE(FMTB,STRL(3),2)
      CALL UPDTE(FMTC,STRL(5),2)
      IER=0
      GO TO 30
   20 IER=1
   30 RETURN
      END
```

```
      SUBROUTINE UPDTE(A,B,N)
C
C  UPDATE A IF B NON-BLANK
C
      DOUBLE PRECISION A(1),B(1),BLANK
      DATA BLANK/'        '/
      DO 10 I=1,N
      IF (B(I).NE.BLANK) GO TO 20
   10 CONTINUE
      GO TO 40
   20 DO 30 I=1,N
   30 A(I)=B(I)
   40 RETURN
      END
```

```
      SUBROUTINE OPENCK(IFILE,FNAME,X0,Y0,DELX,DELY,
     1 XDATA,NWORK,IQUAD)
C
C   INITIALIZE GRID INPUT FILE.
C
      DOUBLE PRECISION FNAME,DUMMY
      DIMENSION XDATA(1)
      DOUBLE PRECISION TITLE1,TITLE2,TITLE3,FMTX,FMTY
      COMMON /SEXY/
     1  TITLE1(7),TITLE2(8),TITLE3(8),FMTX(2),FMTY(2),
     2 XX(2),YY(2),XP(4),YP(4),IPLOTR,SIZEL,NCHARX,NCHARY,
     3 SIZEX,SIZEY,ADELX,ADELY,PLLX,PLLY,LINTX,LINTY,
     4 XSCALE,YSCALE,XXSCAL,YYSCAL
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      NCOL=0
      IPASS=0
   10 OPEN(UNIT=IFILE,DEVICE='DSK',MODE='BINARY',
     1 ACCESS='SEQIN',FILE=FNAME)
      IF (IPASS.EQ.0) GO TO 20
      READ(IFILE)
      IF (DELX.EQ.0.) READ(IFILE)
      RETURN
C
   20 IPASS=1
      READ(IFILE,END=90) TITLE1,DUMMY,NCOL,NROW,
     1 IQUAD,X0,DELX,Y0,DELY
      IF (NCOL.LT.3.OR.NROW.LT.3) GO TO 110
      IF (IQUAD.EQ.3.AND.DELX.EQ.0..AND.DELY.EQ.0.)
     1 GO TO 80
      IF (IQUAD.NE.1) GO TO 130
      IF (DELX.NE.0.) GO TO 30
      IF (NCOL.GT.NWORK) GO TO 150
      READ(IFILE,END=90) (XDATA(I),I=1,NCOL)
      XX(1)=XDATA(1)
      XX(2)=XDATA(NCOL)
      GO TO 40
   30 XX(1)=X0
      XX(2)=X0+(NCOL-1)*DELX
   40 IF (DELY.NE.0.) GO TO 50
      CALL YREAD(IFILE,YY,NROW)
      IF (NROW.LT.3) GO TO 110
      GO TO 60
   50 YY(1)=Y0
      YY(2)=Y0+(NROW-1)*DELY
   60 IF (DELX.NE.0..AND.DELY.NE.0.) RETURN
   70 CLOSE(UNIT=IFILE)
      GO TO 10
   80 I=NCOL*3
      IF (I.GT.NWORK) GO TO 150
      READ (IFILE)
      CALL XYREAD(IFILE,XX,YY,XDATA,I)
```

```
      .
      IF (NROW.LT.3) GO TO 110
      GO TO70
   90 TYPE 100
  100 FORMAT(' %END OF FILE WHILE PROCESSING HEADER')
      GO TO 170
  110 TYPE 120
  120 FORMAT(' %NO. ROWS OR COLUMNS LESS THAN 3')
      GO TO 170
  130 TYPE 140
  140 FORMAT(' %NO. Z ARGUMENTS GREATER THAN 1')
      GO TO 170
  150 TYPE 160
  160 FORMAT(' %CORE EXCEEDED FOR X OR ROW VALUES')
C
  170 CLOSE(UNIT=IFILE)
      NCOL=0
      RETURN
      END
```

```
      SUBROUTINE XYREAD(IFILE,XR,YR,WORK,NCOL3)
C
C   SCANS QUADRILATERAL GRID FOR X-Y RANGE
C
      DIMENSION XR(2),YR(2),WORK(NCOL3)
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
C   INITIALIZE
      NROW=0
      READ(IFILE,END=70) DUMMY,WORK
      XMIN=FLTMAX
      YMIN=FLTMAX
      XMAX=-FLTMAX
      YMAX=XMAX
C
C   SCAN LOOP
   10 NROW=NROW+1
      IF (WORK(I+2).EQ.GFLG) GO TO 50
      DO 50 I=1,NCOL3,3
      IF (WORK(I).LE.XMAX) GO TO 20
      XMAX=WORK(I)
      GO TO 30
   20 IF (WORK(I).GE.XMIN) GO TO 30
      XMIN=WORK(I)
   30 IF (WORK(I+1).GE.YMIN) GO TO 40
      YMIN=WORK(I+1)
      GO TO 50
   40 IF (WORK(I+1).LE.YMAX) GO TO 50
      YMAX=WORK(I+1)
   50 CONTINUE
C
      READ(IFILE,END=60) DUMMY,WORK
      GO TO 10
C
   60 XR(1)=XMIN
      XR(2)=XMAX
      YR(1)=YMIN
      YR(2)=YMAX
C
   70 RETURN
      END
```

```
      SUBROUTINE YREAD(IFILE,YR,NROW)
C
C   SCANS SPECIFIED ROW GRID
      DIMENSION YR(2)
C
      NROW=0
      READ(IFILE,END=20) YR(1)
C
   10 NROW=NROW+1
      READ(IFILE,END=20) A
      YR(2)=A
      GO TO 10
   20 RETURN
      END
```

```
      $UBROUTINE COORDA(X,Y)
      DIMENSION X(0/0),Y(0/0)
C
C  ESTABLISH LOCAL X AND Y ADDRESSES
      RETURN
      ENTRY COORD(I,J,XV,YV)
      XV=X(I)
      YV=Y(J)
      RETURN
      END
```

```
      SUBROUTINE COORQA(X,Y,NCOL)
C
C  QUADRILATERAL COORDINATES
      DIMENSION X(0:0),Y(0:0)
      RETURN
C
      ENTRY COORDQ(I,J,XV,YV)
      K=J*NCOL + I
      XV=X(K)
      YV=Y(K)
      RETURN
      END
```

```
      SUBROUTINE SEXUAL(IER)
C
C  GENERAL SETUP OF CONTOUR NICETIES.
C
      DOUBLE PRECISION TITLE1,TITLE2,TITLE3,FMTX,FMTY
      COMMON /SEXY/
     1  TITLE1(7),TITLE2(8),TITLE3(8),FMTX(2),FMTY(2),
     2 XX(2),YY(2),XP(4),YP(4),IPLOTR,SIZEL,NCHARX,NCHARY,
     3 SIZEX,SIZEY,ADELX,ADELY,PLLX,PLLY,LINTX,LINTY,
     4 XSCALE,YSCALE,XXSCAL,YYSCAL
C
      IF (SIZEX.LT.0.) SIZEX=0.
      IF (SIZEY.LT.0.) SIZEY=0.
      IF (NCHARX.LT.0) NCHARX=0
      IF (NCHARY.LT.0) NCHARY=0
      IF (ADELX.LT.0.) ADELX=0.
      IF (ADELY.LT.0.) ADELY=0.
      IF (LINTX.LT.0) LINTX=0
      IF (LINTY.LT.0) LINTY=0
      IF (SIZEL.LT.0.) SIZEL=0.
      SIZELB=1.5*SIZEL
C
C  MARGIN REQUIREMENT SETUP.
      IF (SIZEX.GT.0..AND.NCHARX.GT.0) GO TO 10
      NCHARX=0
      SIZEX=0.
   10 IF (SIZEY.GT.0..AND.NCHARY.GT.0) GO TO 20
      NCHARY=0
      SIZEY=0.
   20 IF (PLLX.LE.0.) GO TO 30
      XP(3)=PLLX
      GO TO 40
   30 XP(3)=(NCHARY+0.6)*SIZEY
   40 IF (PLLY.LE.0.) GO TO 50
      YP(3)=PLLY
      GO TO 60
   50 YP(3)=(NCHARX+0.6)*SIZEX+4.0*SIZELB
C
   60 DX=XX(2)-XX(1)
      DY=YY(2)-YY(1)
      CALL PLTSET(IPLOTR,XP(4),YP(4),0)
C
C  CHECK SCALING
      IF (XSCALE.LE.0.) GO TO 70
      IF (YSCALE.LE.0.) YSCALE=XSCALE
      GO TO 80
   70 IF (YSCALE.LE.0.) GO TO 90
      XSCALE=YSCALE
   80 XXSCAL=SIGN(1./XSCALE,DX)
      YYSCAL=SIGN(1./YSCALE,DY)
C
C  FIXED SCALING.
      XP(1)=ABS(DX*XXSCAL)
      YP(1)=ABS(DY*YYSCAL)
```

```
      XP(4)=XP(1)+XP(3)+.501*SIZEX
      YP(4)=YP(1)+YP(3)+.501*SIZEY
      GO TO 120
C
C  RELATIVE SCALING, EG. XXSCAL=YYSCAL=0.
   90 XP(4)=AMIN1(XP(4),10.)
      YP(4)=AMIN1(YP(4),8.)
      XP(1)=XP(4)-XP(3)-.501*SIZEX
      YP(1)=YP(4)-YP(3)-.501*SIZEY
      IF (XP(1).GT.0..AND.YP(1).GT.0.) GO TO 110
      TYPE 100
  100 FORMAT(' %MARGIN REQUIRES ALL PLOT AREA')
      IER=2
      GO TO 140
  110 XXSCAL=XP(1)/DX
      YYSCAL=YP(1)/DY
      XXSCAL=SIGN(AMIN1(ABS(XXSCAL),ABS(YYSCAL)),XXSCAL)
      YYSCAL=SIGN(XXSCAL,YYSCAL)
      XP(1)=ABS(DX*XXSCAL)
      YP(1)=ABS(DY*YYSCAL)
C
C  INITIAL LABELING SCALE CALL
  120 CALL SCALE(XX,YY,XP,YP,4,IER)
      IF (IER.NE.0) GO TO 140
C
C  PLOT LABELS.
      IF (SIZEL.LE.0.) GO TO 130
      CALL CHAR(SIZELB,SIZELB,TITLE3,80,3,SIZEL,0.,0.,0.)
      YT=SIZELB+SIZELB
      CALL CHAR(SIZELB,YT,TITLE2,80,3,SIZEL,0.,0.,0.)
      YT=YT+SIZELB
      CALL CHAR(SIZELB,YT,TITLE1,70,3,SIZEL,0.,0.,0.)
C
C  PLOT AXIS.
  130 CALL XAXIS(XX,YY,XP,ADELX,LINTX,SIZEX,FMTX,NCHARX)
      CALL YAXIS(YY,XX,YP,ADELY,LINTY,SIZEY,FMTY,NCHARY)
      CALL NEATLN
C
C  RESCALE PLOTTER FOR GRID.
      XX(1)=XX(1)*XXSCAL
      XX(2)=XX(2)*XXSCAL
      YY(1)=YY(1)*YYSCAL
      YY(2)=YY(2)*YYSCAL
      CALL SCALE(XX,YY,XP,YP,3,IER)
  140 RETURN
      END
```

```
      SUBROUTINE CONTTR(X,Y,Z,F,C,NROW,NCOL,XXSCAL,YYSCAL,
     1 GFLG,YO,DELY,IQUAD,NPASS,JRES,IFILE)
C
C  BASIC INPUT AND TIER CONTROL
C
      DIMENSION X(1),Y(1),Z(1),F(1),C(1)
C
      EXTERNAL COORD,COORDQ
C
C  SETUP MODE CONTROL
      IF (IQUAD.EQ.3) GO TO 10
      CALL COORDA(X,Y)
      ITYPE=-1
      IF (DELY.EQ.0) ITYPE=0
      GO TO 20
   10 CALL COORQA(X,Y,NCOL)
      ITYPE=1
C
C  REMAINING INITIALIZATION
   20 KLROW=(NROW-1)*NCOL
      LAST=0
      JE=NROW
C
C  TIER LOOP
      DO 190 IPASS=1,NPASS
      IF (IPASS.EQ.NPASS) JE=JRES
      IF (IPASS.EQ.1) GO TO 70
C
C  MOVE DOWN LAST ROW
      JS=2
      JJ=NCOL
      DO 30 I=1,NCOL
   30 Z(I)=Z(KLROW+I)
      IF (ITYPE) 40,40,50
   40 Y(1)=Y(NROW)
      GO TO 80
   50 DO 60 I=1,NCOL
      X(I)=X(KLROW+I)
   60 Y(I)=Y(KLROW+I)
      GO TO 80
C
C  FIRST ROW
   70 JS=1
      JJ=0
C
C  GET DATA
   80 DO 140 J=JS,JE
      IF (ITYPE) 90,100,110
   90 READ(IFILE,END=130) DUM,(Z(JJ+I),I=1,NCOL)
      Y(J)=YO
      YO=YO+DELY
      GO TO 140
  100 READ(IFILE,END=130) Y(J),(Z(JJ+I),I=1,NCOL)
      Y(J)=Y(J)*YYSCAL
```

```
      GO TO 140
  110 READ(IFILE,END=130) DUM,(X(JJ+I),Y(JJ+I),Z(JJ+I),
     1 I=1,NCOL)
      DO 120 I=1,NCOL
      KK=JJ+I
      IF (Z(KK).EQ.GFLG) GO TO 120
      X(KK)=X(KK)*XXSCAL
      Y(KK)=Y(KK)*YYSCAL
  120 CONTINUE
      GO TO 140
  130 JE=J-1
      IF (JE.LT.2) RETURN
      LAST=1
      GO TO 150
  140 JJ=JJ+NCOL
C
C  CALL CONTUR
  150 IF (ITYPE) 160,160,170
  160 CALL CONTR(Z,F,C,COORD,JE)
      GO TO 180
  170 CALL CONTR(Z,F,C,COORDQ,JE)
  180 IF (LAST.NE.0) RETURN
  190 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONTR(GRID,FLAGS,ACVAL,COORD,NROWF)
C
C  BASIC CONTOURING CONTROL SUBROUTINE.
C
      DIMENSION GRID(1),FLAGS(1),ACVAL(1),COORD(1)
      COMMON /CONTRC/ CMIN,CMAX,DCVAL,NCVAL,NSEC,GRADI
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
C
      LOGICAL PRIME
C
      NROW=NROWF
C
C  SET GRADIENT.
      IF (NSEC.LE.0.OR.NCVAL.GT.0) GO TO 10
      GRAD=(GRADI*DCVAL)**2
C
C  SET FLAGS, ETC..
   10 CALL SETUP(GRID,FLAGS,COORD)
      IF (BMIN.GE.BMAX) GO TO 150
      IF (NCVAL.EQ.0) GO TO 50
C
C  FIND LOWER LIMIT.
      DO 20 I=1,NCVAL
      IF (ACVAL(I).LE.BMIN) GO TO 20
      II=I
      GO TO 30
   20 CONTINUE
C
C  CONTOURING LOOP
   30 PRIME=.TRUE.
      DO 40 I=II,NCVAL
      CONT=ACVAL(I)
      IF (CONT.GE.BMAX) GO TO 150
      CALL SETLAB
   40 CALL SCAN(GRID,FLAGS,COORD)
      GO TO 150
C
C  EXECUTION FOR DELTA CONTOUR LEVELS.
   50 IF (CMIN.EQ.0..AND.CMAX.EQ.0.) GO TO 60
      ICONT=AMAX1(BMIN,CMIN)/DCVAL
      AMX=AMIN1(BMAX,CMAX)
      GO TO 70
   60 ICONT=BMIN/DCVAL
      AMX=BMAX
C
C  CONTOURING LOOP.
   70 CONT=ICONT*DCVAL
      IF (CONT.GE.AMX) GO TO 150
      PRIME=NSEC.GT.0.AND.MOD(ICONT,NSEC).EQ.0
      IF (PRIME) GO TO 100
      IF (IDASHS) 80,140,90
```

