UR1985-21

1/60

1985/21. Contouring using the Department of Mines Perkin-Elmer mini-computer. Part B: Gridded data.

R.G. Richardson

Abstract

The programs described use several versions of a contouring subroutine based on the method of Bengtsson and Nordbeck (1967) to contour already gridded data. The programs provide contour labelling and optional thickening of selected contour lines. The different versions of the subroutines cater for data quantities varying from only a few points up to a maximum limited only by the available disk storage and processor time.

USING THE PROGRAMS

Persons using these programs should be familiar with FORTRAN and may need to change the dimensions of AMAP, X and Y to suit a particular data set.

The programs all use logical unit 2 for the raw (ungridded) data, logical unit 3 for prompts for data entry, logical unit 5 for inputting map scale etc., logical unit 6 for outputting the levels successfully contoured, and logical unit 7 for reading the gridded data. In addition some of the programs require one or more work files.

Control data read from logical unit 5 is:

CONTLOW - the minimum contour value

CONTHIGH - the maximum contour value

CONTOUR INTERVAL - the contour interval

NPT - the number of places after the decimal point on the contour labels

NTHK - every NTHK contour starting at CONTLOW will be made thicker. If NTHK = 0 no contours will be thickened.

IOPT - select the functions desired from the program

(a) IOPT = 0
draw and label contours only

(b) IOPT = 1
draw and label contours and mark data points

(c) IOPT = 2
draw and label contours, mark data points and write data values at the data points.

SCALE - the plotting scale. *e.g.* 50 000
The labelling used for the map depends on the value of SCALE. If the map will not plot on the available paper width a new scale will be requested.

Sample plots for a small data set with 231 gridded points and labelling at the contour ends (Appendix 1) or along the contours (Appendix 2), and for a large data set with 2255 gridded points and labelling at contour ends (Appendix 3) or along the contours (Appendix 4) are shown.

THE PROGRAMS

NUCNT1 (Appendix 5)

This program is for contouring small volumes of data entirely within fast memory. It is used in conjunction with subroutine BORDER and either subroutine SMCONT for labelling contour ends or subroutine CNTLBL for labelling along the contour lines.

NUCNT2 (Appendix 6)

This program is for contouring medium volumes of data and uses logical unit 8 as a work file for the X and Y arrays. It is used in conjunction with subroutine BORDER and either subroutine SBCNT1 for labelling contour ends or subroutine CNTLBL1 for labelling along the contour lines.

NUCNT3 (Appendix 7)

This program is for contouring large volumes of data and uses logical unit 1 as a work file for the AMAP array and logical unit 8 as a work file for the X and Y arrays. It is used in conjunction with subroutine BORDER and either SBCNT2 for labelling contour ends or subroutine CNTLBL2 for labelling along the contour lines.

TEST TIMES

Program	*231 gridded points*	*2255 gridded points*
NUCNT1 + SMCONT	2.02	40.6
NUCNT1 + CNTLBL	2.18	42.2
NUCNT2 + SBCNT1	14.32	537.9
NUCNT2 + CNTLBL1	15.46	552.3
NUCNT3 + SBCNT2	20.55	706.3
NUCNT3 + CNTLBL2	21.81	720.0

All times are seconds of processor time.

ACKNOWLEDGEMENT

Dr R.J.G. Lewis wrote the program code for labelling along contour lines.

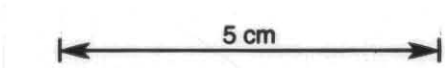
REFERENCE

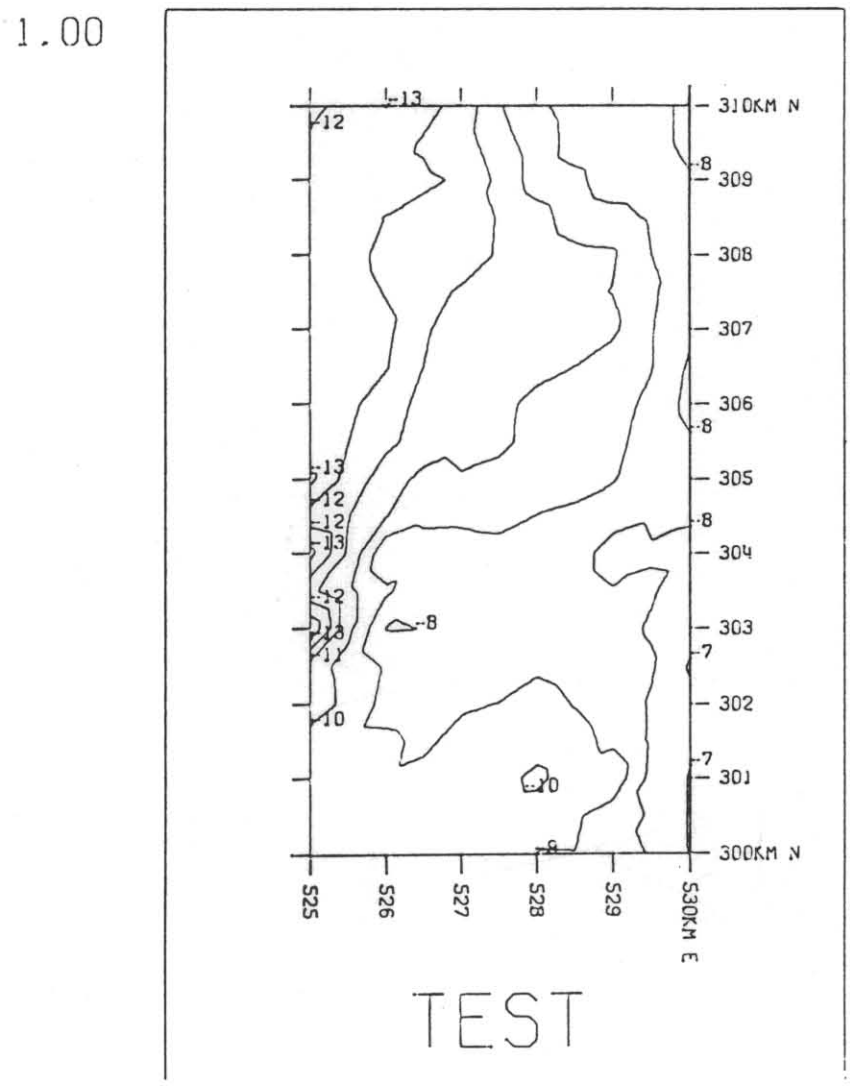
BENGTSSON, B.; NORDBECK, S. 1967. Construction of isarithms and isarithmic maps by computers. *Bit* 4:87-105.

[1 May 1985]

APPENDIX 1

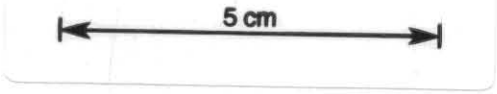
Sample plot : 231 gridded points, labelling on contour ends

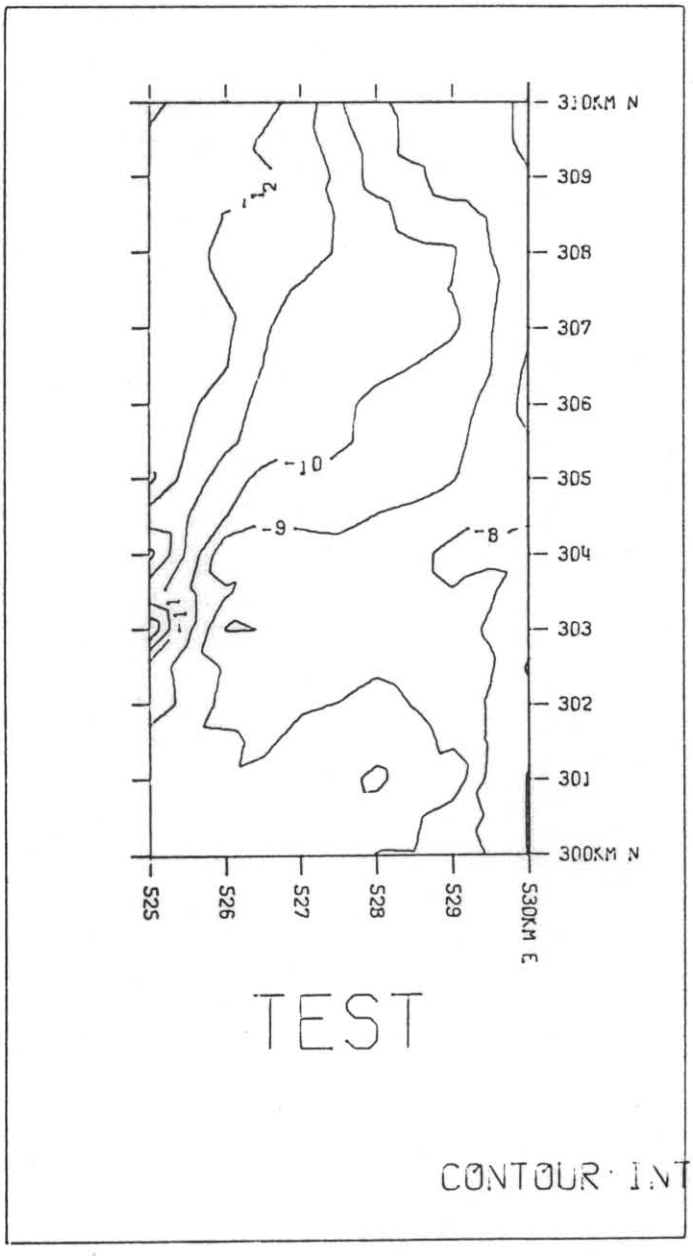




APPENDIX 2

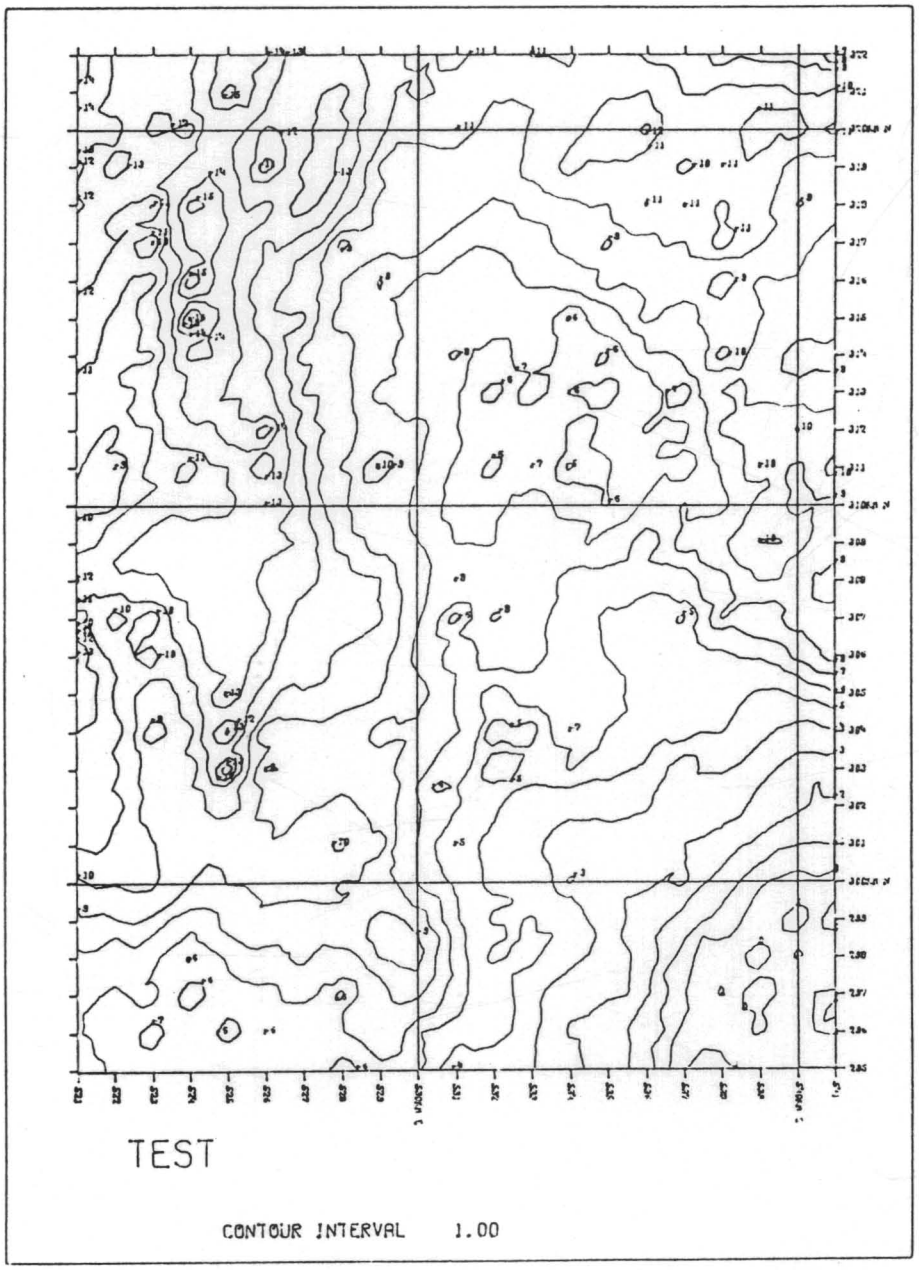
Sample plot : 231 gridded points, labelling along contours





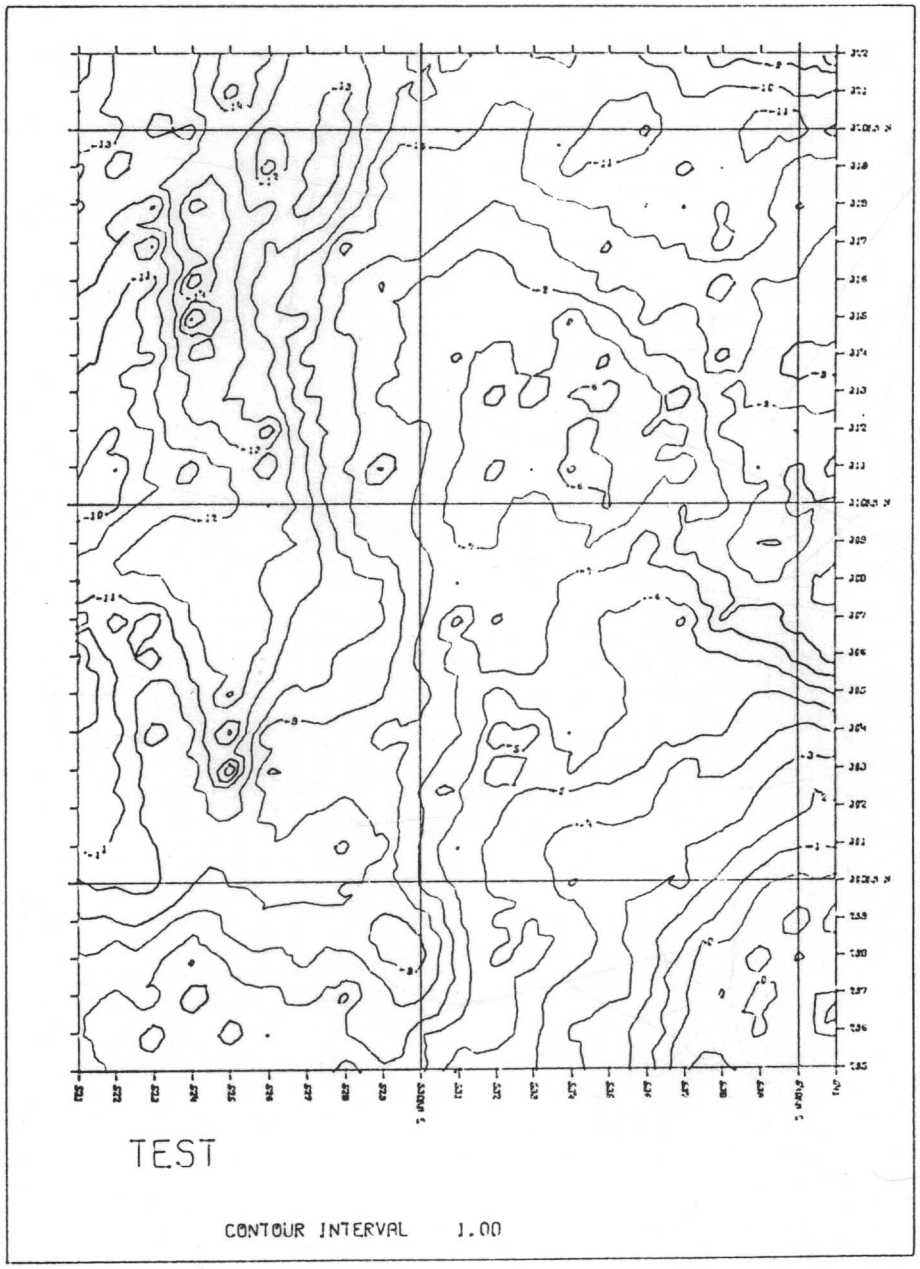
APPENDIX 3

Sample plot : 2255 gridded points, labelling at contour ends



APPENDIX 4

Sample plot : 2255 gridded points, labelling along the contours



7/60

APPENDIX 5

Program NUCNT1

```
L NUCNT1
AS 2,@1.DAT,ERO
AS 3,CON:
AS 5,CON:
AS 6,CON:
AS 7,@1.GRD,ERO
ST
$EXIT
```

```
$TITL NUCNT1 - A PROGRAM FOR CONTOURING SMALL QUANTITIES OF DATA
C LU  2  ORIGINAL FIELD DATA
C LU  3  PROMPTS FOR DATA ENTRY
C LU  5  UNIT FROM WHICH MAP SCALE,CONTOUR RANGE AND INTERVAL ARE READ
C LU  6  OUTPUT OF LAST LEVEL CONTOURED - VDU OR PRINTER
C LU  7  THE GRIDDED DATA
      DIMENSION AMAP(3000),TTLE(24),X(3000),Y(3000)
      COMMON /TRFORM/ SFACT
      COMMON /LABELC/SIZELB,FLAB,CX,CY,NSTRIN,STRING(15)
C USED BY THE FANCY CONTOURING PROGRAM TO SET LETTERING SIZE
      LOGICAL PNUP,THICK,LABL
      INTEGER*2 SURVEY(50),STRING
      EQUIVALENCE (AMAP(1),TTLE(1)),(TTLE(1),SURVEY(1))
C SFACT IS
C THE SCALE FACTOR TO CONVERT FROM CONTOUR COORDS TO PLOTTER INCHES
      SIZELB=0.07
C LABEL HEIGHT
      FLAB=3.5
C FACTOR BY WHICH LENGTH OF CONTOUR MUST EXCEED LENGTH OF LABEL
      REWIND 7
C GRID FILE
      CALL BORDER(NUMCON,ABOT,AINTT,TTLE,SURVEY,NPT,
     .  NTHK)
C GET ALL ARGUMENTS AND DRAW FRAME
      READ(7,101) NX,NY
  101 FORMAT(2I4)
      IF (NX*NY .GT. 3000) PAUSE 'TOO MANY POINTS'
      CALL DATAIN(AMAP,NX,NY)
C READ THE DATA FILE
      DO 20 I=1,NUMCON
      IF (NTHK .NE. 0) THEN
      THICK=MOD(I-1,NTHK) .EQ. 0
      ELSE
      THICK=.FALSE.
      ENDIF
C CHECK TO SEE THAT IT IS A CONTOUR TO BE THICKENED
      LABL=.TRUE.
C ASSUME ALL CONTOURS SHOULD BE LABELLED
      CALL CONTUR(ABOT,AMAP,NX,NY,1,NX,1,NY,X,Y,NPT,THICK,LABL)
      WRITE(6,201) ABOT
  201 FORMAT(' LEVEL',F10.3)
   20 ABOT=ABOT+AINTT
      CALL RSTR(2)
      END
      SUBROUTINE DATAIN(AMAP,NX,NY)
      DIMENSION AMAP(NX,NY)
      DO 10 I=1,NY
      READ(7,111) (AMAP(J,I),J=1,NX)
  111 FORMAT(8F9.2)
   10 CONTINUE
      RETURN
      END
$INCLUDE BORDCNT.FTN,-
$INCLUDE CNTLBL.FTN,-
```

```
      SUBROUTINE XFORM(A,B)
      COMMON /TRFORM/ SFACT
      A=A*SFACT
      B=B*SFACT
      RETURN
      END
      SUBROUTINE BORDER(NUMCON,ABOT,AINTT,TITLE,SURVEY,NPT,
     . NTHK)
      COMMON /TRFORM/ SFACT
      DIMENSION TITLE(18)
      INTEGER*2 SURVNO,NSURV,SURVEY(50),STATNO,AOK,YES,NO
      DOUBLE PRECISION XMIN,XMAX,YMIN,YMAX
      INTEGER*4 IXMIN,IXMAX,IYMIN,IYMAX,SPCE,DFACT,TXSP,DX,DY,
     . IFXMIN,IFXMAX,IFYMIN,IFYMAX,XSP,YSP
      LOGICAL SURV,LABEL,METRES,LINE
      DATA YES/1HY/,NO/1HN/
      REWIND 7
C REWIND GRIDDED DATA
      READ(7,101) XMIN,XMAX,YMIN,YMAX,SPACE,TITLE
  101 FORMAT(5F10.3/18A4)
      WRITE(3,108)
  108 FORMAT(' CONTLOWWW CONTHIGHH')
      READ(5,*) ABOT,ATOP
      WRITE(3,105)
  105 FORMAT(' CONTOUR INTERVAL')
      READ(5,*) AINTT
    9 WRITE(3,106)
  106 FORMAT(' NUMBER OF PLACES AFTER DEC. PT. ON CONTOUR LABELS')
      READ(5,*) NPT
      IF (NPT .GT. 6) GOTO 9
C TOO MANY PLACES
      IF (NPT .LE. 0) NPT=-1
C IF DONT WANT DECIMAL PLACES DONT WANT DEC. PT.
   22 WRITE(3,110)
  110 FORMAT(' EVERY N TH. CONTOUR HEAVY')
      READ(5,*) NTHK
      IF (NTHK .LT. 0) GOTO 22
C MAKE EVERY NTHK CONTOUR THICK
      WRITE(3,103)
  103 FORMAT(' OPTIONS'/
     .  '  0  CONTOURS ONLY'/
     .  '  1  STATION LOCATIONS'/
     .  '  2  STATION NUMBERS')
      READ(5,*) IOPT
      CALL INITAL(9,200,36,0,0,0)
      NSURV=1
C COUNTER FOR NUMBER OF SURVEYS
      NUMCON=((ATOP-ABOT)/AINTT)+1.0
C NUMBER OF CONTOURS
      SURV=.FALSE.
C
      XMIN=XMIN*1000.0D00
      IXMIN=XMIN
      XMAX=XMAX*1000.0D00
      IXMAX=XMAX
C CONVERT TO METRES
      YMIN=YMIN*1000.0D00
      IYMIN=YMIN
      YMAX=YMAX*1000.0D00
```

```
      IYMAX=YMAX
C CONVERT TO METRES
  102 WRITE(3,100)
  100 FORMAT(' SCALE E.G.  50000.')
      READ(5,*) SCALE
      FACT=1.E+2/(2.54*SCALE)
C THE FACTOR TO CONVERT FROM GRID METRES TO PLOTTER INCHES
      SFACT=FACT*1000.0
C TO CONVER FROM GRID KM TO PLOTTER INCHES
      LABEL=.TRUE.
      METRES=.TRUE.
      IF (SCALE .LE.  2001.) GOTO 851
      IF (SCALE .LE.  5001.) GOTO 852
      IF (SCALE .LE.  25005.) GOTO 853
      IF (SCALE .LE.  100010.) GOTO 854
C
C COPE WITH SCALE GREATER THAN 100000.  HERE
      METRES=.FALSE.
      DFACT=10000
      SPCE=100000
      TXSP=10000
      GOTO 855
C
C SCALE .LE.  100000,SCALE .GT.  25000
  854 METRES=.FALSE.
      DFACT=1000
      TXSP=1000
      SPCE=10000
      GOTO 855
C
C SCALE .LE.  25000,SCALE .GT.  5000
  853 DFACT=250
      TXSP=250
      SPCE=1000
      GOTO 855
C
C SCALE .LE.  5000,SCALE .GT.  2000
  852 DFACT=100
      TXSP=100
      SPCE=500
      GOTO 855
C
C SCALE .LE.  2000
  851 DFACT=100
      LABEL=.FALSE.
      TXSP=10
      SPCE=100
  855 CONTINUE
      IFXMIN=((IXMIN+1)/DFACT)*DFACT
      FXMIN=IFXMIN
      IFXMAX=((IXMAX+DFACT-1)/DFACT)*DFACT
      FXMAX=IFXMAX
      IFYMIN=((IYMIN+1)/DFACT)*DFACT
      FYMIN=IFYMIN
      IFYMAX=((IYMAX+DFACT-1)/DFACT)*DFACT
      FYMAX=IFYMAX
C GET THE RANGES AND ROUND UP AND DOWN TO THE NEAREST PIP
C GET THE MAXIMUM Y VALUE TO WORK OUT IF THE PLOT CAN POSSIBLY
C FIT ON THE PAGE
      YTMP=FYMAX
```

```
      CALL GDFORM(YTMP,FYMIN,FACT)
      YTMP=YTMP+2.1
C GET TOTAL WIDTH OF NEW PLOT INCLUDING TICKS AND LABELS
      IF (YTMP .LE. 35.5) GOTO 870
C WILL FIT ON PAPER
      WRITE(3,230)
  230 FORMAT(' PLOT WILL NOT FIT')
      GOTO 102
  870 CONTINUE
      CALL PLOT(1.0,2.0,-3)
C MOVE TO 1.0,2.0 AND CALL IT 0.,0.
C
      FACTMK=1.0
C THE FACTOR FOR SCALING THE SYMBOLS
  232 WRITE(3,233)
  233 FORMAT(' SMALL SYMBOLS? (Y OR N)')
      READ(5,229) AOK
  229 FORMAT(A1)
      IF (AOK .EQ. NO) GOTO 234
      IF (AOK .NE. YES) GOTO 232
      FACTMK=0.5
  234 CONTINUE
C
C
C NOW FOR SOME AXES
      DX=IFXMIN-TXSP
   10 DX=DX+TXSP
      XTMP=DX
      CALL GDFORM(XTMP,FXMIN,FACT)
      CALL PLOT(XTMP,0.,2)
C MOVE TO START OF PIP
      LINE=MOD(DX,SPCE) .EQ. 0
C CHECK TO SEE IF NEED TO DRAW LINE
      CALL XLABEL(DX,FYMIN,FYMAX,FACT,XTMP,LABEL,LINE,METRES)
      IF (IFXMAX .GT. DX) GOTO 10
C GO ALONG BOTTOM AND FINISH AT A PIP WITH DX=FXMAX,XTMP=FXMAX IN PLOTTER INCHES
C
C UP R.H.S.
      DY=IFYMIN-TXSP
   11 DY=DY+TXSP
      YTMP=DY
      CALL GDFORM(YTMP,FYMIN,FACT)
      CALL PLOT(XTMP,YTMP,2)
      LINE=MOD(DY,SPCE) .EQ. 0
C CHECK TO SEE IF MULTIPLE
      CALL YLABEL(DY,FXMIN,FXMAX,FACT,YTMP,LABEL,LINE,METRES)
      IF (IFYMAX .GT. DY) GOTO 11
C NOW GONE UP R.H.S.   DY=FYMAX
C                        YTMP=FYMAX IN PLOTTER COORDS.
C
      CALL PENDN
      XUP=XTMP+0.8
      YUP=YTMP+0.5
C USED FOR DRAWING FRAME
   12 XTMP=DX
      CALL GDFORM(XTMP,FXMIN,FACT)
      CALL PLOT(XTMP,YTMP,1)
      CALL PLOT(XTMP,YTMP+0.1,1)
      CALL PLOT(XTMP,YTMP,1)
      DX=DX-TXSP
```