```
   80 LINET=1
      GO TO 140
   90 LINET=0
      GO TO 140
  100 IF (IDASHS) 110,130,120
  110 LINET=0
      GO TO 130
  120 LINET=1
  130 CALL SETLAB
  140 CALL SCAN(GRID,FLAGS,COORD)
      ICONT=ICONT+1
      GO TO 70
C
C
C  DONE WITH BLOCK
  150 RETURN
      END
```

```
      SUBROUTINE SETUP(GRID,FLAGS,COORD)
C
C  INITIALIZE FLAG ARRAY.
C
      DIMENSION GRID(0/0),FLAGS(0/0)
      INTEGER FLAGS
C
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
C
C INITIALIZATION OF ALL PARAMETERS (ARITHMETIC AND LOGICAL)
C  UNIQUE TO ROW BLOCK BEING CONTOURED
C
C
C SYSTEM OF FLAGS
C     BIT          DEFINITION (FOR BIT ON)
C     0         EDGE-SIDE 4 POSSIBLE CUT
C     1         EDGE-SIDE 3 POSSIBLE CUT
C     2         EDGE-SIDE 2 POSSIBLE CUT
C     3         EDGE-SIDE 1 POSSIBLE CUT
C     4         EDGE 1 CUT NOT MADE AND NOT CHECKED
C     5         INTERIOR SCAN (EDGE 1)
C     6         GRADIENT (DROP SECONDARY CONTOURS)
C     7         BLOCK GOOD (CONTOURABLE)
C
      INTEGER FG
      LOGICAL CCC,COLLRC,ROWLG,BIT,KK,COLLG
C
C START OF OPERATIONS
C
      NROW1=NROW-1
      NCOL1=NCOL-1
      IJ=0
      I1J=1
      IJ1=NCOL
      I1J1=IJ1+1
      IJN1=-NCOL
      IJS=(NCOL*NROW+3)/4 -1
      DO 10 I=0,IJS
   10 FLAGS(I)=0
      IJS="377777777777
      IJE=0
C
C PRESET BMIN,BMAX
C
      BMIN=FLTMAX
      BMAX=-BMIN
C
C START OF SETUP SCANNING
C
      DO 130 J=0,NROW1
```

```
      ROWLG=.FALSE.
      COLLRC=.FALSE.
      IF (GRAD.EQ.0.OR.J.EQ.NROW1) GO TO 20
      CALL COORD(0,J,XLL,YLL)
      CALL COORD(0,J+1,XUL,YUL)
   20 CONTINUE
      DO 120 I=0,NCOL1
      IF (J.EQ.0) GO TO 30
      COLLG=BIT(FLAGS,IJN1,1)
      GO TO 40
   30 COLLG=.FALSE.
   40 IF (I.EQ.NCOL1) GO TO 70
      CCC=COLLG
      IF (J.EQ.NROW1) GO TO 70
C
C DETERMINE IF MESH BLOCK FLAGGED
C
      IF (GFLG.NE.0.AND.
     1 (GRID(IJ).EQ.GFLG.OR.GRID(I1J).EQ.GFLG.OR.
     2  GRID(IJ1).EQ.GFLG.OR.GRID(I1J1).EQ.GFLG)) GO TO 70
C
C MESH BLOCK CONTOURABLE
C
      FG=1
      IJS=MIN0(IJS,IJ)
      IJE=MAX0(IJE,IJ)
C
C CHECK AND SET LEFT EDGE (2)
C
      IF (ROWLG) GO TO 50
      IF (GRID(IJ1).GT.GRID(IJ)) FG=FG+32
      ROWLG=.TRUE.
C
C CHECK AND SET LOWER EDGE (1)
C
   50 L=0
      IF (GRID(IJ).GT.GRID(I1J)) L=4
      IF (.NOT.COLLG) L=L+4
      FG=FG+L
C
C CHECK AND SET GRADIENT
C
      IF (GRAD.EQ.0.) GO TO 60
      CALL COORD(I+1,J,XLR,YLR)
      CALL COORD(I+1,J+1,XUR,YUR)
      IF ((GRID(IJ)-GRID(I1J))**2.GT.
     1       ((XLL-XLR)**2+(YLL-YLR)**2)*GRAD
     2 .OR.(GRID(IJ)-GRID(IJ1))**2.GT.
     3       ((XLL-XUL)**2+(YLL-YUL)**2)*GRAD
     4 .OR.(GRID(IJ1)-GRID(I1J1))**2.GT.
     5       ((XUL-XUR)**2+(YUL-YUR)**2)*GRAD
     6 .OR.(GRID(IJ1)-GRID(I1J1))**2.GT.
     7       ((XUR-XLR)**2+(YUR-YLR)**2)*GRAD
     8 .OR.(GRID(IJ)-GRID(I1J1))**2.GT.
```

```
      9        ((XLL-XUR)**2+(YLL-YUR)**2)*GRAD
      1  .OR.(GRID(I1J)-GRID(IJ1))**2.GT.
      2        ((XUL-XLR)**2+(YUL-YLR)**2)*GRAD)  FG=FG+2
       XLL=XLR
       YLL=YLR
       XUL=XUR
       YUL=YUR
C
C DETERMINE BMIN,BMAX
C
   60 BMIN=AMIN1(BMIN,GRID(IJ))
      BMAX=AMAX1(BMAX,GRID(IJ))
      L=IJ/4
      FLAGS(L)=FLAGS(L).OR.(FG*LMULT(IJ.AND.3))
      GO TO 110
C
C MESH BLOCK NON-CONTOURABLE
C
C CHECK AND SET SIDE 3 AND 4 FLAGS
C
   70 KK=.FALSE.
      IF (.NOT.ROWLG) GO TO 80
      IF (GRID(IJ).GT.GRID(IJ1)) CALL SET(FLAGS,IJ-1,128)
      KK=.TRUE.
   80 IF (.NOT.COLLG) GO TO 90
      IF (GRID(IJ).LT.GRID(I1J)) CALL SET(FLAGS,IJN1,64)
      KK=.TRUE.
   90 IF (.NOT.(COLLRC.OR.KK)) GO TO 100
      BMIN=AMIN1(BMIN,GRID(IJ))
      BMAX=AMAX1(BMAX,GRID(IJ))
  100 ROWLG=.FALSE.
      FG=0
C
C END OF MESH
C
  110    IJ=I1J
      I1J=I1J+1
      IJ1=I1J1
      I1J1=I1J1+1
      COLLRC=CCC
      IJN1=IJN1+1
  120 CONTINUE
C
  130 CONTINUE
C
      IJS=IJS/4
      IJE=IJE/4
      IJSI=(IJS+NCOL)/4
      IJEI=IJE
      RETURN
C
      END
```

```
      SUBROUTINE SCAN(GRID,FLAGS,COORD)
C
C  SCANS GRID FOR UNDRAFTED CONTOUR LEVEL.
C
      DIMENSION GRID(0/0),FLAGS(0/0),COORD(1)
      INTEGER FLAGS
C
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
C
C  SET INTERIOR FLAGS.
      IF (IJSI.LT.0) GO TO 20
      DO 10 IJ=IJSI,IJEI
   10 FLAGS(IJ)=FLAGS(IJ).OR.
     1   ((FLAGS(IJ).AND."4004004004)*2)
C
C  SETUP FOR EDGE SCAN
C
   20 IF (IJS.LT.0) GO TO 180
      ASSIGN 40 TO ISWA
      ASSIGN 160 TO ISWB
      EDGE=.TRUE.
      IJA=IJS
      IJB=IJE
      IJSNEW=-1
      MASK="360360360360
      MASKB="360
      MASKD="20
      MASKF=1
   30 IJ4B=IJA*4+3
C
C  BASIC WORD SCAN
C
      DO 170 IJ=IJA,IJB
      LFLAG=FLAGS(IJ)
      IF ((LFLAG.AND.MASK).EQ.0) GO TO ISWB,(160,150)
      IF (IJSNEW.LT.0) IJSNEW=IJ
      IJENEW=IJ
      IJ4=IJ4B
      IJ4N=IJ4+NCOL
C
C  SUB WORD SCAN
C
      DO 140 K=1,4
      IF ((LFLAG.AND.MASKB).EQ.0) GO TO 130
      GO TO ISWA,(40,110)
C
C  RIGHT EDGE
C
   40 IF ((LFLAG.AND."200).EQ.0) GO TO 60
      IF (CONT.LE.GRID(IJ4+1)) GO TO 50
```

```
      CALL RESET(FLAGS,IJ4,"200)
      GO TO 80
   50 IF (CONT.LE.GRID(IJ4N+1)) GO TO 60
      CALL TRACE(GRID,FLAGS,COORD,4,MASKF)
      GO TO 80
C
C  TOP EDGE
C
   60 IF ((LFLAG.AND."100).EQ.0) GO TO 80
      IF (CONT.LE.GRID(IJ4N+1)) GO TO 70
      CALL RESET(FLAGS,IJ4,"100)
      GO TO 100
   70 IF (CONT.LE.GRID(IJ4N)) GO TO 80
      CALL TRACE(GRID,FLAGS,COORD,3,MASKF)
      GO TO 100
C
C  LEFT EDGE
C
   80 IF ((LFLAG.AND."40).EQ.0) GO TO 100
      IF (CONT.LE.GRID(IJ4N)) GO TO 90
      CALL RESET(FLAGS,IJ4,"40)
      GO TO 130
   90 IF (CONT.LE.GRID(IJ4)) GO TO 100
      CALL TRACE(GRID,FLAGS,COORD,2,MASKF)
      GO TO 130
C
C  BOTTOM EDGE AND INTERIOR SCAN
C
  100 IF ((LFLAG.AND."20).EQ.0) GO TO 130
  110 IF (CONT.LE.GRID(IJ4)) GO TO 120
      CALL RESET(FLAGS,IJ4,MASKD)
      GO TO 130
  120 IF (CONT.GT.GRID(IJ4+1))
     1 CALL TRACE(GRID,FLAGS,COORD,1,MASKF)
C
  130 LFLAG=LFLAG/512
      IJ4=IJ4-1
      IJ4N=IJ4N-1
  140 CONTINUE
C
      GO TO 160
C
C  INTERIOR RANGE SET
C
  150 IF ((LFLAG.AND."004004004004).EQ.0) GO TO 160
      IF (IJSNEW.LT.0) IJSNEW=IJ
      IJENEW=IJ
  160 IJ4B=IJ4B+4
C
  170 CONTINUE
C
C  FIRST TIME?
C
      IF (.NOT.EDGE) GO TO 190
```

```
C
C  UPDATE EDGE RANGE
C
      IJS=IJSNEW
      IJE=IJENEW
  180 IF (IJSI.LT.0) GO TO 200
C
C  SETUP FOR INTERIOR SCAN
C
      EDGE=.FALSE.
      ASSIGN 110 TO ISWA
      ASSIGN 150 TO ISWB
      IJSNEW=-1
      IJA=IJSI
      IJB=IJEI
      MASK="010010010010
      MASKB="010
      MASKD="4
      MASKF=MASKB
      GO TO 30
C
C  UPDATE INTERIOR BOUNDARY INDEXES
C
  190 IJSI=IJSNEW
      IJEI=IJENEW
C
C  END OF SCAN
C
  200 RETURN
C
      END
```

```
      SUBROUTINE TRACE(GRID,FLAGS,COORD,ISIDE,MASK)
C
C  FOLLOWS CONTOUR THROUGH GRID UNTIL
C  EDGE OR CLOSURE FOUND.
C
      DIMENSION GRID(0/0),FLAGS(0/0)
      INTEGER FLAGS
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
C
C TRACE AND PLOT CONTOUR THROUGH GRID
C
C
C GRID INDEXING
C                     SIDE 3
C          I01 +             + I11
C
C      SIDE 2                   SIDE 4
C
C          I00 +             + I10
C                     SIDE 1
C
C IN-- ENTRANCE SIDE
C HIGH-EQUAL POINT(INN) ALWAYS ON LEFT WHEN LOOKING ALONG
C     CONTOUR LINE
C IOL-- POINT OPPOSITE ENTRANCE SIDE ON LEFT
C IOR-- POINT OPPOSITE ENTRANCE SIDE ON RIGHT
C
      DIMENSION X(100),Y(100)
C
C SET UP TRACING START
C
      LOGICAL BIT,POST
      POST=PRIME
      I=MOD(IJ4,NCOL)
      J=IJ4/NCOL
      NPTS=1
      IC=0
      I00=IJ4
      I10=I00+1
      I01=I00+NCOL
      I11=I01+1
      GO TO (10,20,30,40),ISIDE
   10 CALL RESET(FLAGS,I00,"10)
      FRACT=(CONT-GRID(I00))/(GRID(I10)-GRID(I00))
      CALL COORD(I,J,XA,YA)
      CALL COORD(I+1,J,XB,YB)
      IOL=I01
      IOR=I11
      IN=1
      GO TO 50
```

```
   20 FRACT=(CONT-GRID(I00))/(GRID(I01)-GRID(I00))
      CALL COORD(I,J,XA,YA)
      CALL COORD(I,J+1,XB,YB)
      IOL=I11
      IOR=I10
      IN=2
      GO TO 50
   30 FRACT=(CONT-GRID(I01))/(GRID(I11)-GRID(I01))
      CALL COORD(I,J+1,XA,YA)
      CALL COORD(I+1,J+1,XB,YB)
      IOL=I10
      IOR=I00
      IN=3
      GO TO 50
   40 FRACT=(CONT-GRID(I10))/(GRID(I11)-GRID(I10))
      CALL COORD(I+1,J,XA,YA)
      CALL COORD(I+1,J+1,XB,YB)
      IOL=I00
      IOR=I01
      IN=4
C
C START TRACE LOOP
C
   50 X(NPTS)=XA+(XB-XA)*FRACT
      Y(NPTS)=YA+(YB-YA)*FRACT
      IF (.NOT.PRIME.AND.BIT(FLAGS,I00,2)) GO TO 80
      NPTS=NPTS+1
      IF (NPTS.LE.100) GO TO 90
      IF (POST) GO TO 60
      CALL LINE(X,Y,100,IC,LINET)
      POST=PRIME
      GO TO 70
   60 CALL LABEL(X,Y,100,IC,POST)
   70 NPTS=1
      IC=1
      GO TO 90
   80 IF (NPTS.EQ.1) GO TO 90
      CALL LINE(X,Y,NPTS,IC,LINET)
      IC=0
      NPTS=1
C
C DETERMINE EXIT SIDE
C
   90 K=IN+2
      IF (GRID(IOR).LT.CONT) K=K-1
      IF (GRID(IOL).LT.CONT) K=K-2
      IF (K-IN) 110,100,120
C
C SADDLE DECISION (DAYHOFF)
C
  100 IF ((GRID(I00)+GRID(I10)+GRID(I01)+GRID(I11))
     1  *0.25.LT.CONT) GO TO 120
      K=IN+2
      GO TO 120
```