12/60

```
      IF (DX .GE. IFXMIN) GOTO 12
C DRAWN PIPS ALONG TOP
C
   14 YTMP=DY
      CALL GDFORM(YTMP,FYMIN,FACT)
      CALL PLOT(0.,YTMP,1)
      CALL PLOT(-0.1,YTMP,1)
      CALL PLOT(0.,YTMP,1)
      DY=DY-TXSP
      IF (DY .GE. IFYMIN) GOTO 14
C DRAWN PIPS DOWN L.H.S.
      CALL PLOT(-0.75,-2.0,3)
      CALL PLOT(XUP,-2.0,2)
      CALL PLOT(XUP,YUP,1)
      CALL PLOT(-0.75,YUP,1)
      CALL PLOT(-0.75,-2.0,1)
C DRAW FRAME
      CALL PENUP
      CALL PWRITE(0.5,-1.0,0.28,0.,TITLE)
C THE TITLE IS PACKED BY THE READ
C WRITE THE TITLE
      ENCODE(SURVEY,301) AINTT
  301 FORMAT('CONTOUR INTERVAL',F8.2,'&')
      DECODE(SURVEY,302) TITLE
  302 FORMAT(18A4)
      CALL PWRITE(1.5,-1.75,0.14,0,TITLE)
C SET UP THE CONTOUR INTERVAL AND WRITE
      IF (IOPT .EQ. 0) GOTO 24
C SKIP IF NOT PLOTTING STATIONS
C
C NOW TO GO THROUGH THE DATA
      REWIND 2
C REWIND RAW DATA FILE
   30 READ(2,104,END=50) SURVNO,STATNO,X,Y
  104 FORMAT(I5,1X,I4,2F9.1)
      IF (X .LT. XMIN .OR. X .GT. XMAX .OR. Y .LT. YMIN
     . .OR. Y .GT. YMAX) GOTO 30
C CHECK TO SEE IF IN AREA
      IF (SURV) GOTO 53
      SURV=.TRUE.
      SURVEY(1)=SURVNO
   53 CONTINUE
      DO 15 ISYMB=1,NSURV
      IF (SURVEY(ISYMB) .EQ. SURVNO) GOTO 20
C ALREADY KNOW THIS NUMBER
   15 CONTINUE
      NSURV=NSURV+1
      ISYMB=NSURV
      SURVEY(NSURV)=SURVNO
C ADD NEW SURVEY NUMBER TO INDEX
   20 CONTINUE
      ASURV=STATNO
C GOT STATION NUMBER ALMOST AS AN INTEGER
      CALL GDFORM(X,FXMIN,FACT)
      CALL GDFORM(Y,FYMIN,FACT)
      CALL PLOT(X,Y,1)
      CALL FACTOR(FACTMK)
      IF (ISYMB .GT. 14) ISYMB=14
C ONLY HAVE 14 SYMBOLS AVAILABLE
      CALL MARKER(ISYMB)
```

```
      CALL FACTOR(1.)
      IF (IOPT .EQ. 1) GOTO 30
C ONLY WANTED THE MARKER
      IF (IOPT .EQ. 2) CALL
     . NUMBER(X+0.07,Y,FACTMK*0.07,ASURV,0.,-1)
      GOTO 30
C PLOTTED AND LABELLED STATION
   50 CONTINUE
C END OF RUN WHEN GET TO HERE SO TIDY UP A LITTLE
      IF (.NOT. SURV) GOTO 24
C NO STATIONS IN AREA
      IF (NSURV .EQ. 1 .AND. SURVEY(1) .EQ. 0) GOTO 24
C HAD DATA WITHOUT SURVEY NUMBERS
      X=0.5
      Y=-1.2
      DO 51 I=1,NSURV
      IMIN=12000
      DO 52 J=1,NSURV
      IF (SURVEY(J) .EQ. 0) GOTO 52
C ALREADY AND PRINTED IN ORDER
      IF (SURVEY(J) .GT. IMIN) GOTO 52
C ALREADY LOOKING AT AN EARLIER SURVEY
      IMIN=SURVEY(J)
      ISYMB=J
   52 CONTINUE
      CALL PLOT(X,Y,1)
      J=ISYMB
      IF (ISYMB .GT. 14) ISYMB=14
      CALL MARKER(ISYMB)
      ASURV=IMIN
      CALL NUMBER(X+0.15,Y-0.07,.14,ASURV,0.,-1)
      SURVEY(J)=0
      X=X+1.0
C AND SPACE ALONG A BIT
   51 CONTINUE
   24 CONTINUE
      XTMP=XMIN
      YTMP=YMIN
      CALL GDFORM(XTMP,FXMIN,FACT)
      CALL GDFORM(YTMP,FYMIN,FACT)
      CALL PLOT(XTMP,YTMP,1)
      SFACT=SFACT*SPACE
      CALL PLOT(SFACT,SFACT,0)
C GET CONVERSION FROM CONTOUR COORDS TO PLOTTER INCHES
C AND RESET ORIGIN
      RETURN
      END
      SUBROUTINE GDFORM(X,XBASE,FACT)
C FIND THE POSITION OF X RELATIVE TO XBASE IN PLOTTER INCHES
C AND RETURNS THE ANSWER IN X!!!!!!!!!!
      X=(X-XBASE)*FACT
      RETURN
      END
      SUBROUTINE XLABEL(IX,YMIN,YMAX,FACT,XT,LABEL,LINE,METRES)
      LOGICAL LABEL,LINE,METRES
      REAL TITLE(2)
      DATA TITLE/3HM E,4HKM E/
      CALL PLOT(XT,-0.1,1)
      IF (.NOT. LABEL .AND. .NOT. LINE) GOTO 10
C ONLY WANT PIP
```

```
      IIX=IX
      IF (.NOT. METRES) IIX=IIX/1000
      CALL NUMPRT(XT-0.035,-0.15,IIX,270.)
      IF (.NOT. LINE) GOTO 10
      CALL WHERE(XX,YY,AX)
C FIND THE PEN
      CALL MOVREL
      IF (METRES) CALL SYMBOL(XX,YY,0.07,TITLE(1),270.,3)
      IF (.NOT. METRES) CALL SYMBOL(XX,YY,0.07,TITLE(2),270.,4)
      CALL MOVABS
      YY=YMAX
      CALL GDFORM(YY,YMIN,FACT)
      CALL PLOT(XT,YY+0.1,1)
      CALL PENDN
   10 CALL PLOT(XT,0.,1)
      RETURN
      END
      SUBROUTINE YLABEL(IY,XMIN,XMAX,FACT,YT,LABEL,LINE,METRES)
      LOGICAL LINE,LABEL,METRES
      REAL TITLE(2)
      DATA TITLE/3HM N,4HKM N/
C USE FOR LABELLING IN SYMBOL
      X=XMAX
      CALL GDFORM(X,XMIN,FACT)
      CALL PLOT(X+0.1,YT,1)
      IF (.NOT. LABEL .AND. .NOT. LINE) GOTO 10
C ONLY WANTED PIP
      IIY=IY
      IF (.NOT. METRES) IIY=IIY/1000
      CALL NUMPRT(X+0.15,YT-0.035,IIY,0.0)
      IF (.NOT. LINE) GOTO 10
      CALL WHERE(XX,YY,AY)
C FIND PEN
      CALL MOVREL
      IF (METRES) CALL SYMBOL(XX,YY,0.07,TITLE(1),0.0,3)
      IF (.NOT. METRES) CALL SYMBOL(XX,YY,0.07,TITLE(2),0.,4)
      CALL MOVABS
      CALL PLOT(-0.1,YT,1)
      CALL PENDN
   10 CALL PLOT(X,YT,1)
      RETURN
      END
      SUBROUTINE NUMPRT(X,Y,IVAR,ANG)
C TO WRITE THE INTEGER IVAR WITH NO LEADING BLANKS
C STARTING AT X,Y AT AN ANGLE ANG
C A HEIGHT OF 0.07 IS ASSUMED
      INTEGER*4 BUFF(3)
      ENCODE(BUFF,100) IVAR
  100 FORMAT(I10,'&')
      K=0
      DO 10 I=1,11
      CALL ILBYTE(IB,BUFF,I-1)
      IF (IB .EQ. 32) GOTO 10
C SKIP SPACES
      CALL ISBYTE(IB,BUFF,K)
      K=K+1
   10 CONTINUE
      CALL PWRITE(X,Y,0.07,ANG,BUFF)
      RETURN
      END
```

```
$TITL  SMCONT - SIMPLE CONTOURING PACKAGE
      SUBROUTINE CONTUR(VALUE,AMAP,M,N,IX1,IX2,IY1,IY2,X,Y,NPT,THICK,
     .    LABL)
C VALUE IS THE CONTOUR LEVEL
C AMAP CONTAINS THE DATA TO BE USED STORED WITH THE RIGHT HAND SUBSCRIPT
C    REPRESENTING THE Y VALUES
C IX1,IX2 THE STARTING AND STOPPING X VALUES IN AMAP
C IY1,IY2 THE STARTING AND STOPPING Y VALUES IN AMAP
C X,Y ARE USED AS WORKSPACE AND SHOULD BE DIMENSIONED TO
C   (IX2-IX1)*(IY2-IY1)
C CONTOURS ARE LABELLED USING A CALL OF NUMBER WITH NPT FIGURES AFTER THE
C DECIMAL POINT IN ACCORDANCE WITH NUMBER
C IF A HEAVY CONTOUR IS NEEDED SET THICK=.TRUE.
C THE USER MUST SUPPLY THEIR OWN XFORM ROUTINE
C A0 VARIES FROM IX1 TO IX2
C B0 VARIES FROM IY1 TO IY2
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,OPEPS
      DIMENSION AMAP(M,N),X(5),Y(5)
      LOGICAL FIRST,THICK,LABL
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C SEARCH ALONG RHS BOUNDARY
      V1=AMAP(IX1,IY2)
      I1=IX1
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=AMAP(I,IY2)
      CALL INTERP(I1,IY2,I,IY2,V1,V2,X,Y,M*N)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=AMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=AMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2,X,Y,M*N)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=AMAP(I1,J1)
      V2=AMAP(I,J1)
      DO 3 J=IY1P,IY2
      CALL INTERP(I1,J1,I,J1,V1,V2,X,Y,M*N)
      V3=AMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3,X,Y,M*N)
      V4=AMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4,X,Y,M*N)
```

```
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
C WRITE A FILEMARK
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      A0=X(IT)
      B0=Y(IT)
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
C     AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C      SEARCH
  300 GOTO 405
  301 EXTA=X(IP)
      EXTB=Y(IP)
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      X(IP)=X(IBOT)
      X(IBOT)=ASTA
      Y(IP)=Y(IBOT)
      Y(IBOT)=ASTB
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      X(IP)=X(ITOP)
      X(ITOP)=ASTA
      Y(IP)=Y(ITOP)
      Y(ITOP)=ASTB
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
```

```
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      X(IB)=X(I)
      Y(IB)=Y(I)
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C      DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
      GOTO 510
  505 I=IBOT+1
  510 ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE TO THE EAST THEN TO THE NORTH
      A0=X(I)
      B0=Y(I)
      CALL XFORM(A0,B0)
      CALL PENUP
      IF (ITHK .EQ. 1) CALL NUMBER(A0,B0,0.07,VALUE,0.,NPT)
C     WRITE VALUE ON CONTOUR
      CALL PLOT(A0+XBASE,B0+YBASE,1)
      CALL PENDN
      J=ITOP-1
      IIP=IBOT+2
      IF(IIP .GT. IB) GOTO 521
      DO 520 IP=IIP,IB
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
  520 CALL PLOT(A0+XBASE,B0+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
  530 CALL PLOT(A0+XBASE,B0+YBASE,1)
  531 CALL PENUP
      IF (.NOT. THICK) GOTO 536
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
  536 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2,X,Y,LENTH)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION X(5),Y(5)
      LOGICAL L1,L2
      Z1=V1
```