```
  110 K=IN
C
C COMPUTE SIDE BRANCH
C
  120 GO TO (140,150,160,130,140,150),K
C
C EXIT BOTTOM -- SIDE 1
C
  130 FRACT=(CONT-GRID(I00))/(GRID(I10)-GRID(I00))
      CALL COORD(I,J,XA,YA)
      CALL COORD(I+1,J,XB,YB)
      IF (J.EQ.0) GO TO 170
      I01=I00
      I00=I00-NCOL
      IF (.NOT.BIT(FLAGS,I00,1)) GO TO 170
      IN=3
      I11=I10
      I10=I10-NCOL
      IOL=I10
      IOR=I00
      J=J-1
      GO TO 50
C
C EXIT LEFT -- SIDE 2
C
  140 FRACT=(CONT-GRID(I00))/(GRID(I01)-GRID(I00))
      CALL COORD(I,J,XA,YA)
      CALL COORD(I,J+1,XB,YB)
      IF (I.EQ.0) GO TO 170
      I10=I00
      I00=I00-1
      IF (.NOT.BIT(FLAGS,I00,1)) GO TO 170
      IN=4
      I11=I01
      I01=I01-1
      IOL=I00
      IOR=I01
      I=I-1
      GO TO 50
C
C EXIT TOP -- SIDE 3
C
  150 FRACT=(CONT-GRID(I01))/(GRID(I11)-GRID(I01))
      CALL COORD(I,J+1,XA,YA)
      CALL COORD(I+1,J+1,XB,YB)
      IF (.NOT.BIT(FLAGS,I01,MASK)) GO TO 170
      CALL RESET(FLAGS,I01,"10)
      IN=1
      I00=I01
      I10=I11
      I01=I01+NCOL
      I11=I11+NCOL
      IOL=I01
      IOR=I11
```

```
      J=J+1
      GO TO 50
C
C EXIT RIGHT -- SIDE 4
C
  160 FRACT=(CONT-GRID(I10))/(GRID(I11)-GRID(I10))
      CALL COORD(I+1,J,XA,YA)
      CALL COORD(I+1,J+1,XB,YB)
      IF (.NOT.BIT(FLAGS,I10,1)) GO TO 170
      IN=2
      I00=I10
      I01=I11
      I10=I10+1
      I11=I11+1
      IOL=I11
      IOR=I10
      I=I+1
      GO TO 50
C
C END OF TRACE LOOP
C
  170 X(NPTS)=XA+(XB-XA)*FRACT
      Y(NPTS)=YA+(YB-YA)*FRACT
      IF (NPTS.LE.1.AND.IC.EQ.0) GO TO 190
C
C DUMP RESIDUAL X-Y TO PLOTTER
C
      IF (POST) GO TO 180
      CALL LINE(X,Y,NPTS,IC,LINET)
      GO TO 190
  180 CALL LABEL(X,Y,NPTS,IC,POST)
  190 RETURN
C
C END OF TRACING
C
      END
```

```
      SUBROUTINE LABEL(X,Y,NPTS,ICC,POST)
C
C  LABEL SCANS CONTOUR LINE ARRAY LOOKING FOR
C  A STRAIGHT SEGMENT.
C  IF FOUND,  THE CONTOUR IS LABELLED.
C
      DIMENSION X(1),Y(1),FMT(1)
      LOGICAL POST
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
      DOUBLE PRECISION CHARST(3)
      LOGICAL NOLAB
      EQUIVALENCE (F,SUM),(ISA,YA)
      IF (NOLAB) GO TO 50
C
C SETUP AND SCAN
C
      IS=2
      ISA=3
      SUM=0.0
      IN1=2
      IE=NPTS-1
      IF (IE.LT.3) GO TO 50
      XB=X(IS)
      YB=Y(IS)
      DO 40 I=ISA,IE
      SUM=SUM+SQRT((X(I)-X(IN1))**2+(Y(I)-Y(IN1))**2)
   10 DIST=(X(I)-XB)**2 +  (Y(I)-YB)**2
      IF (DIST.LT.WDIST2) GO TO 30
      DIST=SQRT(DIST)
      IF ((SUM/DIST).GT.1.02) GO TO 20
      IE=I
      GO TO 60
   20 IS=IS+1
      SUM=SUM-SQRT((X(IS)-XB)**2+(Y(IS)-YB)**2)
      XB=X(IS)
      YB=Y(IS)
      IF (IS.LT.I) GO TO 10
   30 IN1=I
   40 CONTINUE
C
C  CAN'T FIND SPOT, CONTINUE LINE WITHOUT LABELING.
   50 CALL LINE(X,Y,NPTS,ICC,LINET)
      GO TO 70
C
C  PLOTTABLE LOCATION.
   60 CALL LINE(X,Y,IS,ICC,LINET)
      XA=X(IE)-X(IN1)
      YA=Y(IE)-Y(IN1)
      F=1.-(DIST-WDIST)/SQRT(XA*XA+YA*YA)
      XA=F*XA + X(IN1)
```

```
      YA=F*YA + Y(IN1)
      PHI=ATAN2(YA-YB,XA-XB)
      IF (ABS(PHI).GT.1.5707963)
     1  PHI=PHI-SIGN(3.1415927,PHI)
      CALL CHAR(0.5*(XA+XB),0.5*(YA+YB),CHARST,NCHF,
     1 2,SIZE,PHI,CCOR,0.)
      CALL LINE(XA,YA,1,0,LINET)
      CALL LINE(X(IE),Y(IE),NPTS-IE+1,1,LINET)
      POST=.FALSE.
C
   70 RETURN
C
C  ENTRY TO SET LABELING CONSTANTS.
C  MUST BE CALLED BEFORE SETLAB.
      ENTRY SETCON(FMT,SIZ,NCHA)
      NCHAR=NCHA
      SIZE=SIZ
      NOLAB=.NOT.(NCHAR.GT.0.AND.SIZE.GT.0.)
      RETURN
C
C  ENTRY TO ESTABLISH LABELING CHARACTER STRING.
C  MUST BE CALLED BEFORE LABEL.
      ENTRY SETLAB
      IF (NOLAB) GO TO 80
      ENCODE(30,FMT,CHARST) CONT
      NCHF=NCHAR
      CALL LFJUST(CHARST,NCHF)
      WDIST=(NCHF+1)*SIZE
      WDIST2=WDIST*WDIST
      CCOR=-.5*WDIST+SIZE
   80 RETURN
      END
```

```
      LOGICAL FUNCTION BIT(LF,I,MASK)
C
C  RETURNS TRUE VALUE IF ANY BIT(S)
C  OF LF ARRAY MATCH MASK BITS.
C
      DIMENSION LF(0/0)
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
      BIT=((LF(I/4)/LMULT(I.AND.3)).AND.MASK).NE.0
      RETURN
      END
```

```
      SUBROUTINE SET(LF,I,MASK)
C
C  SETS BITS OF LF TO SETTING
C  OF MASK BITS.
C
      DIMENSION LF(0/0)
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
      K=I/4
      LF(K)=LF(K).OR.(MASK*LMULT(I.AND.3))
      RETURN
      END
```

```
      SUBROUTINE RESET(LF,I,MASK)
C
C  RESETS BITS OF LF CORRESPONDING
C  TO BITS IN MASK.
C
      DIMENSION LF(0/0)
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
      K=I/4
      LF(K)=LF(K).AND.(MASK*LMULT(I.AND.3)
     1  .XOR."777777777777)
      RETURN
      END
```

```
      .
      BLOCK DATA BLKCON
      DOUBLE PRECISION TITLE1,TITLE2,TITLE3,FMTX,FMTY
      COMMON /SEXY/
     1  TITLE1(7),TITLE2(8),TITLE3(8),FMTX(2),FMTY(2),
     2 XX(2),YY(2),XP(4),YP(4),IPLOTR,SIZEL,NCHARX,NCHARY,
     3 SIZEX,SIZEY,ADELX,ADELY,PLLX,PLLY,LINTX,LINTY,
     4 XSCALE,YSCALE,XXSCAL,YYSCAL
C
      COMMON /CONTRC/ CMIN,CMAX,DCVAL,NCVAL,NSEC,GRADI
      COMMON /CONCOM/ NCOL,NROW,BMIN,BMAX,
     1 GRAD,GFLG,IJS,IJE,IJSI,IJEI,PRIME,IJ4,CONT,
     2 FLTMAX,LMULT(0/3),IDASHS,LINET
C
      LOGICAL PRIME
      DATA LMULT/134217728,262144,512,1/
      DATA CMIN,CMAX,DCVAL,NCVAL,NSEC,SIZEL/3*0.,2*0,0.07/
      DATA GRADI,GFLG,FLTMAX/30.,2*"377777777777/
      DATA TITLE2,TITLE3/16*'          '/
      DATA IPLOTR,NCHARX,NCHARY,LINTX,LINTY/5*0/
      DATA SIZEX,SIZEY,ADELX,ADELY,PLLX,PLLY,
     1 XSCALE,YSCALE/8*0./
      DATA XP(2),XP(2)/2*0./
      END
```

APPENDIX C.

LISTING OF PROGRAM SPLN2D

```
C
C  TWO DIMENSIONAL SPLINE INTERPOLATION SYSTEM
C
C  CODED BY
C      GERALD IAN EVENDEN
C      U. S. GEOLOGICAL SURVEY
C      DENVER FEDERAL CENTER
C      DENVER, COLORADO  80225
C
C
      DIMENSION WORK(20000)
C
C  WHEN COMPILED ON THE DEC SYSTEM-10 WITH THE FORTRAN-10
C  COMPILER THE PROGRAM REQUIRES APPROXIMATELY
C          12 + NWORK/1000 K-WORDS OF MEMORY.
C
C  TO COMPILE A VERSION TO HANDLE LARGER GRIDS, MERELY
C  CHANGE THE VALUE OF NWORK AND THE DIMENSION OF WORK.
C
C
C  NOTE... THIS PROGRAM CONTAINS SEVERAL NON-ANSI FORTRAN
C  STATEMENTS.  ALTHOUGH SEVERAL OF THE CONSTRUCTS MAY BE
C  AVAILABLE ON OTHER COMPUTER SYSTEMS, CARE MUST BE TAKEN
C  IN THE TRANSPORTATION OF THIS PROGRAM.
C
C
      NAMELIST /PARMS/ IDELX,IDELY,XOFF,YOFF,XDEL,YDEL
     1 ,NCOLO,NROWO,IDER
      DOUBLE PRECISION TITLE,PGMID
      COMMON TITLE(7),PGMID,NCOL,NROW,NZ,XO,DELX,
     1 YO,DELY,IYP,IUP,IPP,IQP,ISP,IWP,IRP,IO,IOUT
     2 ,DVAL,NFX,NFY,NROWO,NCOL1,NROW1,IDER
      EQUIVALENCE (NFX,IDELX),(NFY,IDELY)
C
      DOUBLE PRECISION NAME,BLANK
      DIMENSION INAME(3)
      EQUIVALENCE (INAME(1),NAME)
C
      DATA NWORK/20000/
      DATA INAME(3),BLANK/0,'          '/
      DATA ITTY/1/
C
      DVAL="377777777777
      IO=10
      IOUT=11
      IDER=0
C
      OPEN (UNIT=ITTY,DEVICE='TTY',MODE='ASCII')
C
   10 WRITE(ITTY,20)
```

```
   20 FORMAT(' ENTER INPUT FILE NAME.EXT:',$)
      READ(ITTY,30) NAME
   30 FORMAT(A10)
      IF (NAME.NE.BLANK) GO TO 60
C
C  GET FROM SAVE AREA
      WRITE(ITTY,280)
      READ(ITTY,30) NAME
      IF (NAME.NE.BLANK) GO TO 50
      WRITE(ITTY,40)
   40 FORMAT(' %NO ORIGINAL OR SAVE FILE GIVEN')
      GO TO 490
   50 ISP=-1
      GO TO 70
C
C GET NEW INPUT
   60 IF (NAME.EQ.'EXIT     ') GO TO 490
      ISP=0
   70 OPEN(UNIT=IO,DEVICE='DSK',ACCESS='SEQIN',
     1 MODE='BINARY',FILE=NAME)
      READ(IO,END=470) TITLE,PGMID,NCOL,NROW,NZ,
     1XO,DELX,YO,DELY
      IF (ISP.GE.0) GO TO 80
      IF (PGMID.EQ.'SPLCOEFF**'.AND.NZ.EQ.16) GO TO 110
      GO TO 90
   80 IF (NZ.EQ.1) GO TO 110
   90 WRITE(ITTY,100)
  100 FORMAT(' %INVALID INPUT FILE')
      GO TO 490
C
C  ALLOCATE MEMORY
  110 IYP=NCOL+1
      IF (ISP.LT.0) GO TO 140
      IUP=IYP+NROW
      NXY=NCOL*NROW
      IPP=IUP+NXY
      IQP=IPP+NXY
      ISP=IQP+NXY
      IWP=ISP+NXY
      IF (IWP+6*MAX0(NCOL,NROW).LE.NWORK) GO TO 150
  120 WRITE(ITTY,130)
  130 FORMAT(' %INPUT TOO LARGE')
      GO TO 490
  140 IUP=IYP+2
      NXY=16*NCOL
      IWP=IUP+NXY
      IRP=IWP+NXY
      GO TO 160
C
  150 IRP=IWP+NCOL*16
  160 IF (IRP.GE.NWORK) GO TO 120
      IF (DELX.NE.0.) GO TO 170
      READ(IO,END=470) (WORK(I),I=1,NCOL)
      GO TO 190
```

```
  .
  170 F=0.
      DO 180 I=1,NCOL
      WORK(I)=X0+F*DELX
  180 F=F+1.
C
C  INPUT GRID
  190 IF (ISP.LT.0) GO TO 290
      J=IUP-1
      K=IYP
      DO 220 I=1,NROW
      READ(IO,END=200) WORK(K), (WORK(J+L),L=1,NCOL)
      GO TO 210
  200 NROW=I-1
      GO TO 230
  210 J=J+NCOL
      K=K+1
  220 CONTINUE
C
  230 IF (NCOL.GT.2.AND.NROW.GT.2) GO TO 250
      WRITE(ITTY,240)
  240 FORMAT(' %NCOL OR NROW LESS THAN 3')
      GO TO 490
C
C  GENERATE Y VALUES, IF NECESSARY
  250 IF (DELY.EQ.0.) GO TO 270
      F=0.
      J=IYP+NROW-1
      DO 260 I=IYP,J
      WORK(I)=Y0+F*DELY
  260 F=F+1.
C
C  EVERYTHING LOADED, GENERATE COEFFICIENT ARRAYS
  270 CLOSE(UNIT=IO)
      CALL SPLN2A(WORK(IUP),NCOL,NROW,DELX,DELY,
     1 WORK,WORK(IYP),DVAL,WORK(IPP),WORK(IQP),
     2 WORK(ISP),WORK(IWP))
C
C  SAVE GOODIES.
      WRITE(ITTY,280)
  280 FORMAT(' ENTER COEFFICIENT FILE NAME.EXT:',$)
      READ(ITTY,30) NAME
      IF (NAME.EQ.BLANK) GO TO 290
      OPEN(UNIT=IO,DEVICE='DSK',ACCESS='SEQOUT',
     1 MODE='BINARY',FILE=NAME)
      CALL PUTCOF(WORK,WORK(IYP),WORK(IWP),NCOL)
      CLOSE(UNIT=IO)
      GO TO 10
C
C  GET OUTPUT FILE NAME
  290 WRITE(ITTY,300)
  300 FORMAT(' ENTER INTERPOLATED FILE NAME.EXT:',$)
      READ(ITTY,30) NAME
      IF (NAME.EQ.BLANK) GO TO 10
      OPEN(UNIT=IOUT,DEVICE='DSK',ACCESS='SEQOUT',
```

```
     1 MODE='BINARY',FILE=NAME)
      PGMID='SPLINE-INT'
      IDELX=0
      IDELY=0
      XOFF=0.
      YOFF=0.
      XDEL=0.
      YDEL=0.
      WRITE(ITTY,310)
  310 FORMAT(' ENTER PARAMETERS:')
      READ(ITTY,PARMS,END=450)
      IF (XDEL.NE.0.) GO TO 360
      IF (YDEL.NE.0.) GO TO 370
      IF (IDELX.GT.0) GO TO 340
      IF (IDELY.GT.0) GO TO 330
      WRITE(ITTY,320)
  320 FORMAT(' %INVALID SUBDIVISION PARAMETERS')
      GO TO 490
C
  330 IDELX=IDELY
      GO TO 350
  340 IF (IDELY.LE.0) IDELY=IDELX
  350 NCOLO=(NCOL-1)*IDELX+1
      NROWO=(NROW-1)*IDELY+1
      DELX=DELX/IDELX
      DELY=DELY/IDELY
      GO TO 400
C
C  EQUAL GRID
  360 IF (YDEL.NE.0.) GO TO 380
      YDEL=XDEL
      GO TO 380
  370 IF (XDEL.EQ.0.) XDEL=YDEL
  380 DELX=XDEL
      DELY=YDEL
      XO=XO+XOFF
      YO=YO+YOFF
      IDELX=0
      IF (NCOLO.GT.0.AND.NROWO.GT.0) GO TO 400
      WRITE(ITTY,390)
  390 FORMAT(' %NCOLO OR NROWO LESS THAN 1')
      GO TO 440
C
C  CHECK WORK OVERFLOW
  400 IF (IRP+NCOLO-1.LT.NWORK) GO TO 420
      WRITE(ITTY,410)
  410 FORMAT(' %NO MEMORY FOR NEW GRID GENERATION')
      GO TO 440
C
C  WRITE HEADER
  420 NZ=1
      WRITE(IOUT) TITLE,PGMID,NCOLO,NROWO,NZ,XO,DELX,YO,DELY
      NCOL1=NCOL-1
      NROW1=NROW-1
```

```
      .
      IF (IDELX.NE.0) GO TO 430
      CALL DELGR(WORK,WORK(IYP),WORK(IRP),WORK(IWP),
     1  WORK(IUP),NCOLD)
      GO TO 440
  430 CALL FINEGR(WORK,WORK(IYP),WORK(IRP),WORK(IWP)
     1   ,WORK(IUP),NCOLD)
C
C  DONE
  440 CLOSE(UNIT=IOUT)
      CLOSE(UNIT=IO)
      GO TO 10
C
C
  450 WRITE(ITTY,460)
  460 FORMAT(' %EOF ON NAMELIST READ')
      GO TO 490
  470 WRITE(ITTY,480)
  480 FORMAT(' %EOF ON HEADER OR COLUMN SPACING')
C
  490 CLOSE(UNIT=ITTY)
      CLOSE(UNIT=IO)
      CLOSE(UNIT=IOUT)
      STOP
      END
```

```
      SUBROUTINE FINEGR(X,Y,R,C,W,NCOLD)
C
C  SUBDIVIDE EACH ORIGINAL GRID BLOCK INTO
C  EQUAL SECTIONS.
      DIMENSION W(1),X(1),Y(1),R(NCOLD),C(16,1),B(4),DR(4)
C
      DOUBLE PRECISION TITLE,PGMID
      COMMON TITLE(7),PGMID,NCOL,NROW,NZ,XO,DELX,
     1 YO,DELY,IYP,IUP,IPP,IQP,ISP,IWP,IRP,IO,IOUT
     2 ,DVAL,NFX,NFY,NROWO,NCOL1,NROW1,IDER
C
      IF (IDER) 10,20,30
   10 ILS=2
      GO TO 40
   20 ILS=1
      GO TO 40
   30 ILS=3
   40 YFRACT=1./FLOAT(NFY)
      XFRACT=1./FLOAT(NFX)
      NFX1=NFX-1
      NFY1=NFY+1
      IF (DELX.NE.0.) GO TO 70
C
C  GENERATE AND WRITE X ARRAY, IF NECESSARY
      MO=2
      R(1)=X(1)
      DO 60 I=1,NCOL1
      XDEL=(X(I+1)-X(I))*XFRACT
      XV=XDEL
      DO 50 J=1,NFX
      R(MO)=X(I)+XV
      MO=MO+1
      XV=XV+XDEL
   50 CONTINUE
   60 CONTINUE
      WRITE(IOUT) R
C
C  SET R TO DVALS
   70 DO 80 I=1,NCOLD
   80 R(I)=DVAL
C
C  ROW LOOP
      DO 320 JJ=1,NROW1
      JJJ=JJ
      IF (ISP.LT.0) GO TO 90
      CALL SPLN2B(C,JJ)
      GO TO 100
   90 CALL GETSPL(C,JJJ,Y,W,NCOL)
  100 YVA=Y(JJJ)
      YDEL=(Y(JJJ+1)-YVA)*YFRACT
      YV=0.
C
C  ROW SUBDIVISION
      DO 310 J=1,NFY1
```

```
      MO=2
      MOE=1
C
C  COLUMN LOOP
      DO 300 I=1,NCOL1
      IF (J.EQ.1) GO TO 260
      IF (C(1,I).EQ.DVAL) GO TO 270
  110 XDEL=(X(I+1)-X(I))*XFRACT
      IF (IDER) 120,160,140
  120 DO 130 L=1,4
  130 DR(L)=C(L+4,I)+YV*(2.*C(L+8,I)+3.*YV*C(L+12,I))
      GO TO 160
  140 DO 150 L=1,4
  150 DR(L)=2.*C(L+8,I)+6.*YV*C(L+12,I)
  160 DO 170 L=ILS,4
  170 B(L)=((YV*C(L+12,I)+C(L+8,I))*YV+C(L+4,I))*YV+C(L,I)
      IF (IDER) 180,200,190
  180 IF (R(MOE).EQ.DVAL)
     1  R(MOE)=SQRT(B(2)**2 + DR(1)**2)
      GO TO 210
  190 IF (R(MOE).EQ.DVAL)
     1  R(MOE)=-(2.*B(3) + DR(1))
      GO TO 210
  200 IF (R(MOE).EQ.DVAL) R(MOE)=B(1)
  210 MOE=MO+NFX1
      XV=0.
C
C  FILL WITH VALUES
      DO 250 L=MO,MOE
      XV=XV+XDEL
      IF (IDER) 230,220,240
  220 R(L)=((B(4)*XV+B(3))*XV+B(2))*XV+B(1)
      GO TO 250
  230 DRX=B(2)+XV*(2.*B(3)+XV*3.*B(4))
      DRY=DR(1)+XV*(DR(2)+XV*(DR(3)+XV*DR(4)))
      R(L)=SQRT(DRX*DRX + DRY*DRY)
      GO TO 250
  240 R(L)=-(2.*B(3)+DR(1)+XV*(6.*B(4)+DR(2)+XV*(
     1  DR(3)+XV*DR(4))))
  250 CONTINUE
      GO TO 290
C
C  FIRST ROW CHECK
  260 IF ((R(MOE).EQ.DVAL.OR.
     1 R(MO).EQ.DVAL).AND.C(1,I).NE.DVAL) GO TO 110
      MOE=MO+NFX1
      GO TO 290
C
C  INSERT DVALS
  270 MOE=MO+NFX1
      DO 280 L=MO,MOE
  280 R(L)=DVAL
  290 MO=MOE+1
  300 CONTINUE
```

```
      .
      IF (J.EQ.NFY1) GO TO 310
      WRITE(IOUT) YVA,R
      IF (J.NE.NFY1) R(1)=DVAL
      YV=YV+YDEL
      YVA=YVA+YDEL
  310 CONTINUE
  320 CONTINUE
      WRITE(IOUT) YVA,R
      RETURN
      END
```

```
      SUBROUTINE DELGR(X,Y,R,C,W,NCOLO)
C
C  SUBDIVIDE GRID INTO EQUALLY SPACED GRID.
C
      DIMENSION W(1),X(1),Y(1),R(NCOLO),C(16,1),B(4),DR(4)
C
      DOUBLE PRECISION TITLE,PGMID
      COMMON TITLE(7),PGMID,NCOL,NROW,NZ,XO,DELX,
     1 YO,DELY,IYP,IUP,IPP,IQP,ISP,IWP,IRP,IO,IOUT
     2 ,DVAL,NFX,NFY,NROWO,NCOL1,NROW1,IDER
C
      IF (IDER) 10,20,30
   10 ILS=2
      GO TO 40
   20 ILS=1
      GO TO 40
   30 ILS=3
   40 JJ=1
      JJJ=1
      IF (ISP.LT.0) CALL GETSPL(C,JJJ,Y,W,NCOL)
      FY=0.
C
C  ROW LOOP
      DO 280 J=1,NROWO
      YVA=YO+FY*DELY
C
C  CHECK Y RANGE
      IF (DELY.LT.0.) GO TO 60
      IF (YVA.LT.Y(JJJ)) GO TO 250
   50 IF (YVA.LE.Y(JJJ+1)) GO TO 80
      IF (JJ.GE.NROW1) GO TO 250
      JJ=JJ+1
      JJJ=JJ
      IF (ISP.LT.0) CALL GETSPL(C,JJJ,Y,W,NCOL)
      GO TO 50
   60 IF (YVA.GT.Y(JJJ)) GO TO 250
   70 IF (YVA.GE.Y(JJJ+1)) GO TO 80
      IF (JJ.GE.NROW1) GO TO 250
      JJ=JJ+1
      JJJ=JJ
      IF (ISP.LT.0) CALL GETSPL(C,JJJ,Y,W,NCOL)
      GO TO 70
C
C  SET COEFFICIENTS FOR ROW
   80 YV=YVA-Y(JJJ)
      IF (ISP.GE.0) CALL SPLN2B(C,JJ)
      IL=0
      II=1
      FX=0.
C
C  COLUMN LOOP
      DO 240 I=1,NCOLO
      XVA=XO+FX*DELX
C
```

```
C  CHECK RANGE
      IF (DELX.LT.0.) GO TO 100
      IF (XVA.LT.X(II)) GO TO 220
   90 IF (XVA.LE.X(II+1)) GO TO 120
      IF (II.GE.NCOL1) GO TO 220
      II=II+1
      GO TO 90
  100 IF (XVA.GT.X(II)) GO TO 220
  110 IF (XVA.GE.X(II+1)) GO TO 120
      IF (II.GE.NCOL1) GO TO 220
      II=II+1
      GO TO 110
C
C  CHECK EMPTY CELL
  120 IF (C(1,II).EQ.DVAL) GO TO 220
C
C  GENERATE B IF NEW CELL
      IF (II.EQ.IL) GO TO 180
      DO 130 L=ILS,4
  130 B(L)=((C(L+12,II)*YV+C(L+8,II))*YV+
     1 C(L+4,II))*YV+C(L,II)
      IL=II
      IF (IDER) 140,180,160
  140 DO 150 L=1,4
  150 DR(L)=C(L+4,II)+YV*(2.*C(L+8,II)+3.*YV*C(L+12,II))
      GO TO 180
  160 DO 170 L=1,4
  170 DR(L)=2.*C(L+8,II)+6.*YV*C(L+12,II)
C
C  GENERATE R VALUE
  180 XV=XVA-X(II)
      IF (IDER) 200,190,210
  190 R(I)=((B(4)*XV+B(3))*XV+B(2))*XV+B(1)
      GO TO 230
  200 DRX=B(2)+XV*(2.*B(3)+3.*XV*B(4))
      DRY=DR(1)+XV*(DR(2)+XV*(DR(3)+XV*DR(4)))
      R(I)=SQRT(DRX*DRX + DRY*DRY)
      GO TO 230
  210 R(I)=-(2.*B(3)+DR(1)+XV*(6.*B(4)+DR(2)+XV*(
     1  DR(3)+XV*DR(4))))
      GO TO 230
C
C  FILL WITH DVAL
  220 R(I)=DVAL
  230 FX=FX+1.
  240 CONTINUE
CC
      GO TO 270
C
C  GENERATE DVAL ROW
  250 DO 260 I=1,NCOLO
  260 R(I)=DVAL
  270 WRITE(IOUT) YVA,R
      FY=FY+1.
```

```
  280 CONTINUE
C
C  ALL DONE
      RETURN
      END
```

```
      SUBROUTINE SPLN2A(U,NX,NY,DX,DY,X,Y,DVAL,P,Q,S,W)
C         BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,
C              DENVER, COLORADO
C--GENERALIZED TWO-DIM. BICUBIC SPLINE COEFFICIENT
C   DETERMINATION WHERE FIRST ENTRY (CALL SPLN2A(...))
C   COMPUTES ARRAYS P,Q,S NEEDED BY SECOND ENTRY
C   (CALL SPLN2B(...))--WHICH IN TURN
C   COMPUTES THE BICUBIC SPLINE COEFFICIENTS FOR ANY
C   SELECTED GRID ROW.
C
C==========================================================
C--FIRST ENTRY (SPLN2A) PARAMETERS:
C      U      = INPUT GRID ARRAY DIMENSIONED U(NX,NY)--ALSO
C                SEE DVAL PARM.
C      NX     = NO. COLUMNS IN ARRAYS U,P,Q,S (NX.GT.2)
C      NY     = NO. ROWS IN ARRAYS U,P,Q,S (NY.GT.2)
C      DX     = EQUAL X-INTERVAL OPTION WHEN DX.GT.0.
C               (USE DUMMY ARRAY X)
C             = 0.0 WHEN UNEQUAL X-VARIABLE ARRAY X IS
C               SUPPLIED (SEE BELOW)
C      DY     = EQUAL Y-INTERVAL OPTION WHEN DY.GT.0.
C              (USE DUMMY ARRAY Y)
C             = 0.0 WHEN UNEQUAL Y-VARIABLE ARRAY Y IS
C               SUPPLIED (SEE BELOW)
C      X      = X-VARIABLE ARRAY DIMENSIONED X(NX)--USED ONLY
C               IF DX=0.0
C                ARRAY X MUST BE IN ASCENDING ORDER WHEN DX=0.0
C      Y      = Y-VARIABLE ARRAY DIMENSIONED Y(NY)--USED ONLY
C               IF DY=0.0
C                ARRAY Y MUST BE IN ASCENDING ORDER WHEN DY=0.0
C      DVAL   = FLAG FOR MISSING DATA--WHEN ANY
C                 U(IX,IY).GE.DVAL.
C                E.G. USE DVAL=1.0E38
C      P,Q,S  = INTERMEDIATE PARTIAL DERIV. ARRAYS DIMN'D
C                P(NX,NY),Q(NX,NY),S(NX,NY)--OUTPUT FROM ENTRY
C                SPLN2A AND INPUT TO ENTRY SPLN2B.
C      W      = WORK VECTOR DIMENSIONED W(6*MAX0(NX,NY))
C
C--SUBROUTINE "SPLIN1" AND "DVALS" CALLED BY SPLN2A.
C  (NOTE: SPLN2A MUST BE CALLED BEFORE ANY CALLS TO SPLN2B)
C
C==========================================================
C--SECOND ENTRY (SPLN2B) PARAMETERS:
C      ENTRY SPLN2B(U,NX,NY,DX,DY,X,Y,DVAL,P,Q,S,C,IY)
C      U,NX,NY,DX,DY,X,Y,DVAL,P,Q,S -- SAME DEFINITIONS AS
C           FOR SPLN2A --
C      C      = OUTPUT BICUBIC COEFF ARRAY DIMENSIONED
C(16,NX)
C              CORRESPONDS TO SELECTED RECTANGLE ROW IY.
C              NOTE: C(1,IX)=DVAL IF ANY CORNER OF RECTANGLE
C              HAS U=DVAL.
C      ------ C(I,IX) IS GIVEN FOR
C              ROW IY WHERE I=1,16 FOR EACH IX=1,NX-1
C              ALSO C(I,NX) IS USED AS WORKING STORAGE,I=1,16.
```