18/60

```
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. LENTH) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X(ICOUNT)=FIA1+Z1*FIA2L1
      Y(ICOUNT)=FIB1+Z1*FIB2L1
      RETURN
      END
```

19/60

```
$TITL  CNTLBL - CONTOURING PACKAGE WITH LETTERING ALONG CONTOURS
      SUBROUTINE CONTUR(VALUE,AMAP,M,N,IX1,IX2,IY1,IY2,X,Y,NPT,
     1 THICK,LABL)
C      THIS ROUTINE DRAWS THE CONTOUR AT LEVEL VALUE.  THE DATA TO BE
C     CONTOURED IS IN AMAP WHICH IS A DOUBLY DIMENSIONED ARRAY IN WHICH
C      THE DATA TO BE CONTOURED IS STORED.  THERE ARE M ROWS AND N
C     COLUMNS.  ONLY THE PART OF THE DATA IN ROW RANGE (IX1,IX2) AND COL
C     RANGE (IY1,IY2) IS CONTOURED SO THAT LARGE AREAS CAN BE PARTITION-
C      ED.  POINTS ARE GENERATED AT THE CONTOUR LEVEL BY LINEAR INTERPOL-
C      ATION ALONG LINES AS SHOWN:
C
C          I----I----I----I
C          I   /I   /I   /I
C          I  / I  / I  / I  R. J. G. L.  27/10/78.
C          I /  I /  I /  I
C          I/---I/---I/---I
C          I   /I   /I   /I
C          I  / I  / I  / I
C          I /  I /  I /  I
C          I/---I/---I/---I
C
C      AND THEN SORTED FOR PLOTTING.  AMBIGUITIES ARE AVOIDED BY
C      PERTURBING POINTS AT THE CONTOUR LEVEL.
C      THE VALUES ARE STORED IN ARRAYS X,Y WHICH SHOULD HAVE ADEQUATE
C      BOUNDS:- IT IS NOT KNOWN FOR THE NUMBER OF POINTS TO EXCEED
C      (IX2-IX1)*(IY2-IY1).  LENGTH IS THE LENGTH OF X,Y.
C      ICOLOR IS THE COLOUR IN WHICH TO DRAW IF OUTPUT IS TO A
C      COLOUR DISPLAY.
C      LABL IS A LOGICAL VARIABLE INDICATING WHETHER THE CONTOUR IS
C      TO BE LABELLED.
C
C      THE USER MUST SUPPLY A SUBROUTINE XFORM(X,Y) WHICH THIS ROUTINE
C     SUPPLIES X,Y AS COORDINATES OF THE NEXT POINT TO BE DRAWN AND
C     XFORM RETURNS AS THE CORRESPONDING COORDINATES IN PLOTTER STEPS.
C     AT THE CONTOUR LEVEL.
C
C      REFERENCE:-
C          CONSTRUCTION OF ISARITHMS AND ISARITHMIC MAPS BY COMPUTERS.
C          BENGT-ERIK BENGTSSON AND STIG NORDBECK.  BIT 4 (1967) PP87-105
C
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,
     1 OPEPS
      DIMENSION AMAP(M,N),X(5),Y(5)
      LOGICAL FIRST,LABL,THICK
      IF (LABL) CALL SLABEL(VALUE,NPT)
C SET UP THE LABEL FOR THIS CONTOUR
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C HERE FIND ALL POINTS AT THE CONTOUR LEVEL BY GRID SEARCH
C SEARCH ALONG RHS BOUNDARY
      V1=AMAP(IX1,IY2)
      I1=IX1
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=AMAP(I,IY2)
```

```
      CALL INTERP(I1,IY2,I,IY2,V1,V2,X,Y,M*N)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=AMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=AMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2,X,Y,M*N)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=AMAP(I1,J1)
      V2=AMAP(I,J1)
      DO 3 J=IY1P,IY2
      CALL INTERP(I1,J1,I,J1,V1,V2,X,Y,M*N)
      V3=AMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3,X,Y,M*N)
      V4=AMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4,X,Y,M*N)
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      A0=X(IT)
      B0=Y(IT)
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
C     AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
```

```
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C       SEARCH
  300 GOTO 405
  301 EXTA=X(IP)
      EXTB=Y(IP)
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      X(IP)=X(IBOT)
      X(IBOT)=ASTA
      Y(IP)=Y(IBOT)
      Y(IBOT)=ASTB
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      X(IP)=X(ITOP)
      X(ITOP)=ASTA
      Y(IP)=Y(ITOP)
      Y(ITOP)=ASTB
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      X(IB)=X(I)
      Y(IB)=Y(I)
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C       DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
      GOTO 510
  505 I=IBOT+1
  510 J=ITOP-1
      IF(LABL) GOTO 530
      CALL CODRAW(X,Y,I,IBOT+2,IB,IT,J,THICK)
      GOTO 550
  530 CALL CLABEL(X,Y,I,IBOT+2,IB,IT,J,THICK)
  550 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2,X,Y,LENTH)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION X(5),Y(5)
      LOGICAL L1,L2
      Z1=V1
```

```
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. LENTH) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X(ICOUNT)=FIA1+Z1*FIA2L1
      Y(ICOUNT)=FIB1+Z1*FIB2L1
      RETURN
      END
      SUBROUTINE CODRAW(X,Y,I,II,IB,IT,J,THICK)
C       DRAW AN UN-ANNOTATED CONTOUR...
      DIMENSION X(5),Y(5)
      LOGICAL THICK
      ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE EAST THEN NORTH
      A0=X(I)
      B0=Y(I)
      CALL XFORM(A0,B0)
      CALL PENUP
      CALL PLOT(A0+XBASE,B0+YBASE,1)
      CALL PENDN
      IF(II .GT. IB) GOTO 521
      DO 520 IP=II,IB
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
  520 CALL PLOT(A0+XBASE,B0+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
  530 CALL PLOT(A0+XBASE,B0+YBASE,1)
  531 CALL PENUP
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE CLABEL(X,Y,I,IIP,IB,IT,J,THICK)
```

```
C       DRAW, IF POSSIBLE, AN ANNOTATED CONTOUR.  IF NOT POSSIBLE DRAW IT
C       WITHOUT AN ANNOTATION
        COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
        COMMON/ROOMYC/II,JIB,JIT,JJ,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
        DIMENSION XCEN(18),YCEN(18),X(5),Y(5)
       1 ,IISAVE(18),ISSAVE(18),SXCEN(18),SYCEN(18)
        LOGICAL CYCLE2,ROOM,THICK
        INTEGER*2 STRING
        CYCLE2=.FALSE.
C       SET UP SOME SCALE INFO...  CHARACTERS ARE ISIZE*IX BY ISIZE*IY
C       STEPS = CX BY CY IN COORDS.
        XL=0.
        YL=0.
        CALL XFORM(XL,YL)
        A=1.
        B=1.
        CALL XFORM(A,B)
        CY=SIZELB/ABS(XL-A)
C INTERCHANGE CX AND CY AS COORDS ROTATED**************************
        CX=SIZELB/ABS(YL-B)
        XL=X(I)
        YL=Y(I)
        II=IIP
        JIB=IB
        JIT=IT
        JJ=J
        REQX=CX
        REQY=CY
        NSTRI=NSTRIN+2
        ICC=0
        FOM=0.
        N=0
        DO 1 K=1,NSTRI
        CALL GETROO(X,Y)
        IF(.NOT. ROOM) GOTO 50
        ICC=ICC+1
        XCEN(K)=XC
        YCEN(K)=YC
        IISAVE(K)=N
        IF(K .GT. 1) FOM=FOM+ABS(YC-YCEN(K-1))
     1  CONTINUE
C       LABEL AT LEAST FITS ...  BEST SPOT?
        IPOINT=1
        JPOINT=2
        LPOINT=NSTRI
        KPOINT=NSTRI
        GOTO 25
    20  CALL GETROO(X,Y)
        IF(.NOT. ROOM) GOTO 500
        ICC=ICC+1
        FOM=FOM-ABS(YCEN(JPOINT)-YCEN(IPOINT))+ABS(YC-YCEN(LPOINT))