```
C      IY     = SELECTED ROW INDEX TO COMPUTED C ARRAY C
C              1.GE.IY.LE.NY-1).
C
C--SUBROUTINE "GMPRD" CALLED BY SPLN2B
C              (GMPRD IS AN IBM SSP-ROUTINE).
C--ALSO CALLED ARE SUBROUTINES "SETAX" AND "SETAY"
C    (AS REQUIRED)
C
C==========================================================
C--NOTE: NO CHECKS ARE MADE ON CALLING PARAMETERS..
C  THEREFORE THE USER SHOULD INSURE ALL PARAMETERS MEET
C  SPECIFICATIONS AS DESCRIBED ABOVE.
C
      DIMENSION U(NX,NY),P(NX,NY),Q(NX,NY),S(NX,NY),
     1 AX(16),AY(16),AA(16),
     2 X(1),Y(1),W(1),D(2),C(16,NX)
      DATA AX,AY/1.0,4*0.,1.,10*0.,
     1            1.0,4*0.,1.,10*0./
      KD=0
      D(1)=0.0
      D(2)=0.0
      M=MAX0(NX,NY)
      M1=1+M
      M2=M+M1
      M3=M+M2
      M4=M+M3
      M5=M+M4
      DO 20 J=1,NY
      DO 10 I=1,NX
   10 W(I)=U(I,J)
      CALL DVALS(W,NX,DVAL)
      CALL SPLIN1(NX,DX,X,W,P(1,J),W(M2),W(M3),KD,
     1 D,W(M4),W(M5))
   20 CONTINUE
      DO 50 I=1,NX
      DO 30 J=1,NY
   30 W(J)=U(I,J)
      CALL DVALS(W,NY,DVAL)
      CALL SPLIN1(NY,DY,Y,W,W(M1),W(M2),W(M3),KD,
     1 D,W(M4),W(M5))
      DO 40 L=1,NY
   40 Q(I,L)=W(M+L)
   50 CONTINUE
      CALL SPLIN1(NX,DX,X,Q(1,1),S(1,1),W(M2),W(M3),KD,
     1 D,W(M4),W(M5))
      CALL SPLIN1(NX,DX,X,Q(1,NY),S(1,NY),W(M2),W(M3),KD,
     1 D,W(M4),W(M5))
      KD=1
      DO 80 I=1,NX
      DO 60 J=1,NY
   60 W(J)=P(I,J)
      D(1)=S(I,1)
      D(2)=S(I,NY)
      CALL SPLIN1(NY,DY,Y,W,W(M1),W(M2),W(M3),KD,
```

```
     1 D,W(M4),W(M5))
      DO 70 L=1,NY
   70 S(I,L)=W(M+L)
   80 CONTINUE
      RETURN
C$$$$$$ ENTRY SPLN2B $$$$$$
      ENTRY SPLN2B(C,IY)
      IF(KD.EQ.0) GO TO 90
C--KD=1 (AFTER CALL SPLN2A) TO INITILIZE SPLN2B:
      KD=0
      NX1=NX-1
      IF(DX.NE.0.0) CALL SETAX(AX,DX)
      IF(DY.NE.0.0) CALL SETAY(AY,DY)
   90 IY1=IY+1
      IF(DY.EQ.0.0) CALL SETAY(AY,Y(IY1)-Y(IY))
      DO 110 I=1,NX1
      I1=I+1
      IF(U(I,IY).GE.DVAL.OR.U(I1,IY).GE.DVAL.OR.U(I,IY1)
     1 .GE.DVAL.OR.U(I1,IY1).GE.DVAL) GO TO 100
      IF(DX.EQ.0.0) CALL SETAX(AX,X(I1)-X(I))
      AA(1)=U(I,IY)
      AA(2)=P(I,IY)
      AA(3)=U(I1,IY)
      AA(4)=P(I1,IY)
      AA(5)=Q(I,IY)
      AA(6)=S(I,IY)
      AA(7)=Q(I1,IY)
      AA(8)=S(I1,IY)
      AA(9)=U(I,IY1)
      AA(10)=P(I,IY1)
      AA(11)=U(I1,IY1)
      AA(12)=P(I1,IY1)
      AA(13)=Q(I,IY1)
      AA(14)=S(I,IY1)
      AA(15)=Q(I1,IY1)
      AA(16)=S(I1,IY1)
      CALL GMPRD(AX,AA,C(1,NX),4,4,4)
      CALL GMPRD(C(1,NX),AY,C(1,I),4,4,4)
      GO TO 110
  100 C(1,I)=DVAL
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DVALS(Y,LIM,DVAL)
C--PRESETS ROW(OR COLUMN) VECTOR CONTAINING ANY DVAL'S
      DIMENSION Y(1)
      IF(Y(1).GE.DVAL) GO TO 30
   10 DO 20 I=2,LIM
      IF(Y(I).LT.DVAL) GO TO 20
      Y(I)=Y(I-1)
      IF(I.EQ.LIM) GO TO 20
      IF(Y(I+1).LT.DVAL) Y(I)=Y(I+1)
   20 CONTINUE
      RETURN
   30 DO 40 I=2,LIM
      IF(Y(I).GE.DVAL) GO TO 40
      Y(1)=Y(I)
      GO TO 10
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETAX(AX,DD)
C--SET ROWS 2,3 OF ARRAY AX(16) GIVEN SPACING DD
      DIMENSION AX(1)
      DX1=1.0/DD
      DX2=DX1*DX1
      DX3=DX2*DX1
      AX(3)=-3.0*DX2
      AX(4)=2.0*DX3
      AX(7)=-2.0*DX1
      AX(8)=DX2
      AX(11)=-AX(3)
      AX(12)=-AX(4)
      AX(15)=-DX1
      AX(16)=DX2
      RETURN
      END
```

```
      SUBROUTINE SETAY(AY,DD)
C--SET COLUMNS 2,3 OF ARRAY AY(16) GIVEN SPACING DD
      DIMENSION AY(1)
      DY1=1.0/DD
      DY2=DY1*DY1
      DY3=DY2*DY1
      AY(9)=-3.0*DY2
      AY(10)=-2.0*DY1
      AY(11)=-AY(9)
      AY(12)=-DY1
      AY(13)=2.0*DY3
      AY(14)=DY2
      AY(15)=-AY(13)
      AY(16)=DY2
      RETURN
      END
```

```
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,T,D,P,S)
C--ONE DIMENSIONAL CUBIC SPLINE INTERPOLATION
C
C         BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,
C              DENVER, COLORADO
C
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0.
C              (USE DUMMY X HERE),
C              UNEQUAL INTERVALS IF H=0.
C               (X REQUIRED STORAGE)
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS
C                 OF A,B,&C.
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.
C           T= TYPE OF BOUNDARY CONDITION SUPPLIED IN D
C                 ARRAY.
C              USE T=1 IF 1ST DERIVATIVES GIVEN AT END
C              POINTS, OR T=0 IF 2ND DERIVATIVES GIVEN
C              AT END POINTS.
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M
C              RESPECTIVELY.
C           P,S= WORK ARRAYS (EACH DIM=M).
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3
C       FOR I=1,2,...,M-1
C
      REAL   X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL
      INTEGER  T
      IF(T.LT.0.OR.T.GT.1.OR.M.LT.3) GO TO 190
      N=M-1
      IF(T.EQ.0) GO TO 130
C--1ST DERIVATIVE BOUNDARIES GIVEN
      NE=N-1
      IF(H) 10,80,10
C--EQUAL SPACING H .GT. 0. AND T=1
   10 HH=3.0/H
      DO 20 I=1,NE
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
   20 P(I)=HH*(Y(I+2)-Y(I))
      P(1)=P(1)-D(1)
      P(NE)=P(NE)-D(2)
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE
   30 FA=1./B(1)
      C(1)=C(1)*FA
      P(1)=P(1)*FA
      DO 40 I=2,NE
      MUL=1.0/(B(I)-A(I)*C(I-1))
      C(I)=MUL*C(I)
```

```
   40 P(I)=MUL*(P(I)-A(I)*P(I-1))
C--OBTAIN SPLINE COEFFICIENTS
      A(NE+T)=P(NE)
      I=NE-1
   50 A(I+T)=P(I)-C(I)*A(I+T+1)
      I=I-1
       IF(I.GE.1) GO TO 50
      IF(T.EQ.0) GO TO 60
      A(1)=D(1)
      A(M)=D(2)
   60 IF(H.EQ.0.) GO TO 110
      HH=1.0/H
      DO 70 I=1,N
      MUL=HH*(Y(I+1)-Y(I))
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))
   70 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))
      RETURN
C--UNEQUAL SPACING H=0.. AND T=1
   80 DO 90 I=1,N
   90 S(I+1)=X(I+1)-X(I)
      DO 100 I=1,NE
      B(I)=2.0*(S(I+1)+S(I+2))
      C(I)=S(I+1)
      A(I)=S(I+2)
  100 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*
     1 (Y(I+1)-Y(I)))/(S(I+1)*S(I+2))
      P(1)=P(1)-S(3)*D(1)
      P(NE)=P(NE)-S(N)*D(2)
      GO TO 30
  110 DO 120 I=1,N
      HH=1.0/S(I+1)
      MUL=(Y(I+1)-Y(I))*HH**2
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH
  120 C(I)=(-2.0*MUL+(A(I+1)+A(I))*HH)*HH
      RETURN
C--2ND DERIVATIVE BOUNDARIES GIVEN
  130 NE=N+1
      IF(H) 140,160,140
C--EQUAL SPACING H .GT. 0 AND T=0
  140 HH=3.0/H
      DO 150 I=2,N
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
  150 P(I)=HH*(Y(I+1)-Y(I-1))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
      C(NE)=1.0
      A(NE)=1.0
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)
      GO TO 30
C--UNEQUAL SPACING H=0 AND T=0
```

```
      .
  160 DO 170 I=1,N
  170 S(I+1)=X(I+1)-X(I)
      N1=N-1
      DO 180 I=1,N1
      B(I+1)=2.0*(S(I+1)+S(I+2))
      C(I+1)=S(I+1)
      A(I+1)=S(I+2)
  180 P(I+1)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*
     1 (Y(I+1)-Y(I)))/(S(I+1)*S(I+2))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
      C(NE)=1.0
      A(NE)=1.0
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)
      GO TO 30
  190 M=-IABS(M)
      RETURN
      END
```

```
C
C  ROUTINE TO SAVE SPLINE COEFFICIENTS
C
      SUBROUTINE PUTCOF(X,Y,C,NC)
      DIMENSION C(16,NC),X(NC),Y(1)
      DOUBLE PRECISION TITLE,PGMID
      COMMON TITLE(7),PGMID,NCOL,NROW,NZ,XO,DELX,
     1 YO,DELY,IYP,IUP,IPP,IQP,ISP,IWP,IRP,IO,IOUT
     2 ,DVAL,NFX,NFY,NROWO,NCOL1,NROW1,IDER
      DOUBLE PRECISION PGMID2
      DATA NZA,PGMID2/16,'SPLCOEFF**'/
C
      WRITE(IO) TITLE,PGMID2,NCOL,NROW,NZA,XO,DELX,YO,DELY
      IF (DELX.EQ.0.) WRITE(IO) X
      DO 40 I=1,NROW
      IF (I.EQ.NROW) GO TO 10
      CALL SPLN2B(C,I)
      C(1,NCOL)=DVAL
      GO TO 30
   10 DO 20 J=1,NCOL
   20 C(1,J)=DVAL
   30 WRITE(IO) Y(I),C
   40 CONTINUE
      RETURN
      END
```

```
C
C  ROUTINE TO RETRIEVE SPLINE COEFFICIENTS
C
      SUBROUTINE GETSPL(C,J,Y,WORK,NC)
      DIMENSION C(16,NC),WORK(16,NC),Y(1)
      DOUBLE PRECISION TITLE,PGMID
      COMMON TITLE(7),PGMID,NCOL,NROW,NZ,XO,DELX,
     1 YO,DELY,IYP,IUP,IPP,IQP,ISP,IWP,IRP,IO,IOUT
     2 ,DVAL,NFX,NFY,NROWO,NCOL1,NROW1,IDER
C
      IF (J.GT.1) GO TO 20
      READ(IO,END=50) Y(1),C
   10 READ(IO,END=50) Y(2),WORK
      J=1
      RETURN
   20 DO 40 I=1,NCOL
      DO 30 J=1,16
   30 C(J,I)=WORK(J,I)
   40 CONTINUE
      Y(1)=Y(2)
      GO TO 10
C
   50 WRITE(ITTY,60)
   60 FORMAT(' %PREMATURE EOF ON COEFFICIENT FILE')
      STOP
      END
```

APPENDIX D.

LISTING OF PROGRAM SPLN2X

```
C
C   SPLINE COEFFICIENT PROGRAM
C
C   CODED BY
C     GERALD IAN EVENDEN
C     U. S. GEOLOGICAL SURVEY
C     DENVER FEDERAL CENTER
C     DENVER, COLORADO 80225
C
C   COEFFICIENTS FOR GRID UP TO 500 ROWS AND COLUMNS
C   CAN BE DETERMINED BY THIS PROGRAM.  IF LARGER GRIDS
C   CHANGE THE DIMENSIONS IN SUBROUTINE SPLN2X.
C
C   FOR THE DEC-SYSTEM 10 MEMORY REQUIRED IS APPROXIMATELY..
C       10 + 26*K/1000 K-WORDS
C   WHERE K IS THE DIMENSION OF THE SPLN2X ARRAYS.
C
C   NOTE... THIS PROGRAM CONTAINS SEVERAL NON-ANSI FORTRAN
C   STATEMENTS.  ALTHOUGH SEVERAL OF THE CONSTRUCTS ARE
C   AVAILABLE ON OTHER COMPUTER SYSTEMS, CARE MUST BE
C   EXERCISED IN THE TRANSPORTATION OF THIS ROUTINE.
C
      DOUBLE PRECISION NAME,FILES,PGM
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 X0,DX,Y0,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
C
      DOUBLE PRECISION BLANK
      DATA BLANK/'          '/
C
   10 TYPE 20
   20 FORMAT(' ENTER INPUT FILE NAME.EXT:',$)
      ACCEPT 30,FILES(1)
   30 FORMAT(A10)
      NAME=FILES(1)
      IF (NAME.EQ.BLANK) GO TO 10
      IF (NAME.EQ.'EXIT') STOP
      OPEN(UNIT=10,FILE=NAME,DEVICE='DSK',
     1 ACCESS='SEQIN',MODE='BINARY')
      READ(10,END=80,ERR=80) HEAD
      IF (NCOL.LE.500.AND.NROW.LE.500) GO TO 60
      TYPE 40
   40 FORMAT(' %NCOL OR NROW GREATER THAN 500')
   50 CLOSE(UNIT=10)
      GO TO 10
C
   60 IF (NZ.EQ.1) GO TO 100
      TYPE 70
```