        XCEN(IPOINT)=XC
        YCEN(IPOINT)=YC
        IPOINT=IPOINT+1
        IF(IPOINT .GT. NSTRI) IPOINT=1
        JPOINT=JPOINT+1
        IF(JPOINT .GT. NSTRI) JPOINT=1
        LPOINT=LPOINT+1
        IF(LPOINT .GT. NSTRI) LPOINT=1
        KPOINT=KPOINT+1
```

```
      IF(KPOINT .GT. NSTRI) KPOINT=1
      IISAVE(KPOINT)=N
      IF(FOM .GE. FOMS) GOTO 20
C     HERE HAVE A BETTER SITE
   25 FOMS=FOM
      IPS=IPOINT
      KPS=KPOINT
      MX=NSTRI+1
      DO 57 M=1,MX
      SXCEN(M)=XCEN(M)
      SYCEN(M)=YCEN(M)
      ISSAVE(M)=IISAVE(M)
   57 CONTINUE
      GOTO 20
C     LABEL IMPOSSIBLE ........
   50 CALL CODRAW(X,Y,I,IIP,IB,IT,J,THICK)
      RETURN
C      HERE HAVE OPTIMUM LABEL SITE, TOTAL CHARACTERS AVAILABLE (ICC)
C     CENTER COORDS FOR CHARACTERS AND INTERVAL SSTART/END FOR LABEL.
C      DOES LENGTH EXCEED LABEL LENGTH BY SPECIFIED FACTOR?
  500 NREQ=F*FLOAT(NSTRI)
      IF(ICC .LE. NREQ) GOTO 50
      IPOINT=IPS
      NEND=ISSAVE(KPS)
      KPS=KPS+1
      IF(KPS .GT. NSTRI) KPS=1
      NST=ISSAVE(KPS)
      CALL ANNOT8(X,Y,I,IIP,IB,IT,J,NST,NEND,SXCEN,SYCEN,IPOINT,THICK)
      RETURN
      END
      SUBROUTINE GETROO(X,Y)
C      FIND ROOM FOR NEXT CHARACTER IF POSSIBLE. .
      COMMON/ROOMYC/II,IB,IT,J,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
      DIMENSION X(5),Y(5)
      LOGICAL CYCLE2,ROOM
      DX=0.
      DY=0.
      XL0=XL
      YL0=YL
      IF(CYCLE2) GOTO 250
       IF(II .GT. IB) GOTO 150
      DO 100 IP=II,IB
      DX=DX+X(IP)-XL
      DY=DY+Y(IP)-YL
      IF(ABS(DX) .GE. REQX) GOTO 500
      IF(ABS(DY) .GE. REQY) GOTO 510
      N=N+1
      XL=X(IP)
      YL=Y(IP)
  100 CONTINUE
  150 II=IT
      CYCLE2=.TRUE.
  250 IF(II .GT. J) GOTO 251
      DO 200 IP=II,J
      DX=DX+X(IP)-XL
      DY=DY+Y(IP)-YL
      IF(ABS(DX) .GE. REQX) GOTO 500
      IF(ABS(DY) .GE. REQY) GOTO 510
      N=N+1
      XL=X(IP)
```

```
      YL=Y(IP)
  200 CONTINUE
  251 ROOM=.FALSE.
      RETURN
C      FOUND ROOM....
  500 SXL=XL
      ADX=SIGN((ABS(DX)-REQX),(X(IP)-XL))
      XL=X(IP)-ADX
      YL=Y(IP)-ADX*(Y(IP)-YL)/(X(IP)-SXL)
      GOTO 550
  510 SYL=YL
      ADY=SIGN((ABS(DY)-REQY),(Y(IP)-YL))
      YL=Y(IP)-ADY
      XL=X(IP)-ADY*(X(IP)-XL)/(Y(IP)-SYL)
  550 XC=(XL0+XL)*0.5
      YC=(YL0+YL)*0.5
      ROOM=.TRUE.
      II=IP
      RETURN
      END
      SUBROUTINE ANNOT8(X,Y,I,II,IB,IT,J,NST,NEND,XC,YC,IPOINT,THICK)
C      DRAW AN ANNOTATED CONTOUR...
      DIMENSION XC(5),YC(5),X(5),Y(5)
      LOGICAL INLABL,THICK
      ITHK=1
  515 XBASE=0.0
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE
      INLABL=.FALSE.
      N=0
      NEND=NST+1
C WE WILL SET NEND TI ITS PROPER VALUE IN STRUNG AS THE CHARACTERS
C ARE WRITTEN OUT
      A0=X(I)
      B0=Y(I)
      CALL XFORM(A0,B0)
      CALL PENUP
      CALL PLOT(A0+XBASE,B0+YBASE,1)
      CALL PENDN
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
      IF(II .GT. IB) GOTO 521
      DO 520 IP=II,IB
      IF(INLABL) GOTO 510
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,2)
  510 N=N+1
      IF (N .EQ. NEND) INLABL=.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  520 CONTINUE
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      IF(INLABL) GOTO 525
      A0=X(IP)
      B0=Y(IP)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,2)
```

```
  525 N=N+1
      IF (N .EQ. NEND) INLABL =.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  530 CONTINUE
  531 CALL PLOT(A0+XBASE,B0+YBASE,3)
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE STRUNG(INLABL,XCEN,YCEN,IPOINT,NEND,ITHK)
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      DIMENSION XCEN(5),YCEN(5)
      LOGICAL INLABL
      INTEGER*2 STRING
      XBASE=0.0
      YBASE=0.0
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
      INLABL=.TRUE.
      NSTRI=NSTRIN+2
C     LAST MOVE WITH PEN DOWN
      A0=XCEN(IPOINT)
      B0=YCEN(IPOINT)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,2)
C      IF SLOPE IS TOO STEEP WRITE CHARACTERS FROM S-N,OTHERWISE
C      HORIZONTALLY.
      JPOINT=IPOINT-1
      IF(JPOINT .LT. 1) JPOINT=NSTRI
      DIR=0.0
      IF(ABS(YCEN(JPOINT)-YCEN(IPOINT)) .GT. ABS(XCEN(JPOINT)
     * -XCEN(IPOINT))) DIR=90.0
      XOFFSE=-0.5
      IF (DIR .GE. 89.) XOFFSE=-XOFFSE
C     DO WE SCAN LABEL OUT FORWARDS/BACKWARDS?
      JPOINT=IPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
      KPOINT=JPOINT+1
      IF (KPOINT .GT. NSTRI) KPOINT=1
      IF(XCEN(KPOINT) .GE. XCEN(JPOINT) .AND. DIR .EQ. 0.) GOTO 50
      IF(YCEN(KPOINT) .GE. YCEN(JPOINT) .AND. DIR .NE. 0.) GOTO 50
C     SCAN OUT BACKWARDS
      INDX=NSTRIN
      INC=-1
      GOTO 51
   50 INDX=1
      INC=1
   51 DO 100 I=1,NSTRIN
      A0=XCEN(JPOINT)+XOFFSE*CX
      B0=YCEN(JPOINT)-0.5*CY
      CALL XFORM(A0,B0)
      IF (ITHK.EQ.1) CALL SYMBOL(A0,B0,SIZELB,STRING(INDX),DIR,1)
      INDX=INDX+INC
      NEND=NEND+1
C USED TO COUNT WHERE THE STRING ENDS - SET INITIALLY TO THE START
C OF THE STRING
      JPOINT=JPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
  100 CONTINUE
```

```
C       AND MOVE PEN OUT OF LABEL AND DOWN...
        A0=XCEN(JPOINT)
        B0=YCEN(JPOINT)
        CALL XFORM(A0,B0)
        CALL PLOT(A0+XBASE,B0+YBASE,3)
        CALL PENDN
        RETURN
        END
        SUBROUTINE SLABEL(VALUE,NPT)
        COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
        DIMENSION FMT(2),BUFFR(4)
        INTEGER*2 STRING,BLAH,ZERO,POINT
C
        DATA BLAH,ZERO,POINT/1H ,1H0,1H./
C SET UP FORMAT TO MATCH NUMBER OF DECIMAL POINTS
        IF (NPT .LT. 0) NPT=0
C IN OTHER VERSIONS THAT USE A CALL OF NUMBER NPT=-1 IMPLIES NO
C DECIMAL POINT IN THE NUMBER
        ENCODE(BUFFR,100) NPT
  100   FORMAT('(F15.',I1,')')
        DECODE(BUFFR,101) FMT
  101   FORMAT(2A4)
C CAN DYNAMICALLY SET NO. OF DECIMAL PLACES
        ENCODE(BUFFR,FMT) VALUE
        DECODE(BUFFR,2) (STRING(K),K=1,15)
    2   FORMAT(80A1)
C        EDIT FURTHER
        DO 5 I=1,15
        IF(STRING(I) .NE. BLAH) GOTO 15
    5   CONTINUE
   15   J=1
        IF(I .EQ. 1) GOTO 20
        DO 25 K=J,15
        M=K+I-J
        STRING(K)=STRING(M)
   25   CONTINUE
   20   NSTRIN=15-I+1
C       ALSO REMOVE TRAILING ZEROS AFTER DECIMAL POINT...
        K=NSTRIN
        DO 50 I=1,NSTRIN
        IF(STRING(K) .NE. ZERO) GOTO 60
        K=K-1
   50   CONTINUE
   60   NSTRIN=K
        IF(STRING(NSTRIN) .EQ. POINT) NSTRIN=NSTRIN-1
        RETURN
        END
```

APPENDIX 6

Program NUCNT2

```
L NUCNT2
AS 2,@1.DAT,ERO
AS 3,CON:
AS 5,CON:
AS 6,CON:
AS 7,@1.GRD,ERO
TEMPFILE 8,IN,8/10/5
ST
$EXIT
```

```
$TITL NUCNT2 - A PROGRAM FOR CONTOURING MEDIUM QUANTITIES OF DATA
C LU  2  ORIGINAL FIELD DATA
C LU  3  PROMPTS FOR DATA ENTRY
C LU  5  UNIT FROM WHICH MAP SCALE, CONTOUR RANGE AND INTERVAL ARE READ
C LU  6  OUTPUT OF LAST LEVEL CONTOURED - VDU OR PRINTER
C LU  7  THE GRIDDED DATA
C LU  8  VIRTUAL ARRAY STORAGE FOR X, Y ARRAYS
      DIMENSION AMAP(3000), TTLE(24)
      COMMON /TRFORM/ SFACT
      COMMON /LABELC/SIZELB, FLAB, CX, CY, NSTRIN, STRING(15)
C USED BY THE FANCY CONTOURING PROGRAM TO SET LETTERING SIZE
      LOGICAL PNUP, THICK, LABL
      INTEGER*2 SURVEY(50), STRING
      EQUIVALENCE (AMAP(1), TTLE(1)), (TTLE(1), SURVEY(1))
C SFACT IS
C THE SCALE FACTOR TO CONVERT FROM CONTOUR COORDS TO PLOTTER INCHES
      SIZELB=0.07
C LABEL HEIGHT
      FLAB=3.5
C FACTOR BY WHICH LENGTH OF CONTOUR MUST EXCEED LENGTH OF LABEL
      REWIND 7
C GRID FILE
      CALL BORDER(NUMCON, ABOT, AINTT, TTLE, SURVEY, NPT,
     . NTHK)
C GET ALL ARGUMENTS AND DRAW FRAME
      READ(7, 101) NX, NY
  101 FORMAT(2I4)
      IF (NX*NY .GT. 3000) PAUSE 'TOO MANY POINTS'
      CALL DATAIN(AMAP, NX, NY)
C READ THE DATA FILE
      OPEN(UNIT=8, ACCESS='DIRECT', FORM='BINARY', RECL=8)
C SET UP THE VIRTUAL ARRAY FILE FOR X AND Y  RECORDLENGTH=8 BYTES
      DO 20 I=1, NUMCON
      IF (NTHK .NE. 0) THEN
      THICK=MOD(I-1, NTHK) .EQ. 0
      ELSE
      THICK=.FALSE.
      ENDIF
C CHECK TO SEE THAT IT IS A CONTOUR TO BE THICKENED
      LABL=.TRUE.
C ASSUME ALL CONTOURS SHOULD BE LABELLED
      CALL CONTUR(ABOT, AMAP, NX, NY, 1, NX, 1, NY, NPT, THICK, LABL)
      WRITE(6, 201) ABOT
  201 FORMAT(' LEVEL', F10.3)
   20 ABOT=ABOT+AINTT
      CALL RSTR(2)
      END
      SUBROUTINE DATAIN(AMAP, NX, NY)
      DIMENSION AMAP(NX, NY)
      DO 10 I=1, NY
      READ(7, 111) (AMAP(J, I), J=1, NX)
  111 FORMAT(8F9.2)
   10 CONTINUE
      RETURN
      END
$INCLUDE BORDCNT.FTN, -
$INCLUDE CNTLBL1.FTN, -
```

```
$TITL  SBCNT1 - SIMPLE CONTOURING PACKAGE FOR MEDIUM DATA VOLUMES
      SUBROUTINE CONTUR(VALUE,AMAP,M,N,IX1,IX2,IY1,IY2,NPT,THICK,
     .    LABL)
C VALUE IS THE CONTOUR LEVEL
C AMAP CONTAINS THE DATA TO BE USED STORED WITH THE RIGHT HAND SUBSCRIPT
C    REPRESENTING THE Y VALUES
C IX1,IX2 THE STARTING AND STOPPING X VALUES IN AMAP
C IY1,IY2 THE STARTING AND STOPPING Y VALUES IN AMAP
C X,Y ARE USED AS WORKSPACE AND SHOULD BE DIMENSIONED TO
C   (IX2-IX1)*(IY2-IY1)
C CONTOURS ARE LABELLED USING A CALL OF NUMBER WITH NPT FIGURES AFTER THE
C DECIMAL POINT IN ACCORDANCE WITH NUMBER
C IF A HEAVY CONTOUR IS NEEDED SET THICK=.TRUE.
C THE USER MUST SUPPLY THEIR OWN XFORM ROUTINE
C A0 VARIES FROM IX1 TO IX2
C B0 VARIES FROM IY1 TO IY2
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,OPEPS
      DIMENSION AMAP(M,N),XY(2)
      LOGICAL FIRST,THICK,LABL
      EQUIVALENCE (XY(1),X),(XY(2),Y)
C USED FOR VIRTUAL ARRAY
      REWIND 8
C POSITION THE VIRTUAL ARRAY AT THE START
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C HERE FIND ALL POINTS AT THE CONTOUR LEVEL BY GRID SEARCH
C SEARCH ALONG RHS BOUNDARY
      V1=AMAP(IX1,IY2)
      I1=IX1
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=AMAP(I,IY2)
      CALL INTERP(I1,IY2,I,IY2,V1,V2)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=AMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=AMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=AMAP(I1,J1)
      V2=AMAP(I,J1)
      DO 3 J=IY1P,IY2
```

```
      CALL INTERP(I1,J1,I,J1,V1,V2)
      V3=AMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3)
      V4=AMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4)
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
C WRITE A FILEMARK
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      READ(8,REC=IT) XY
      A0=X
      B0=Y
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
C     AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C      SEARCH
  300 GOTO 405
  301 READ(8,REC=IP) XY
      EXTA=X
      EXTB=Y
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      READ(8,REC=IBOT) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=IBOT) XY
      X=XTMP
      Y=YTMP
```

```
      WRITE(8,REC=IP) XY
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      READ(8,REC=ITOP) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=ITOP) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      READ(8,REC=I) XY
      WRITE(8,REC=IB) XY
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C      DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
      GOTO 510
  505 I=IBOT+1
  510 ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE TO THE EAST THEN TO THE NORTH
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
      IF (ITHK .EQ. 1) CALL NUMBER(X,Y,0.07,VALUE,0.,NPT)
C     WRITE VALUE ON CONTOUR
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      J=ITOP-1
      IIP=IBOT+2
      IF(IIP .GT. IB) GOTO 521
      DO 520 IP=IIP,IB
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  520 CALL PLOT(X+XBASE,Y+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      READ(8,REC=IP) XY
```

```
      CALL XFORM(X,Y)
  530 CALL PLOT(X+XBASE,Y+YBASE,1)
  531 CALL PENUP
      IF (.NOT. THICK) GOTO 536
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
  536 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION XY(2)
      LOGICAL L1,L2
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      Z1=V1
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. 100000) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X=FIA1+Z1*FIA2L1
      Y=FIB1+Z1*FIB2L1
      WRITE(8) XY
C USE A SEQUENTIAL WRITE HERE
      RETURN
      END
```

```
$TITL  CNTLBL1 - CONTOURING PACKAGE WITH LETTERING ALONG CONTOURS
C FOR CONTOURING MEDIUM VOLUMES OF DATA
      SUBROUTINE CONTUR(VALUE,AMAP,M,N,IX1,IX2,IY1,IY2,NPT,
     1 THICK,LABL)
C       THIS ROUTINE DRAWS THE CONTOUR AT LEVEL VALUE.  THE DATA TO BE
C      CONTOURED IS IN AMAP WHICH IS A DOUBLY DIMENSIONED ARRAY IN WHICH
C       THE DATA TO BE CONTOURED IS STORED.  THERE ARE M ROWS AND N
C      COLUMNS.  ONLY THE PART OF THE DATA IN ROW RANGE (IX1,IX2) AND COL
C      RANGE (IY1,IY2) IS CONTOURED SO THAT LARGE AREAS CAN BE PARTITION-
C       ED.  POINTS ARE GENERATED AT THE CONTOUR LEVEL BY LINEAR INTERPOL-
C       ATION ALONG LINES AS SHOWN:
C
C           I----I----I----I
C           I   /I   /I   /I
C           I  / I  / I  / I  R. J. G. L.  27/10/78.
C           I /  I /  I /  I
C           I/---I/---I/---I
C           I   /I   /I   /I
C           I  / I  / I  / I
C           I /  I /  I /  I
C           I/---I/---I/---I
C
C       AND THEN SORTED FOR PLOTTING.  AMBIGUITIES ARE AVOIDED BY
C       PERTURBING POINTS AT THE CONTOUR LEVEL.
C       THE VALUES ARE STORED IN VIRTUAL ARRAYS X,Y WHICH SHOULD HAVE ADEQUATE
C       BOUNDS:- IT IS NOT KNOWN FOR THE NUMBER OF POINTS TO EXCEED
C       (IX2-IX1)*(IY2-IY1).  LENGTH IS THE LENGTH OF X,Y.
C       LABL IS A LOGICAL VARIABLE INDICATING WHETHER THE CONTOUR IS
C       TO BE LABELLED.
C
C       THE USER MUST SUPPLY A SUBROUTINE XFORM(X,Y) WHICH THIS ROUTINE
C      SUPPLIES X,Y AS COORDINATES OF THE NEXT POINT TO BE DRAWN AND
C      XFORM RETURNS AS THE CORRESPONDING COORDINATES IN PLOTTER STEPS.
C      AT THE CONTOUR LEVEL.
C
C       REFERENCE:-
C           CONSTRUCTION OF ISARITHMS AND ISARITHMIC MAPS BY COMPUTERS.
C           BENGT-ERIK BENGTSSON AND STIG NORDBECK.  BIT 4 (1967) PP87-105
C
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,
     1 OPEPS
      DIMENSION AMAP(M,N),XY(2)
      LOGICAL FIRST,LABL,THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
C USED FOR THE VIRTUAL ARRAYS
      REWIND 8
C POSITION THE VIRTUAL ARRAY AT THE START
      IF (LABL) CALL SLABEL(VALUE,NPT)
C SET UP THE LABEL FOR THIS CONTOUR
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C HERE FIND ALL POINTS AT THE CONTOUR LEVEL BY GRID SEARCH
C SEARCH ALONG RHS BOUNDARY
      V1=AMAP(IX1,IY2)
      I1=IX1
```

```
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=AMAP(I,IY2)
      CALL INTERP(I1,IY2,I,IY2,V1,V2)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=AMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=AMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=AMAP(I1,J1)
      V2=AMAP(I,J1)
      DO 3 J=IY1P,IY2
      CALL INTERP(I1,J1,I,J1,V1,V2)
      V3=AMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3)
      V4=AMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4)
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      READ(8,REC=IT) XY
      A0=X
      B0=Y
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
```

```
C      AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C       SEARCH
  300 GOTO 405
  301 READ(8,REC=IP) XY
      EXTA=X
      EXTB=Y
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      READ(8,REC=IBOT) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=IBOT) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      READ(8,REC=ITOP) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=ITOP) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      READ(8,REC=I) XY