```
   70 FORMAT(' %NZ NOT EQUAL TO 1')
      GO TO 50
C
   80 TYPE 90
   90 FORMAT(' %INPUT EOF OR ERROR ON HEADER')
      GO TO 50
C
C  INPUT LOOKS OK, PROCEED
  100 TYPE 110
  110 FORMAT(' ENTER COEFFICIENT FILE NAME.EXT:',$)
      ACCEPT 30,FILES(5)
      NAME=FILES(5)
      IF (NAME.EQ.BLANK) GO TO 100
      CALL SPLN2X
      GO TO 10
      END
```

```
      BLOCK DATA BLKSPX
      DOUBLE PRECISION NAME,FILES,PGM
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 X0,DX,Y0,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
      DATA FILES/0D0,'LPS991.TMP','LPS992.TMP',
     1 'LPS993.TMP',0D0/
      DATA DVAL/"377777777777/
      END
```

```
C
C   DISK STORAGE OVERLAY SPLINE COEFFICIENT ROUTINE.
C
C   BASED ON EARLIER ANDERSON ROUTINE IN PROGRAM BIGGRID.
C   THIS ROUTINE, HOWEVER, REQUIRES 3 (VS. 8) SCRATCH
C   FILES AND 8*NROW+4*NCOL LESS READS AND WRITES.
C
C   NOTE...IF DIMENSIONS ARE CHANGED, ENSURE ARRAYS VA AND CU
C   HAVE ONE MORE WORD THAN BASIC SIZE.  ONLY THE VALUES 500
C   AND 501 SHOULD BE CHANGED.
C
      SUBROUTINE SPLN2X
      DOUBLE PRECISION NAME,FILES,PGM
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 XO,DX,YO,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
C
      DIMENSION V(500),VA(501),A(500),B(500),C(500),
     1 W1(500),W2(500),DJ1(500),DJY(500),D(2),
     2 X(500),Y(500),CO(16,500)
C
      EQUIVALENCE (V(1),VA(2))
C
C
      DIMENSION CU(501,2),CP(500,2),CQ(500,2),CS(500,2)
     1 ,WC(500)
      EQUIVALENCE (VA(1),CU(1,1)),(A(1),CU(1,2)),
     1 (B(1),CP(1,1)),(C(1),CP(1,2)),
     2 (DJ1(1),CQ(1,1)),(DJY(1),CQ(1,2)),
     3 (W1(1),CS(1,1)),(W2(1),CS(1,2)),
     4 (CO(1,1),WC(1))
      NCOL1=NCOL+1
      D(1)=0.
      D(2)=0.
C
C   IF EQUI-X GENERATE X ARRAY OR INPUT.
      IF (DX.NE.0.) GO TO 10
      CALL GETV(10,2,X,NCOL)
      GO TO 30
C
   10 F=0.
      DO 20 I=1,NCOL
      X(I)=XO+F*DX
   20 F=F+1.
C
C   GENERATE P ARRAY
   30 DO 40 I=1,NROW
      CALL GETV(10,2,VA,NCOL1)
C
C   SAVE Y VALUES
      Y(I)=VA(1)
      CALL DVALS(V,NCOL,DVAL)
```

```
      .
      CALL SPLIN1(NCOL,DX,X,V,A,B,C,0,D,W1,W2)
      CALL PUTV(11,I,A,NCOL)
   40 CONTINUE
C
      CLOSE(UNIT=10)
      CLOSE(UNIT=11)
C
C  Y ARRAY NEEDS TO BE GENERATED?
      IF (DY.EQ.0.) GO TO 60
C
C  YES.
      F=0.
      DO 50 I=1,NROW
      Y(I)=YO+F*DY
   50 F=F+1.
C
C  GENERATE Q ARRAY
   60 DO 80 I=1,NCOL
      DO 70 J=1,NROW
      CALL GETV(10,J,VA,NCOL1)
   70 A(J)=V(I)
      CLOSE(UNIT=10)
      CALL DVALS(A,NROW,DVAL)
      CALL SPLIN1(NROW,DY,Y,A,V,B,C,0,D,W1,W2)
      CALL PUTV(12,I,V,NROW)
      DJY(I)=V(NROW)
      DJ1(I)=V(1)
   80 CONTINUE
C
C
C  GENERATE CROSS DERIVATIVES
      CALL SPLIN1(NCOL,DX,X,DJ1,A,B,C,0,D,W1,W2)
      DO 90 I=1,NCOL
   90 DJ1(I)=A(I)
      CALL SPLIN1(NCOL,DX,X,DJY,A,B,C,0,D,W1,W2)
      DO 100 I=1,NCOL
  100 DJY(I)=A(I)
C
C  GENERATE REMAINDER OF S ARRAY
      DO 120 I=1,NCOL
      DO 110 J=1,NROW
      CALL GETV(11,J,W1,NCOL)
  110 V(J)=W1(I)
      CLOSE(UNIT=11)
      D(1)=DJ1(I)
      D(2)=DJY(I)
      CALL SPLIN1(NROW,DY,Y,V,A,B,C,1,D,W1,W2)
      CALL PUTV(13,I,A,NROW)
  120 CONTINUE
C
C
      NZ=16
      PGM='SPLCOEFF**'
      NAME=FILES(5)
```

```
      .
      OPEN (UNIT=14,FILE=NAME,MODE='BINARY',
     1 ACCESS='SEQOUT',DEVICE='DSK')
      WRITE(14) HEAD
      IF (DX.EQ.0.) CALL PUTV(14,2,X,NCOL)
      CALL SPL2AX(X,Y,CO,NCOL)
C
C  INITIAL LOAD
      CALL GETV(10,1,CU(1,1),NCOL1)
      CALL GETV(11,1,CP(1,1),NCOL)
      CLOSE(UNIT=12)
      CLOSE(UNIT=13)
      DO 130 J=1,NCOL
      CALL GETV(12,J,WC,NROW)
      CQ(J,1)=WC(1)
      CALL GETV(13,J,WC,NROW)
      CS(J,1)=WC(1)
  130 CONTINUE
C
C  GENERATE SPLINE COEFFICIENTS
      DO 150 I=2,NROW
      K2=MOD(I-1,2)+1
      K1=MOD(I,2)+1
      CALL GETV(10,I,CU(1,K2),NCOL1)
      CALL GETV(11,I,CP(1,K2),NCOL)
      CLOSE(UNIT=12)
      CLOSE(UNIT=13)
      DO 140 J=1,NCOL
      CALL GETV(12,J,WC,NROW)
      CQ(J,K2)=WC(I)
      CALL GETV(13,J,WC,NROW)
      CS(J,K2)=WC(I)
  140 CONTINUE
      CALL SPL2BX(CU(2,K1),CP(1,K1),CQ(1,K1),CS(1,K1),
     1 CU(2,K2),CP(1,K2),CQ(1,K2),CS(1,K2),I-1)
  150 CONTINUE
C
C  LAST ROW OF DVALS REQUIRED
      DO 160 I=1,NCOL
  160 CO(1,I)=DVAL
      CALL PUTX(Y(NROW),CO,16*NCOL)
      CLOSE(UNIT=10)
      CLOSE(UNIT=11,DISPOSE='DELETE')
      CLOSE(UNIT=12,DISPOSE='DELETE')
      CLOSE(UNIT=13,DISPOSE='DELETE')
      CLOSE(UNIT=14)
      RETURN
      END
```

```
      SUBROUTINE SPL2AX(X,Y,C,NX)
      DOUBLE PRECISION NAME,FILES,PGM
      DIMENSION U(1),P(1),Q(1),S(1),UI(1),PI(1),
     1 QI(1),SI(1)
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 X0,DX,Y0,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
      DIMENSION X(1),Y(1),C(16,NX)
C
      DIMENSION AX(16),AY(16),AA(16)
      DATA AX,AY/1.,4*0.,1.,10*0.,1.,4*0.,1.,10*0./
C
C  ESTABLISH ONE TIME VALUES AND ADDRESSES.
C
      NX1=NX-1
      NX16=NX*16
      IF(DX.NE.0.0) CALL SETAX(AX,DX)
      IF(DY.NE.0.0) CALL SETAY(AY,DY)
      RETURN
C
C
C  GENERATE ROW OF SPLINE POLYNOMIAL
C  COEFFICIENTS.
C
      ENTRY SPL2BX(U,P,Q,S,UI,PI,QI,SI,IY)
      IY1=IY+1
      IF(DY.EQ.0.0) CALL SETAY(AY,Y(IY1)-Y(IY))
      DO 20 I=1,NX1
      I1=I+1
      IF (U(I).EQ.DVAL.OR.U(I1).EQ.DVAL.OR.
     1 UI(I).EQ.DVAL.OR.UI(I1).EQ.DVAL) GO TO 10
      IF(DX.EQ.0.0) CALL SETAX(AX,X(I1)-X(I))
      AA(1)=U(I)
      AA(2)=P(I)
      AA(3)=U(I1)
      AA(4)=P(I1)
      AA(5)=Q(I)
      AA(6)=S(I)
      AA(7)=Q(I1)
      AA(8)=S(I1)
      AA(9)=UI(I)
      AA(10)=PI(I)
      AA(11)=UI(I1)
      AA(12)=PI(I1)
      AA(13)=QI(I)
      AA(14)=SI(I)
      AA(15)=QI(I1)
      AA(16)=SI(I1)
      CALL GMPRD(AX,AA,C(1,NX),4,4,4)
      CALL GMPRD(C(1,NX),AY,C(1,I),4,4,4)
      GO TO 20
   10 C(1,I)=DVAL
```

```
   .
20 CONTINUE
   C(1,NX)=DVAL
   CALL PUTX(Y(IY),C,NX16)
   RETURN
   END
```

```
      SUBROUTINE GETV(IN,IREC,V,IDIM)
      DIMENSION V(IDIM)
      DOUBLE PRECISION NAME,FILES,PGM
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 X0,DX,Y0,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
C
C  GET ROUTINE FOR INPUTTING SPLINE DATA.
C
C     IN      FILE NAME      DATA       RECORD SIZE
C
C     11     LPS991.TMP     P-ROWS       NCOL
C     12     LPS992.TMP     Q-COLS       NROW
C     13     LPS993.TMP     S-COLS       NROW
C
      IF (IREC.NE.1) GO TO 10
C
C  FILE NEEDS TO BE REOPENED.
      NAME=FILES(IN-9)
      OPEN(UNIT=IN,FILE=NAME,MODE='BINARY',
     1 ACCESS='SEQIN',DEVICE='DSK')
      IF (IN.NE.10) GO TO 10
      READ(IN,END=20,ERR=20)
      IF (DX.EQ.0.) READ (IN,ERR=20,END=20)
C  GET DATA
   10 READ(IN,END=20,ERR=20) V
      RETURN
C
C  ERROR CONDITION, LIGHT FUSE AND BOMB OUT.
   20 TYPE 30
   30 FORMAT(' %I/O ERROR ON INPUT FILES')
      STOP
      END
```