      WRITE(8,REC=IB) XY
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C       DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
```

```
      GOTO 510
  505 I=IBOT+1
  510 J=ITOP-1
      IF(LABL) GOTO 530
      CALL CODRAW(I,IBOT+2,IB,IT,J,THICK)
      GOTO 550
  530 CALL CLABEL(I,IBOT+2,IB,IT,J,THICK)
  550 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION XY(2)
      LOGICAL L1,L2
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      Z1=V1
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. 100000) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X=FIA1+Z1*FIA2L1
      Y=FIB1+Z1*FIB2L1
      WRITE(8) XY
C USE A SEQUENTIAL WRITE HERE
      RETURN
      END
      SUBROUTINE CODRAW(I,II,IB,IT,J,THICK)
C      DRAW AN UN-ANNOTATED CONTOUR...
      DIMENSION XY(2)
      LOGICAL THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE EAST THEN NORTH
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      IF(II .GT. IB) GOTO 521
```

```
      DO 520 IP=II,IB
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  520 CALL PLOT(X+XBASE,Y+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  530 CALL PLOT(X+XBASE,Y+YBASE,1)
  531 CALL PENUP
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE CLABEL(I,IIP,IB,IT,J,THICK)
C       DRAW, IF POSSIBLE, AN ANNOTATED CONTOUR.  IF NOT POSSIBLE DRAW IT
C       WITHOUT AN ANNOTATION
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      COMMON/ROOMYC/II,JIB,JIT,JJ,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
      DIMENSION XCEN(18),YCEN(18),XY(2)
     1 ,IISAVE(18),ISSAVE(18),SXCEN(18),SYCEN(18)
      LOGICAL CYCLE2,ROOM,THICK
      INTEGER*2 STRING
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      CYCLE2=.FALSE.
C       SET UP SOME SCALE INFO...  CHARACTERS ARE ISIZE*IX BY ISIZE*IY
C       STEPS = CX BY CY IN COORDS.
      XL=0.
      YL=0.
      CALL XFORM(XL,YL)
      A=1.
      B=1.
      CALL XFORM(A,B)
      CY=SIZELB/ABS(XL-A)
C INTERCHANGE CX AND CY AS COORDS ROTATED**************************
      CX=SIZELB/ABS(YL-B)
      READ(8,REC=I) XY
      XL=X
      YL=Y
      II=IIP
      JIB=IB
      JIT=IT
      JJ=J
      REQX=CX
      REQY=CY
      NSTRI=NSTRIN+2
      ICC=0
      FOM=0.
      N=0
      DO 1 K=1,NSTRI
      CALL GETROO
      IF(.NOT. ROOM) GOTO 50
      ICC=ICC+1
      XCEN(K)=XC
      YCEN(K)=YC
      IISAVE(K)=N
      IF(K .GT. 1) FOM=FOM+ABS(YC-YCEN(K-1))
    1 CONTINUE
C       LABEL AT LEAST FITS ...  BEST SPOT?
```

```
      IPOINT=1
      JPOINT=2
      LPOINT=NSTRI
      KPOINT=NSTRI
      GOTO 25
   20 CALL GETROO
      IF(.NOT. ROOM) GOTO 500
      ICC=ICC+1
      FOM=FOM-ABS(YCEN(JPOINT)-YCEN(IPOINT))+ABS(YC-YCEN(LPOINT))
      XCEN(IPOINT)=XC
      YCEN(IPOINT)=YC
      IPOINT=IPOINT+1
      IF(IPOINT .GT. NSTRI) IPOINT=1
      JPOINT=JPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
      LPOINT=LPOINT+1
      IF(LPOINT .GT. NSTRI) LPOINT=1
      KPOINT=KPOINT+1
      IF(KPOINT .GT. NSTRI) KPOINT=1
      IISAVE(KPOINT)=N
      IF(FOM .GE. FOMS) GOTO 20
C     HERE HAVE A BETTER SITE
   25 FOMS=FOM
      IPS=IPOINT
      KPS=KPOINT
      MX=NSTRI+1
      DO 57 M=1,MX
      SXCEN(M)=XCEN(M)
      SYCEN(M)=YCEN(M)
      ISSAVE(M)=IISAVE(M)
   57 CONTINUE
      GOTO 20
C     LABEL IMPOSSIBLE ........
   50 CALL CODRAW(I,IIP,IB,IT,J,THICK)
      RETURN
C      HERE HAVE OPTIMUM LABEL SITE, TOTAL CHARACTERS AVAILABLE (ICC)
C     CENTER COORDS FOR CHARACTERS AND INTERVAL SSTART/END FOR LABEL.
C      DOES LENGTH EXCEED LABEL LENGTH BY SPECIFIED FACTOR?
  500 NREQ=F*FLOAT(NSTRI)
      IF(ICC .LE. NREQ) GOTO 50
      IPOINT=IPS
      NEND=ISSAVE(KPS)
      KPS=KPS+1
      IF(KPS .GT. NSTRI) KPS=1
      NST=ISSAVE(KPS)
      CALL ANNOT8(I,IIP,IB,IT,J,NST,NEND,SXCEN,SYCEN,IPOINT,THICK)
      RETURN
      END
      SUBROUTINE GETROO
C      FIND ROOM FOR NEXT CHARACTER IF POSSIBLE..
      COMMON/ROOMYC/II,IB,IT,J,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
      DIMENSION XY(2)
      LOGICAL CYCLE2,ROOM
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      DX=0.
      DY=0.
      XL0=XL
      YL0=YL
      IF(CYCLE2) GOTO 250
       IF(II .GT. IB) GOTO 150
```

```
      DO 100 IP=II,IB
      READ(8,REC=IP) XY
      DX=DX+X-XL
      DY=DY+Y-YL
      IF(ABS(DX) .GE. REQX) GOTO 500
      IF(ABS(DY) .GE. REQY) GOTO 510
      N=N+1
      XL=X
      YL=Y
  100 CONTINUE
  150 II=IT
      CYCLE2=.TRUE.
  250 IF(II .GT. J) GOTO 251
      DO 200 IP=II,J
      READ(8,REC=IP) XY
      DX=DX+X-XL
      DY=DY+Y-YL
      IF(ABS(DX) .GE. REQX) GOTO 500
      IF(ABS(DY) .GE. REQY) GOTO 510
      N=N+1
      XL=X
      YL=Y
  200 CONTINUE
  251 ROOM=.FALSE.
      RETURN
C       FOUND ROOM....
  500 SXL=XL
      ADX=SIGN((ABS(DX)-REQX),(X-XL))
      XL=X-ADX
      YL=Y-ADX*(Y-YL)/(X-SXL)
      GOTO 550
  510 SYL=YL
      ADY=SIGN((ABS(DY)-REQY),(Y-YL))
      YL=Y-ADY
      XL=X-ADY*(X-XL)/(Y-SYL)
  550 XC=(XL0+XL)*0.5
      YC=(YL0+YL)*0.5
      ROOM=.TRUE.
      II=IP
      RETURN
      END
      SUBROUTINE ANNOT8(I,II,IB,IT,J,NST,NEND,XC,YC,IPOINT,THICK)
C       DRAW AN ANNOTATED CONTOUR..
      DIMENSION XC(5),YC(5),XY(2)
      LOGICAL INLABL,THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      ITHK=1
  515 XBASE=0.0
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE
      INLABL=.FALSE.
      N=0
      NEND=NST+1
C WE WILL SET NEND TI ITS PROPER VALUE IN STRUNG AS THE CHARACTERS
C ARE WRITTEN OUT
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
```

```
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
      IF(II .GT. IB) GOTO 521
      DO 520 IP=II,IB
      IF(INLABL) GOTO 510
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
      CALL PLOT(X+XBASE,Y+YBASE,2)
  510 N=N+1
      IF (N .EQ. NEND) INLABL=.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  520 CONTINUE
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      IF(INLABL) GOTO 525
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
      CALL PLOT(X+XBASE,Y+YBASE,2)
  525 N=N+1
      IF (N .EQ. NEND) INLABL =.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  530 CONTINUE
  531 CALL PLOT(X+XBASE,Y+YBASE,3)
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE STRUNG(INLABL,XCEN,YCEN,IPOINT,NEND,ITHK)
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      DIMENSION XCEN(5),YCEN(5)
      LOGICAL INLABL
      INTEGER*2 STRING
      XBASE=0.0
      YBASE=0.0
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
      INLABL=.TRUE.
      NSTRI=NSTRIN+2
C     LAST MOVE WITH PEN DOWN
      A0=XCEN(IPOINT)
      B0=YCEN(IPOINT)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,2)
C      IF SLOPE IS TOO STEEP WRITE CHARACTERS FROM S-N,OTHERWISE
C      HORIZONTALLY.
      JPOINT=IPOINT-1
      IF(JPOINT .LT. 1) JPOINT=NSTRI
      DIR=0.0
      IF(ABS(YCEN(JPOINT)-YCEN(IPOINT)) .GT. ABS(XCEN(JPOINT)
     * -XCEN(IPOINT))) DIR=90.0
      XOFFSE=-0.5
      IF (DIR .GE. 89.) XOFFSE=-XOFFSE
C     DO WE SCAN LABEL OUT FORWARDS/BACKWARDS?
      JPOINT=IPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
      KPOINT=JPOINT+1
      IF (KPOINT .GT. NSTRI) KPOINT=1
      IF(XCEN(KPOINT) .GE. XCEN(JPOINT) .AND. DIR .EQ. 0.) GOTO 50
```

```
      IF(YCEN(KPOINT) .GE. YCEN(JPOINT) .AND. DIR .NE. 0.) GOTO 50
C     SCAN OUT BACKWARDS
      INDX=NSTRIN
      INC=-1
      GOTO 51
   50 INDX=1
      INC=1
   51 DO 100 I=1,NSTRIN
      A0=XCEN(JPOINT)+XOFFSE*CX
      B0=YCEN(JPOINT)-0.5*CY
      CALL XFORM(A0,B0)
      IF (ITHK.EQ.1) CALL SYMBOL(A0,B0,SIZELB,STRING(INDX),DIR,1)
      INDX=INDX+INC
      NEND=NEND+1
C USED TO COUNT WHERE THE STRING ENDS - SET INITIALLY TO THE START
C OF THE STRING
      JPOINT=JPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
  100 CONTINUE
C     AND MOVE PEN OUT OF LABEL AND DOWN...
      A0=XCEN(JPOINT)
      B0=YCEN(JPOINT)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,3)
      CALL PENDN
      RETURN
      END
      SUBROUTINE SLABEL(VALUE,NPT)
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      DIMENSION FMT(2),BUFFR(4)
      INTEGER*2 STRING,BLAH,ZERO,POINT
C
      DATA BLAH,ZERO,POINT/1H ,1H0,1H./
C SET UP FORMAT TO MATCH NUMBER OF DECIMAL POINTS
      IF (NPT .LT. 0) NPT=0
C IN OTHER VERSIONS THAT USE A CALL OF NUMBER NPT=-1 IMPLIES NO
C DECIMAL POINT IN THE NUMBER
      ENCODE(BUFFR,100) NPT
  100 FORMAT('(F15.',I1,')')
      DECODE(BUFFR,101) FMT
  101 FORMAT(2A4)
C CAN DYNAMICALLY SET NO. OF DECIMAL PLACES
      ENCODE(BUFFR,FMT) VALUE
      DECODE(BUFFR,2) (STRING(K),K=1,15)
    2 FORMAT(80A1)
C      EDIT FURTHER
      DO 5 I=1,15
      IF(STRING(I) .NE. BLAH) GOTO 15
    5 CONTINUE
   15 J=1
      IF(I .EQ. 1) GOTO 20
      DO 25 K=J,15
      M=K+I-J
      STRING(K)=STRING(M)
   25 CONTINUE
   20 NSTRIN=15-I+1
C     ALSO REMOVE TRAILING ZEROS AFTER DECIMAL POINT...
      K=NSTRIN
      DO 50 I=1,NSTRIN
      IF(STRING(K) .NE. ZERO) GOTO 60
```

```
      K=K-1
   50 CONTINUE
   60 NSTRIN=K
      IF(STRING(NSTRIN) .EQ.  POINT) NSTRIN=NSTRIN-1
      RETURN
      END
```