```
      SUBROUTINE PUTV(IOUT,IREC,V,IDIM)
      DIMENSION V(IDIM)
      DOUBLE PRECISION NAME,FILES,PGM
      COMMON TITLE(14),PGM,NCOL,NROW,NZ,
     1 XO,DX,YO,DY,
     2 FILES(5),NAMES(3),DVAL
      DIMENSION HEAD(23)
      EQUIVALENCE (HEAD(1),TITLE(1)),(NAME,NAMES(1))
C
C   OUTPUT ROUTINE FOR COEFFICIENT DATA.
      IF (IREC.NE.1) GO TO 10
C
C   FILE NEEDS TO BE REOPENED.
      NAME=FILES(IOUT-9)
      OPEN(UNIT=IOUT,FILE=NAME,MODE='BINARY',
     1 ACCESS='SEQOUT',DEVICE='DSK')
C
C   PUT DATA.
   10 WRITE(IOUT) V
      RETURN
C
C   OUTPUT ROUTINE FOR POLYNOMIAL
C   ROW.
C
      ENTRY PUTX(Y,V,IDIM)
C
      WRITE(14) Y,V
      RETURN
      END
```

APPENDIX E.

BASIC GRAPHICS SYSTEM.

THE FOLLOWING USER DOCUMENTATION IS INCLUDED TO PROVIDE EXPLANATION OF THE GRAPHICS CALLS IN PROGRAM CONTUR. THE GRAPHICS SYSTEM IS MOSTLY IN DEC SYSTEM 10 ASSEMBLY LANGUAGE AND, AS A CONSEQUENCE, IS OF LITTLE GENERAL INTEREST.

INTRODUCTION.

PLOTTER SOFTWARE SYSTEMS ARE PROVIDED TO FACILITATE PROGRAMMER USAGE OF DIGITAL PLOTTERS IN MUCH THE SAME MANNER AS COMPILER LANGUAGE INPUT-OUTPUT STATEMENTS FACILITATE USAGE OF I/O DEVICES. SUCH SYSTEMS SHOULD ATTEMPT TO ACHIEVE TWO, OFTEN MUTUALLY EXCLUSIVE, GOALS: 1) SIMPLICITY AND 2) FLEXIBILITY. AN ADDITIONAL FEATURE SHOULD INCLUDE THE ABILITY OF THE BASIC SYSTEM TO CREATE MACHINE INSTRUCTIONS FOR MORE THAN ONE PLOTTER AND THUS ALLOW EASY SWITCHING OF PLOTTING DEVICES WITH MINIMAL PROGRAM RECODING. THIS SYSTEM PROVIDES FOR THREE PLOTTERS SELECTED BY ONE ENTRY CALL AND PROVIDES FOR ALL BASIC PLOTTING OPERATIONS WITH REASONABLE EASE OF USAGE.

THIS PLOTTING SYSTEM HAS FOUR BASIC ENTRIES:

SCALE : FOR PLOT INITIALIZATION AND SCALING,
LINE : FOR LINE DRAWING,
CHAR : FOR CHARACTER DRAWING,
ENDPLT : FOR PLOT TERMINATION AND DIAGNOSTICS

AND SEVERAL SUPPLEMENTARY SERVICE ENTRIES. THE FOLLOWING DESCRIPTION COMPLETELY DEFINES ALL ENTRY PARAMETERS. A FINAL SECTION IS INCLUDED WHICH CONTAINS A PROGRAM EXAMPLE ALONG WITH RESULTANT PLOT. THE READER SHOULD CAREFULLY READ AND UNDERSTAND THE SCALE CALL DESCRIPTION FIRST AS MOST FUNDAMENTAL DEFINITIONS AND CONVENTIONS ARE DESCRIBED IN THIS SECTION.

PLOT SCALING.

THE PLOTTING SYSTEM SCALING CALL IS REQUIRED TO INITIALIZE THE SOFTWARE TO THE USER AND PLOTTING MACHINE REQUIREMENTS. IN ADDITION TO OPENING PLOT OUTPUT FILE AND GENERAL SYSTEM INITIALIZATION A RESULT OF THIS CALL ESTABLISHES TWO REGIONS: 1) A "PLOT AREA" AND 2) A "DATA AREA"
THE PLOT AREA DEFINES THE MAXIMUM LIMITS OF ANY PLOT OPERATIONS. THE PLOT AREA IS, OF COURSE, LIMITED TO THE PHYSICAL DIMENSIONS OF THE PLOTTING AREA OF THE MACHINE EMPLOYED. IN PLOTTING PARAMETERS REFERRING TO THE PLOT AREA "PLOT UNITS " ARE EMPLOYED. PLOT UNITS HAVE A DEFAULT VALUE OF INCHES BUT CAN BE REDEFINED AS CENTIMETRES WITH A CALL TO PLTSET. THE ORIGIN OF THE PLOT AREA COORDINATE SYSTEM IS ALWAYS LOCATED AT THE LOWER-LEFT HAND CORNER OF THE PLOT AREA AND IS POSITIVE ALONG THE X OR RIGHT HAND AXIS AND Y OR VERTICAL AXIS. FIG. 1 SHOWS BASIC LAYOUT AND GENERAL MEANING OF SCALE PARAMETERS.

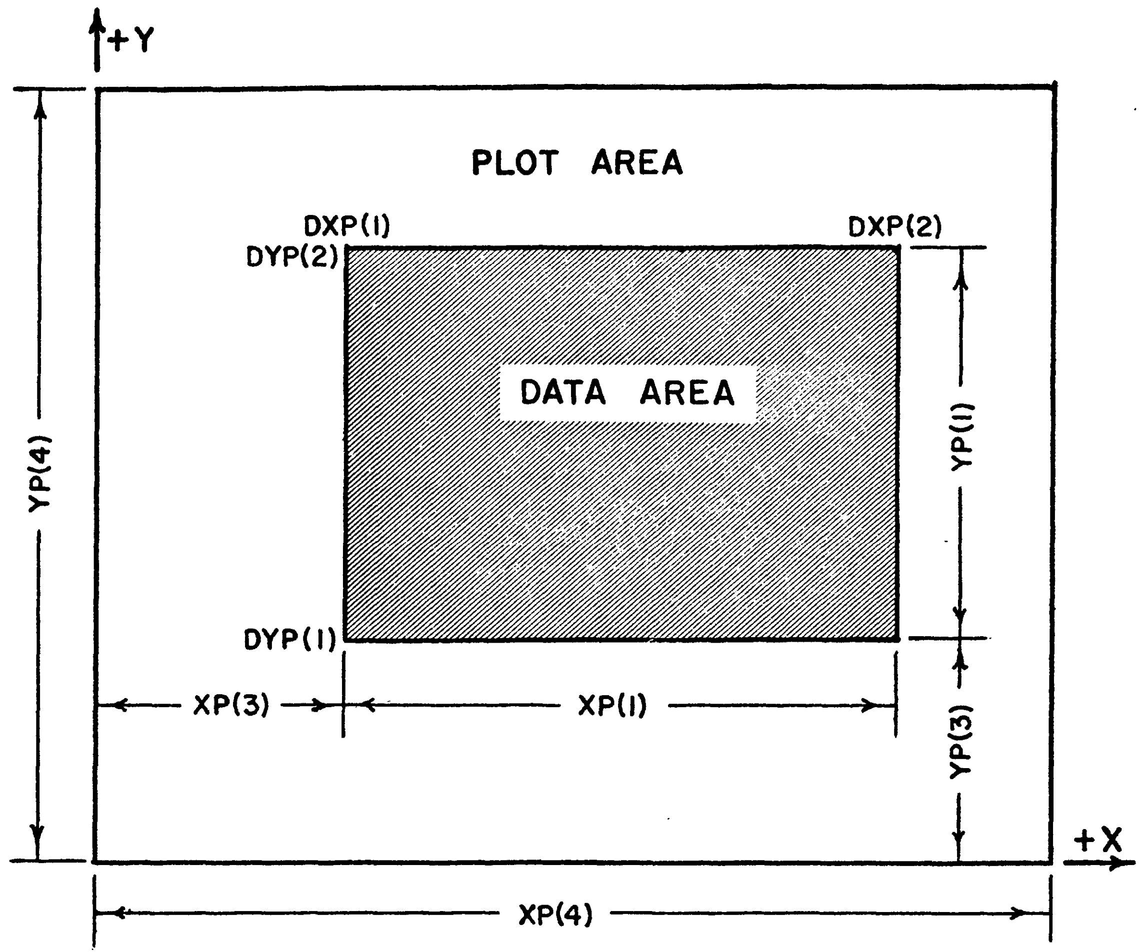


FIGURE 1. SCALE GEOMETRIC PARAMETERS.

THE DATA AREA IS ALWAYS LESS THAN OR EQUAL TO THE PLOT AREA. ONLY DATA POINTS WHOSE X - Y VALUES ARE IN THIS AREA WILL BE PLOTTED. DATA UNITS ARE COMPLETELY ARBITRARY AND MAY BE OF ANY TYPE: REAL, INTEGER, REAL OR IMAGINARY PART OF COMPLEX OR DOUBLE PRECISION.

SCALING IS PERFORMED BY CALLING THE ENTRY SCALE:

CALL SCALE(DXP,DYP,XP,YP,NOPTS,ICODE).

WHEN A SUCCESSFUL SCALE CALL HAS BEEN MADE THE PLOTTING SYSTEM ASSUMES A "SCALED STATE". SCALE MAY BE CALLED AGAIN IN A SCALED STATE FOR REDEFINING FACTORS OF THE PLOT AREA. TERMINATION OF THE SCALED STATE AND RESULTANT COMPLETION OF THE PLOT IS MADE BY CALLING ENDPLT AND A SUBSEQUENT SCALE CALL WILL INITIALIZE A NEW PLOT. PLOTTING CALLS (IE. LINE AND CHAR) ARE ONLY ALLOWED WHEN THE SYSTEM IS IN THE SCALED STATE.

SCALE PARAMETER DESCRIPTION.

DXP AND DYP ARE TWO ELEMENT ARRAYS WHICH DEFINE THE DATA RANGE. THEY MAY BE OF ANY STANDARD FORTRAN IV TYPE AS DEFINED IN PARAMETER XP AND YP. DXP(1) AND DYP(1) DEFINE THE RESPECTIVE LEFT AND BOTTOM LIMITS OF THE DATA AREA IN DATA UNITS. DXP(2) AND DYP(2) DEFINE THE RESPECTIVE RIGHT AND TOP LIMITS OF THE DATA AREA IN DATA UNITS. DXP(1) AND DXP(2), FOR EXAMPLE, ARE MOST OFTEN THE RESPECTIVE MINIMUM AND MAXIMUM X DATA VALUES TO BE PLOTTED. HOWEVER, IF A DATA AXIS IS TO BE REVERSED (INCREASING TO THE LEFT) THE DXP(1) WOULD BE THE MAXIMUM X AND DXP(2) THE MINIMUM X. OF COURSE, THE SAME APPLIES TO DYP(1) AND DYP(2). THE ONLY AND OBVIOUS RESTRICTION IS THAT DXP(1).NE.DXP(2) AND DYP(1).NE.DYP(2).

XP AND YP ARE TYPE REAL ARRAYS WITH NOPTS ELEMENTS.

XP(1) AND YP(1) DEFINE THE RESPECTIVE X AND Y SIZE OF THE DATA AREA IN PLOT UNITS. BOTH MUST ALWAYS BE SPECIFIED AND GREATER THAN ZERO.

XP(2) AND YP(2) IS A SWITCH TO DEFINE THE RESPECTIVE X AND Y DATA TYPE AND WHETHER A LINEAR OR LOGARITHMIC AXIS IS TO BE PLOTTED.

XP(2),YP(2) VALUE	RESPECTIVE AXIS DATA TYPE
<=1 OR =8 OR =9	REAL
=2 OR 10	INTEGER
=3 OR =11	REAL PART OF COMPLEX
=4 OR =12	IMAGINARY PART OF COMPLEX
=5 OR =13	DOUBLE PRECISION

WHEN XP(2) OR YP(2) ARE <8 THEN THE RESPECTIVE AXIS ARE LINEAR. WHEN >= 8 THEN AXIS LOGARITHMIC. NOTE THAT XP(2) AND YP(2) DEFINE THE TYPE FOR ALL DATA VALUE REFERENCES IN LINE AND CHAR CALLS. IF NOPTS<2 THEN X AND Y DATA VALUES ASSUMED TO BE REAL AND LINEAR.

XP(3) AND YP(3) DEFINE THE POSITION OF THE LOWER-LEFT CORNER OF THE DATA AREA ON THE PLOT AREA IN PLOT UNITS. IF NOPTS<=2 THEN THE LOWER-LEFT EDGE OF THE DATA AREA IS COINCIDENT WITH THE LOWER-LEFT EDGE OF THE PLOT AREA (XP(3)=YP(3)=0).

XP(4) AND YP(4) DEFINE THE SIZE OF THE PLOT AREA IN PLOT UNITS. IF NOPTS<=3 AND THIS CALL MADE WHILE NOT IN THE SCALED STATE THEN THE PLOT AREA WILL BE COMPUTED BY THE FOLLOWING:

XP(4)=XP(1)+XP(3)

YP(4)=YP(1)+YP(3)

IF XP(4) AND YP(4) ARE SPECIFIED THEN THEY MUST BE GREATER THAN OR EQUAL TO THE VALUES OF THE PREVIOUS EXPRESSION. XP(4) AND YP(4) SHOULD NOT BE SPECIFIED FOR A SCALE CALL WHEN THE SYSTEM IS IN THE SCALED STATE.

NOPTS IS AN INTEGER VALUE SPECIFYING THE NUMBER OF PARAMETERS IN THE XP-YP ARRAYS. IN ALL CASES: 1<=NOPTS<=4 AND WHEN IN THE SCALED STATE: NOPTS<=3.

ICODE IS AN INTEGER VARIABLE WHICH IS SET TO 0 UPON RETURN FROM SCALE IF NO FAULT COULD BE FOUND WITH THE INPUT PARAMETERS. IF ERRORS WERE DETECTED A NON-ZERO VALUE IS RETURN AND ALL SUBSEQUENT PLOTTING CALLS ARE DISALLOWED EXCEPT FOR ANOTHER SCALE CALL. IF APPROPRIATE USER ACTION IS NOT TAKEN AND ANOTHER SCALE CALL IS MADE WITH THE ASSUMPTION OF REDEFINING THE DATA AREA THEN UNPREDICTABLE RESULTS CAN BE EXPECTED.

LINE PLOTTING.

PLOTTING OF LINES IS PERFORMED BY CALLING THE ENTRY LINE:

```
CALL LINE(X,Y,N,ICON,IPN)
```

THE LINE ENTRY SEQUENTIALLY CONNECTS 2 OF MORE X - Y DATA POINTS WITH STRAIGHT LINE VECTORS. LINE PROVIDES FOR CONTINUATION OF LINES GENERATED BY PREVIOUS CALLS WITHOUT CONCERN FOR STORING THE LAST X - Y POINT OF THE PREVIOUS CALL. ANY PART OF THE LINE INSIDE THE DATA AREA CONNECTING TWO X - Y POINTS IS DRAFTED REGARDLESS OF ONE OR BOTH OF THE X - Y POINTS BEING OUTSIDE THE DATA AREA. THE DRAFTED LINES MAY ALSO BE IN DASHED FORM.

IF SMOOTH, CURVED LINES ARE REQUIRED THE USER IS RESPONSIBLE FOR ENSURING THAT THE SPACING OF THE POINTS IS ADEQUATE FOR THE RESULTING STRAIGHT LINE APPROXIMATION.

LINE PARAMETER DESCRIPTION.

X,Y COORDINATES IN DATA UNITS WHICH DEFINE THE SEQUENCE OF LINE SEGMENTS TO BE DRAWN. IF N=1 THEN X-Y MAY BE SIMPLE VARIABLES, ELSE ARRAYS OF N ELEMENTS.

N IS AN INTEGER VARIABLE DEFINING THE NUMBER OF X-Y POINTS.

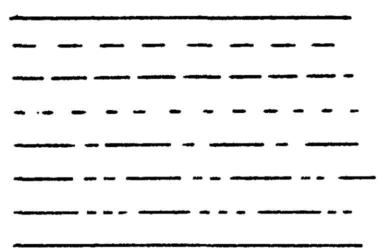
ICON IS AN INTEGER VARIABLE WHICH SPECIFIES WHETHER THE line is to be continued from the last X-Y point of a previous call to the first point of the current call. Continuation only applies to previous call with identical IPN parameter value. If a previous call was never made the line starts at X(1) - Y(1).

ICON=0 do not continue, "new line".
ICON.NE.0 continue line from previous point.

IPN is an integer variable which selects up to 8 pseudo pens which operate independently of each other. Normally, each pseudo pen is set to a unique mode of drafting (solid, dashed) for visual identification on the plot. The default mode for each IPN is:

IPN value   plotting character

0
1
2
3
4
5
6
7



Character plotting.

Plotting or posting of characters and symbols is performed by calling the entry CHAR:

CALL CHAR(X,Y,A,N,ICODE,SIZE,THETA,XOFF,YOFF).

The standard character set contains 101 gothic characters, including upper and lower case alpha, numeric and punctuation and, optionally, the characters may be italicized. In addition, 13 special symbols usually used for point plotting are included. Note that special symbols cannot be italicized.

A call to CHAR may plot either a character or symbol at a number of X - Y locations or plot a string of characters or symbols starting at one X - Y location. The X - Y location, in the later case, may be either in plot units or data units.

CHAR parameter description.

X and Y cordinates in either data or plot units which define the position of the center of the first character to be plotted. Note that when ICODE=0 or 1 X and Y are arrays of N elements. When ICODE<=2

X AND Y ARE IN DATA UNITS AND WHEN ICODE=3 X AND Y ARE IN PLOT UNITS.

A IS A CHARACTER OR CHARACTER STRING TO BE PLOTTED. WHEN ICODE<=1 A CONTAINS ONE CHARACTER. IF THE FIRST CHARACTER OR BYTE OF A IS A BINARY ZERO THEN THE CHARACTER IS SELECTED FROM A RIGHT JUSTIFIED INTEGER WORD. OTHERWISE THE CHARACTER IS ASSUMED TO BE LEFT JUSTIFIED IN A. WHEN ICODE=1, A IS A SIMPLE VARIABLE AND THE CONTAINED CHARACTER IS REPEATED FOR N X - Y POINTS. WHEN ICODE=0, A IS AN INTEGER ARRAY OF DIMENSION N WITH A UNIQUE CHARACTER PLOTTED WITH EACH RESPECTIVE X(N) - Y(N) POINT. FOR ICODE=2 OR 3, A IS A CHARACTER STRING AND N IS THE NUMBER OF CHARACTERS IN THE STRING. X AND Y ARE SIMPLE VARIABLES IN THIS CASE.

N IS AN INTEGER VARIABLE DEFINING THE NUMBER OF POINTS OR CHARACTERS TO BE PLOTTED.

ICODE IS AN INTEGER VARIABLE SELECTING THE DESIRED PLOTTING DEFINITION OF X, Y, A AND ITALICIZING. BASICALLY, THE TWO LOWEST BITS IF ICODE DEFINE THE USE AND MEANING OF X, Y, A, AND N AS PREVIOUSLY DISCUSSED. IF 4 IS ADDED TO ICODE THE CHARACTERS ARE ITALICIZED. THE FOLLOWING IS A SUMMATION OF THE EFFECT OF ICODE ON EACH OF THE FIRST FOUR CHAR ARGUMENTS:

X AND Y	A	N	ICODE
DATA UNITS ARRAY	ARRAY	DIMENSION OF X, Y AND A	0 AND 4
DATA UNITS ARRAY	SIMPLE VARIABLE	DIMENSION OF X AND Y	1 AND 5
SIMPLE VARIABLE IN DATA UNITS	CHARACTER STRING	NUMBER OF CHARACTERS	2 AND 6
SIMPLE VARIABLE IN PLOT UNITS	CHARACTER STRING	NUMBER OF CHARACTERS	3 AND 7

SIZE IS A REAL VARIABLE DEFINING THE SIZE OF THE CHARACTERS TO BE PLOTTED IN PLOT UNITS. NOTE THAT SIZE MUST BE GREATER THAN ZERO.

THETA IS A REAL VARIABLE DEFINING THE ROTATION, IN RADIANS, OF THE CHARACTERS FROM THE POSITIVE X AXIS ABOUT THE X - Y POINT.

XOFF AND YOFF ARE REAL VARIABLES WHICH DEFINE A SHIFT IN PLOT UNITS OF THE CENTER OF THE FIRST CHARACTER

FROM THE SPECIFIED X - Y COORDINATES. NOTE THAT THE SHIFT IS MADE PRIOR TO THETA ROTATION.

PLOT TERMINATION.

A CALL TO ENDPLT IS REQUIRED WHEN A PLOT IS COMPLETE. THIS CALL PERFORMS THE FOLLOWING FUNCTION: 1) SPILLS ALL OUTPUT WORK BUFFER, 2) GENERATES CLOSURE CHECK TICKS FOR INCREMENTAL PLOTTERS, 3) PRINTS PLOT DIAGNOSTICS AND 4) RETURNS PLOTTING SOFTWARE TO AN UNSCALED STATE. ANY SUBSEQUENT CALL TO THE PLOT SYSTEM MUST BE ANOTHER SCALE CALL OR A CALL TO PLTFIL OR PLTSET. TO EXECUTE ENDPLT:

CALL ENDPLT(0).

ENDPLT PARAMETERS.

THE ARGUMENT IS NOT CURRENTLY EMPLOYED BUT IS INCLUDED FOR POSSIBLE FUTURE EXPANSION.

PLOTTER SELECTION.

THE GRAPHICS SYSTEM ENTRY PLTSET ALLOWS THE USER TO SELECT THE PLOTTER DESIRED. IF A CALL IS NOT MADE TO THIS ROUTINE THE GERBER 622 IS SELECTED AS A DEFAULT. THE PLOTTER SELECTED REMAINS IN EFFECT UNTIL ANOTHER PLTSET CALL IS EXECUTED. IN ADDITION, THE ROUTINE RETURNS THE PHYSICAL SIZE OF THE PLOT AREA AND ALLOWS THE USER TO SELECT THE TYPE OF PLOT UNITS DESIRED (INCHES OR CENTIMETRES). TO EXECUTE PLTSET:

CALL PLTSET(ICODE,XBOARD,YBOARD,INCM).

THE CALL TO PLTSET MUST BE MADE WHILE THE PLOTTING SYSTEM IS NOT IN THE SCALED STATE.

PLTSET PARAMETER DESCRIPTION.

ICODE SELECTS THE PLOTTING DEVICE.
ICODE=0 GERBER 622
ICODE=1 TEKTRONIX 4010
ICODE=2 HEWLETT-PACKARD 7202A.
ICODE=3 HEWLETT-PACKARD 7203A.
ANY OTHER VALUE WILL CAUSE AN ERROR CONDITION.

XBOARD AND YBOARD ARE THE RETURNED X AND Y AXIS SIZE IN PLOT UNITS OF THE DEVICE SELECTED. PRINCIPALLY EMPLOYED IN SCALING THE DIMENSIONS OF THE USERS PLOT TO FIT THE DEVICE SELECTED.

INCM selects the plot units desired.
- INCM<=0 does not change plot units
- INCM=1 selects inches as plot units
- INCM>1 selects centimetres as plot units.

Neat line plotting.

A rather trivial, but frequently used, entry is provided to draft a neat line around the data area. To execute NEATLN:

```
CALL NEATLN
```

NEATLN parameter description.

No parameters.

Axis labeling.

For type REAL data four routines are provided for linear or logarithmic axis labeling. The calls for each routine are:

```
CALL XAXIS(DXP,DYP,XP,DEL,IP,SIZE,FMT,NFMT)
```
for linear X-axis labeling,
```
CALL YAXIS(DYP,DXP,YP,DEL,IP,SIZE,FMT,NFMT)
```
for linear Y-axis labeling,
```
CALL XAXISL(DXP,DYP,XP,SIZE,FMT,NFMT)
```
for logarithmic X-axis labeling, and
```
CALL YAXISL(DYP,DXP,YP,SIZE,FMT,NFMT)
```
for Y-axis logarithmic labeling.

For all four entries the parameters have identical usage except that DEL and IP do not apply to the logarithmic calls. An important item to remember when using the axis entries is that the plot area defined by SCALE should be sufficiently larger than the data area to accomodate the axis annotation.

Parameter description.

DXP, DYP, XP and YP are normally the same parameters as occurring in the last SCALE call. They provide the data range and physical size of the data area to the routines.

DEL is the interval (in data units) of the annotation tick. If DEL<=0 and NFMT>0 then only the maxima and minima of the data area are annotated.

IP is the interval between each labeling of the tick marks. If IP.<=.1 then each tick mark labelled.

SIZE is the height (in plot units) of the characters used in annotating the axis.

FMT is an array containing an object time FORMAT character string. If NFMT is zero, FMT may be any dummy value.

NFMT is the number of resultant characters created by the "WRITE"ing of the FMT FORMAT. Restriction: NFMT.<=.20.

File control.

The user may change the logical device name of the output file by using the plotting system entry PLTFIL. Principal usage of this entry are for plots always requiring a specific plotter and creation of "flash" files. To execute PLTFIL:

CALL PLTFIL(DEV,FIL,EXT,NO).

PLTFIL may be called only while the plotting system is not in an scaled state.

PLTFIL parameter description.

DEV is from 1 to 6 ASCII characters to be used as a logical or physical device name of the plotting output file. Default logical output device name is PLOT.

FIL is from 1 to 3 ASCII characters to be employed as a directory device name prefix. If FIL=0 then "PLT" assumed.

EXT is from 1 to 3 ASCII characters to be employed as a directory device name extension. If EXT=0 then "DAT" assumed.

NO is file sequence number used as a 3 digit suffix to the file name. If NO<0 then sequence number not altered. The value is initially zero.

If output is to be a directory device (ie. disk) the output file is generated with a default name of:

PLTxxx.DAT

where xxx (file sequence number) is initially zero and incremented by 1 for each new output file during the job. If the user is merely employing PLTFIL to force output to a single plotter dependent device like the TTY,

```
CALL PLTFIL('TTY',0,0,-1)
```

is adequate. Note that none of the plotter output devices is a directory device.

"Flash" plotting.

In production plotting, a portion of the plot may remain constant from one plot to another. It may be desirable, especially if the constant portion of the plot is complex, to create and save the plotter commands of this section and merely copy from this saved file to the actual plot file when needed. This process is defined here as "flashing".

Basically, the "flash" file is usually created as a separate job by extracting the constant portion of the plot generation code form the original program and, using PLTFIL, route the plotting output to disk. The original program is then modified by replacing the original constant plotting code with a call to the system copying routine FLASH. It should be noted that the flash file must be generated for the same plotter that is employed when the FLASH routine is called. To execute FLASH:

```
CALL FLASH(DEV,FIL,EXT,NO).
```

FLASH parameter description.

The parameters have the same usage as described in the PLTFIL call.

Character string justification.

It is often desirable to determine the number of characters and remove leading and trailing blanks from an unknown character string before plotting. Typical usage would be programmatic input of string data for titling which will centered on the plot. The routine LFJUST is provided for this purpose and is executed as:

```
CALL LFJUST(STRNG,N)
```

On entry STRNG is a string of N characters and upon return the characters in STRNG are left justified (no leading blanks) and N is the number of characters excluding trailing blanks.

EXAMPLE PROGRAM.

THIS IS A SIMPLE PROGRAM TO DEMONSTRATE SEVERAL OF THE COMMON CHARACTER AND LINE PLOTTING METHODS AND THE SCALING OF THE PLOT TO AN ARBITRARY OBJECT TIME SELECTED PLOTTER. THE RESULTANT PLOT IS SHOWN IN FIG. 2.

```
      DIMENSION XD(2),YD(2),XS(4),YS(4),XX(0/49),Y(0/49)
      DATA XD,YD/0.,6.2831853,-1.,1./
      DATA XS,YS/2*0.,1.,3*0.,1.,0./
C
C REQUEST PLOTTER NUMBER
      TYPE 10
   10 FORMAT(' ENTER PLOTTER NO.=',$)
      ACCEPT 20,IPLTR
   20 FORMAT(I)
C
C SELECT PLOTTER AND GET BOARD SIZE
      CALL PLTSET(IPLTR,XBD,YBD,0)
C
C RELATIVE SCALING
      XS(4)=AMIN1(6.,XBD)
      YS(4)=AMIN1(6.,YBD)
      XS(1)=XS(4)-1.1
      YS(1)=YS(4)-1.1
C
C SCALING CALL
      CALL SCALE(XD,YD,XS,YS,4,ICODE)
C
C QUIT IF ERROR
      IF (ICODE.NE.0) GO TO 40
C
C CREATE ARRAY OF SINE VALUES
      DO 30 I=0,49
      XX(I)=FLOAT(I)*.1282283
C
C EXAMPLE OF CONTINUED LINE CALL
      CALL LINE(XX(I),COS(XX(I)),1,1,1)
   30 Y(I)=SIN(XX(I))
C
C PLOT SINE CURVE ARRAY
      CALL LINE(XX,Y,50,0,0)
C
C POST SINE CURVE POINTS
      CALL CHAR(XX,Y,2,50,1,.08,0.,0.,0.)
C
C LABEL AXIS
      CALL XAXIS(XD,YD,XS,.2,5,.09,'(F4.1)',4)
      CALL YAXIS(YD,XD,YS,.1,10,.09,'(F4.1)',4)
      CALL NEATLN
      CALL CHAR((XD(1)+XD(2))*.5,YD(1),'VALUE',
     1 5,2,.12,0.,-.3,-.5)
      CALL CHAR(XD,(YD(1)+YD(2))*.5,'AMPLITUDE',
     1 9,2,.12,1.5706,-.54,.5)
```

```
      CALL CHAR(1.,.1,'PLOT EXAMPLE 1',14,7,.15,
     1 0.,0.,0.)
C
C DONE PLOTTING
      CALL ENDPLT(0)
      GO TO 60
C
   40 TYPE 50
   50 FORMAT(' CANNOT SCALE')
   60 STOP
      END
```

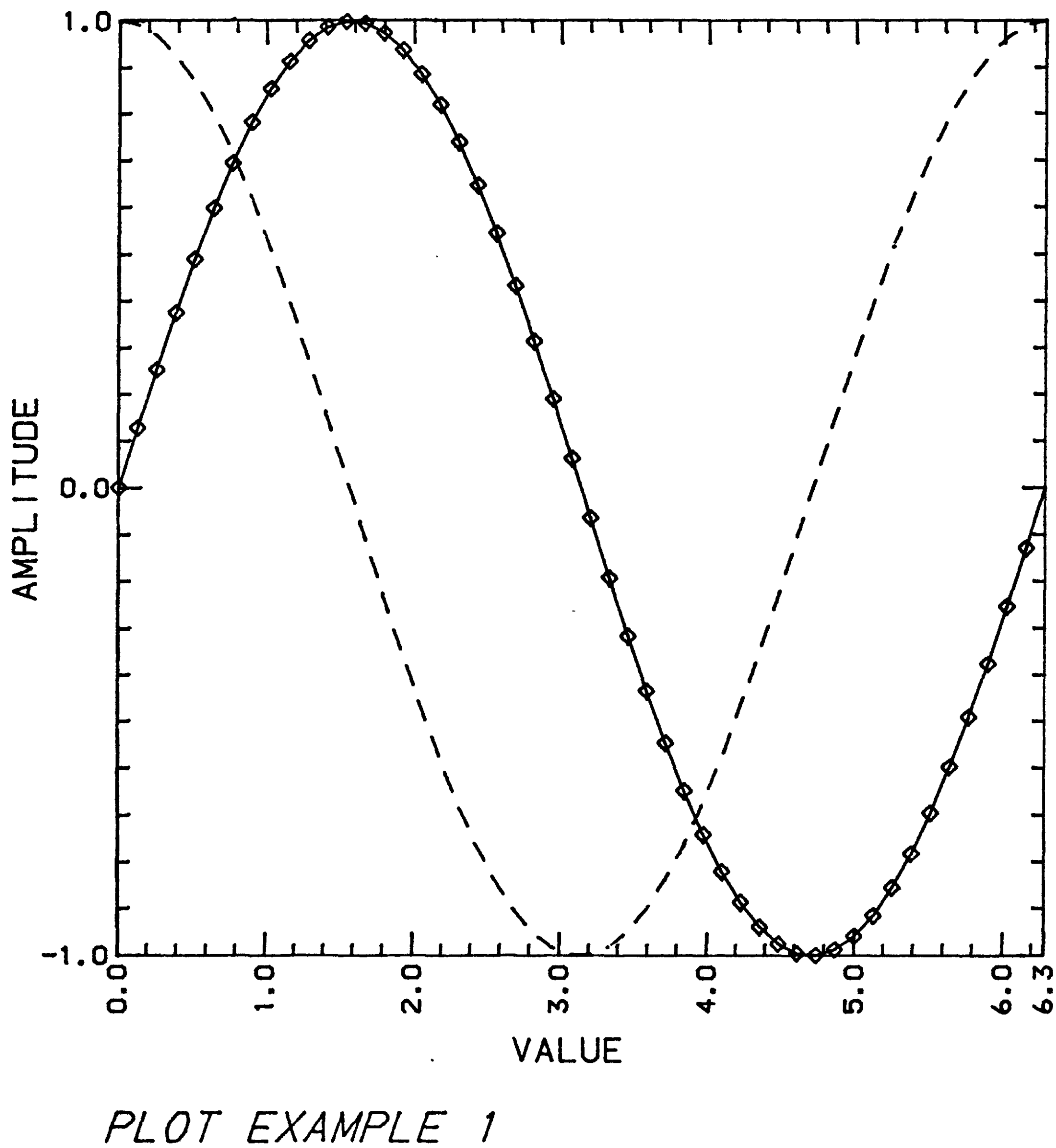


Figure 2. Resultant plot of example program.

PLOT OF PLOT SYSTEM CHARACTER SET.

000	0	0	040			100	@	@	140		
001	□	□	041	!	!	101	A	A	141	a	a
002	◇	◇	042	"	"	102	B	B	142	b	b
003	○	○	043	#	#	103	C	C	143	c	c
004	×	×	044	$	$	104	D	D	144	d	d
005	✳	✳	045	%	%	105	E	E	145	e	e
006	+	+	046	&	&	106	F	F	146	f	f
007	—	—	047	'	'	107	G	G	147	g	g
010	Y	Y	050	(	(	110	H	H	150	h	h
011	X̄	X̄	051	)	)	111	I	I	151	i	i
012	⧖	⧖	052	*	*	112	J	J	152	j	j
013	△	△	053	+	+	113	K	K	153	k	k
014	▽	▽	054	,	,	114	L	L	154	l	l
015	•	•	055	-	-	115	M	M	155	m	m
016			056	.	.	116	N	N	156	n	n
017			057	/	/	117	O	O	157	o	o
020			060	0	0	120	P	P	160	p	p
021			061	1	1	121	Q	Q	161	q	q
022			062	2	2	122	R	R	162	r	r
023			063	3	3	123	S	S	163	s	s
024			064	4	4	124	T	T	164	t	t
025			065	5	5	125	U	U	165	u	u
026			066	6	6	126	V	V	166	v	v
027			067	7	7	127	W	W	167	w	w
030	°	°	070	8	8	130	X	X	170	x	x
031	≠	≠	071	9	9	131	Y	Y	171	y	y
032	¢	¢	072	:	:	132	Z	Z	172	z	z
033	¬	¬	073	;	;	133	[	[	173	{	{
034	≈	≈	074	<	<	134	\	\	174	|	|
035	_	_	075	=	=	135	]	]	175	}	}
036	≤	≤	076	>	>	136	↑	↑	176	~	~
037	≥	≥	077	?	?	137	←	←	177		

INDEX

INDEX (CON'T)