APPENDIX 7

Program NUCNT3

```
L NUCNT3
TEMPFILE 1,IN,4/10/5
AS 2,@1.DAT,ERO
AS 3,CON:
AS 5,CON:
AS 6,CON:
AS 7,@1.GRD,ERO
XAL. XYFILEZ.DZZ,IN,8/10/5
AS 8,XYFILEZ.DZZ
ST
XDE XYFILEZ.DZZ
$EXIT
```

```
$TITL NUCNT3 - A PROGRAM FOR CONTOURING LARGE QUANTITIES OF DATA
C LU  1  VIRTUAL ARRAY FILE FOR GRIDDED DATA
C LU  2  ORIGINAL FIELD DATA
C LU  3  PROMPTS FOR DATA ENTRY
C LU  5  UNIT FROM WHICH MAP SCALE,CONTOUR RANGE AND INTERVAL ARE READ
C LU  6  OUTPUT OF LAST LEVEL CONTOURED - VDU OR PRINTER
C LU  7  THE GRIDDED DATA
C LU  8  VIRTUAL ARRAY STORAGE FOR X,Y ARRAYS
      DIMENSION AMAP(1000),TTLE(24)
      COMMON /TRFORM/ SFACT
      COMMON /LABELC/SIZELB,FLAB,CX,CY,NSTRIN,STRING(15)
C USED BY THE FANCY CONTOURING PROGRAM TO SET LETTERING SIZE
      LOGICAL PNUP,THICK,LABL
      INTEGER*2 SURVEY(50),STRING
      COMMON /ARRAYS/ MDIM,NDIM
      EQUIVALENCE (AMAP(1),TTLE(1)),(TTLE(1),SURVEY(1))
C SFACT IS
C THE SCALE FACTOR TO CONVERT FROM CONTOUR COORDS TO PLOTTER INCHES
      SIZELB=0.07
C LABEL HEIGHT
      FLAB=3.5
C FACTOR BY WHICH LENGTH OF CONTOUR MUST EXCEED LENGTH OF LABEL
      REWIND 7
C GRID FILE
      CALL BORDER(NUMCON,ABOT,AINTT,TTLE,SURVEY,NPT,
     .  NTHK)
C GET ALL ARGUMENTS AND DRAW FRAME
      READ(7,101) NX,NY
  101 FORMAT(2I4)
      IF (NX .GT. 1000) PAUSE ' DATA HAS TOO MANY VALUES PER LINE'
      MDIM=NX
      NDIM=NY
C STORE THE BOUNDS OF WHAT WILL BE THE VIRTUAL ARRAY AMAP
      OPEN(UNIT=1,ACCESS='DIRECT',FORM='BINARY',RECL=4)
C SET UP THE VIRTUAL ARRAY FILE FOR THE GRIDDED DATA
      CALL DATAIN(AMAP,NX,NY)
C READ THE DATA FILE
      OPEN(UNIT=8,ACCESS='DIRECT',FORM='BINARY',RECL=8)
C SET UP THE VIRTUAL ARRAY FILE FOR X AND Y  RECORDLENGTH=8 BYTES
      DO 20 I=1,NUMCON
      IF (NTHK .NE. 0) THEN
      THICK=MOD(I-1,NTHK) .EQ. 0
      ELSE
      THICK=.FALSE.
      ENDIF
C CHECK TO SEE THAT IT IS A CONTOUR TO BE THICKENED
      LABL=.TRUE.
C ASSUME ALL CONTOURS SHOULD BE LABELLED
      CALL CONTUR(ABOT,1,NX,1,NY,NPT,THICK,LABL)
      WRITE(6,201) ABOT
  201 FORMAT(' LEVEL',F10.3)
   20 ABOT=ABOT+AINTT
      CALL RSTR(2)
      END
      SUBROUTINE DATAIN(AMAP,NX,NY)
      DIMENSION AMAP(NX)
      DO 10 I=1,NY
      READ(7,111) (AMAP(J),J=1,NX)
  111 FORMAT(8F9.2)
```

```
      DO 11 J=1,NX
      CALL WRMAP(AMAP(J),J,I)
   11 CONTINUE
C WRITE TO THE VIRTUAL ARRAY FILE WITH THE FIRST INDEX
C VARYING MOST RAPIDLY
   10 CONTINUE
      RETURN
      END
      SUBROUTINE WRMAP(AMAP,M,N)
C WRITE AMAP(M,N) TO A FILE IN A MANNER HAVING M INCREASING MOST
C RAPIDLY
      COMMON /ARRAYS/ MDIM,NDIM
C MDIM IS THE IRST BOUND OF AMAP
C NDIM IS THE SECOND BOUND OF AMAP
      IREC=(N-1)*MDIM+M
      WRITE(1,REC=IREC) AMAP
      RETURN
      END
      REAL FUNCTION RDMAP(M,N)
      COMMON /ARRAYS/ MDIM,NDIM
      IREC=(N-1)*MDIM+M
      READ(1,REC=IREC) RDMAP
      RETURN
      END
$INCLUDE BORDCNT.FTN,-
$INCLUDE CNTLBL2.FTN,-
```

```
$TITL  SBCNT2 - SIMPLE CONTOURING PACKAGE FOR LARGE DATA VOLUMES
      SUBROUTINE CONTUR(VALUE,IX1,IX2,IY1,IY2,NPT,THICK,
     .    LABL)
C VALUE IS THE CONTOUR LEVEL
C AMAP CONTAINS THE DATA TO BE USED STORED WITH THE RIGHT HAND SUBSCRIPT
C    REPRESENTING THE Y VALUES
C IX1,IX2 THE STARTING AND STOPPING X VALUES IN AMAP
C IY1,IY2 THE STARTING AND STOPPING Y VALUES IN AMAP
C X,Y ARE USED AS WORKSPACE AND SHOULD BE DIMENSIONED TO
C   (IX2-IX1)*(IY2-IY1)
C CONTOURS ARE LABELLED USING A CALL OF NUMBER WITH NPT FIGURES AFTER THE
C DECIMAL POINT IN ACCORDANCE WITH NUMBER
C IF A HEAVY CONTOUR IS NEEDED SET THICK=.TRUE.
C THE USER MUST SUPPLY THEIR OWN XFORM ROUTINE
C A0 VARIES FROM IX1 TO IX2
C B0 VARIES FROM IY1 TO IY2
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,OPEPS
      DIMENSION XY(2)
      LOGICAL FIRST,THICK,LABL
      EQUIVALENCE (XY(1),X),(XY(2),Y)
C USED FOR VIRTUAL ARRAY
      REWIND 8
C POSITION THE VIRTUAL ARRAY AT THE START
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C HERE FIND ALL POINTS AT THE CONTOUR LEVEL BY GRID SEARCH
C SEARCH ALONG RHS BOUNDARY
      V1=RDMAP(IX1,IY2)
      I1=IX1
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=RDMAP(I,IY2)
      CALL INTERP(I1,IY2,I,IY2,V1,V2)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=RDMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=RDMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=RDMAP(I1,J1)
      V2=RDMAP(I,J1)
      DO 3 J=IY1P,IY2
```

```
      CALL INTERP(I1,J1,I,J1,V1,V2)
      V3=RDMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3)
      V4=RDMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4)
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
C WRITE A FILEMARK
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      READ(8,REC=IT) XY
      A0=X
      B0=Y
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
C     AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C      SEARCH
  300 GOTO 405
  301 READ(8,REC=IP) XY
      EXTA=X
      EXTB=Y
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      READ(8,REC=IBOT) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=IBOT) XY
      X=XTMP
      Y=YTMP
```

```
      WRITE(8,REC=IP) XY
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      READ(8,REC=ITOP) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=ITOP) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      READ(8,REC=I) XY
      WRITE(8,REC=IB) XY
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C       DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
      GOTO 510
  505 I=IBOT+1
  510 ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE TO THE EAST THEN TO THE NORTH
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
      IF (ITHK .EQ. 1) CALL NUMBER(X,Y,0.07,VALUE,0.,NPT)
C     WRITE VALUE ON CONTOUR
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      J=ITOP-1
      IIP=IBOT+2
      IF(IIP .GT. IB) GOTO 521
      DO 520 IP=IIP,IB
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  520 CALL PLOT(X+XBASE,Y+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      READ(8,REC=IP) XY
```

```
      CALL XFORM(X,Y)
  530 CALL PLOT(X+XBASE,Y+YBASE,1)
 '531 CALL PENUP
      IF (.NOT. THICK) GOTO 536
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
  536 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION XY(2)
      LOGICAL L1,L2
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      Z1=V1
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. 100000) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X=FIA1+Z1*FIA2L1
      Y=FIB1+Z1*FIB2L1
      WRITE(8) XY
C USE A SEQUENTIAL WRITE HERE
      RETURN
      END
```

```
$TITL  CNTLBL2 - CONTOURING PACKAGE WITH LETTERING ALONG CONTOURS
C FOR CONTOURING LARGE VOLUMES OF DATA
      SUBROUTINE CONTUR(VALUE,IX1,IX2,IY1,IY2,NPT,
     1 THICK,LABL)
C       THIS ROUTINE DRAWS THE CONTOUR AT LEVEL VALUE.  THE DATA TO BE
C      CONTOURED IS IN AMAP WHICH IS A DOUBLY DIMENSIONED ARRAY IN WHICH
C       THE DATA TO BE CONTOURED IS STORED.  THERE ARE M ROWS AND N
C      COLUMNS.  ONLY THE PART OF THE DATA IN ROW RANGE (IX1,IX2) AND COL
C      RANGE (IY1,IY2) IS CONTOURED SO THAT LARGE AREAS CAN BE PARTITION-
C       ED.  POINTS ARE GENERATED AT THE CONTOUR LEVEL BY LINEAR INTERPOL-
C       ATION ALONG LINES AS SHOWN:
C
C            I----I----I----I
C            I   /I   /I   /I
C            I  / I  / I  / I  R. J. G. L.  27/10/78.
C            I /  I /  I /  I
C            I/---I/---I/---I
C            I   /I   /I   /I
C            I  / I  / I  / I
C            I /  I /  I /  I
C            I/---I/---I/---I
C
C       AND THEN SORTED FOR PLOTTING.  AMBIGUITIES ARE AVOIDED BY
C       PERTURBING POINTS AT THE CONTOUR LEVEL.
C       THE VALUES ARE STORED IN VIRTUAL ARRAYS X,Y WHICH SHOULD HAVE ADEQUATE
C       BOUNDS:- IT IS NOT KNOWN FOR THE NUMBER OF POINTS TO EXCEED
C       (IX2-IX1)*(IY2-IY1).  LENGTH IS THE LENGTH OF X,Y.
C       LABL IS A LOGICAL VARIABLE INDICATING WHETHER THE CONTOUR IS
C       TO BE LABELLED.
C
C       THE USER MUST SUPPLY A SUBROUTINE XFORM(X,Y) WHICH THIS ROUTINE
C      SUPPLIES X,Y AS COORDINATES OF THE NEXT POINT TO BE DRAWN AND
C      XFORM RETURNS AS THE CORRESPONDING COORDINATES IN PLOTTER STEPS.
C      AT THE CONTOUR LEVEL.
C
C       REFERENCE:-
C           CONSTRUCTION OF ISARITHMS AND ISARITHMIC MAPS BY COMPUTERS.
C           BENGT-ERIK BENGTSSON AND STIG NORDBECK.  BIT 4 (1967) PP87-105
C
      COMMON/CONTRE/CONT,CEPS,OLEPS,ICOUNT,
     1 OPEPS
      DIMENSION XY(2)
      LOGICAL FIRST,LABL,THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
C USED FOR THE VIRTUAL ARRAYS
      REWIND 8
C POSITION THE VIRTUAL ARRAY AT THE START
      IF (LABL) CALL SLABEL(VALUE,NPT)
C SET UP THE LABEL FOR THIS CONTOUR
      CONT=VALUE
      EPS=2.0E-5
      CEPS=5.*EPS
      OLEPS=1.0-CEPS
      OPEPS=1.0+EPS
      ICOUNT=0
C HERE FIND ALL POINTS AT THE CONTOUR LEVEL BY GRID SEARCH
C SEARCH ALONG RHS BOUNDARY
      V1=RDMAP(IX1,IY2)
      I1=IX1
```

```
      IX1P=IX1+1
      DO 1 I=IX1P,IX2
      V2=RDMAP(I,IY2)
      CALL INTERP(I1,IY2,I,IY2,V1,V2)
      I1=I
      V1=V2
    1 CONTINUE
C SEARCH ALONG BOTTOM BOUNDARY
C
      V1=RDMAP(IX2,IY1)
      J1=IY1
      IY1P=IY1+1
      DO 2 J=IY1P,IY2
      V2=RDMAP(IX2,J)
      CALL INTERP(IX2,J1,IX2,J,V1,V2)
      J1=J
      V1=V2
    2 CONTINUE
C SEARCH THROUGH BODY OF DATA
C
      I1=IX1
      DO 4 I=IX1P,IX2
      J1=IY1
      V1=RDMAP(I1,J1)
      V2=RDMAP(I,J1)
      DO 3 J=IY1P,IY2
      CALL INTERP(I1,J1,I,J1,V1,V2)
      V3=RDMAP(I1,J)
      CALL INTERP(I1,J1,I1,J,V1,V3)
      V4=RDMAP(I,J)
      CALL INTERP(I1,J1,I,J,V1,V4)
      J1=J
      V1=V3
      V2=V4
    3 CONTINUE
      I1=I
    4 CONTINUE
C
C     HAVE ALL THE POINTS AT THE CONTOUR LEVEL
      IF(ICOUNT .EQ. 0) RETURN
C NO POINTS AT THIS LEVEL
C
      ITOP=1
      IBOT=ICOUNT
C      NEW
  100 FIRST=.TRUE.
      EPS=ABS(EPS)
      IT=ITOP
      IB=IBOT
      READ(8,REC=IT) XY
      A0=X
      B0=Y
      ASTA=0.0
      ASTB=0.0
      AMIN=FLOAT(IFIX(A0+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(B0-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP+1
      GOTO 300
```

```
C      AGAIN
  200 EPS=-EPS
      AMIN=FLOAT(IFIX(ASTA+EPS))-EPS
      AMAX=AMIN+1.0
      BMIN=FLOAT(IFIX(ASTB-EPS))+EPS
      BMAX=BMIN+1.0
      IP=ITOP
C       SEARCH
  300 GOTO 405
  301 READ(8,REC=IP) XY
      EXTA=X
      EXTB=Y
      IF(AMIN .GE. EXTA .OR. EXTA .GE. AMAX .OR. BMIN .GE. EXTB .OR.
     1 EXTB .GE. BMAX) GOTO 400
      ASTA=EXTA
      ASTB=EXTB
      IF(.NOT. FIRST) GOTO 320
      IF(IP .EQ. IBOT) GOTO 310
      READ(8,REC=IBOT) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=IBOT) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  310 IBOT=IBOT-1
      GOTO 200
  320 IF(IP .EQ. ITOP) GOTO 325
      READ(8,REC=ITOP) XY
      XTMP=X
      YTMP=Y
      X=ASTA
      Y=ASTB
      WRITE(8,REC=ITOP) XY
      X=XTMP
      Y=YTMP
      WRITE(8,REC=IP) XY
  325 ITOP=ITOP+1
      GOTO 200
  400 IP=IP+1
  405 IF(IP .LE. IBOT) GOTO 301
      IF(.NOT. FIRST) GOTO 500
      IF(ASTA .NE. A0 .OR. ASTB .NE. B0) GOTO 410
      IB=IB+1
      I=IBOT+1
      READ(8,REC=I) XY
      WRITE(8,REC=IB) XY
      GOTO 500
  410 FIRST=.FALSE.
      EPS=ABS(EPS)
      ITOP=ITOP+1
      ASTA=A0
      ASTB=B0
      GOTO 200
C       DRAW
  500 IF(IBOT .NE. IB) GOTO 505
      I=IT
      IT=IT+1
```

```
      GOTO 510
  505 I=IBOT+1
  510 J=ITOP-1
      IF(LABL) GOTO 530
      CALL CODRAW(I,IBOT+2,IB,IT,J,THICK)
      GOTO 550
  530 CALL CLABEL(I,IBOT+2,IB,IT,J,THICK)
  550 IF(ITOP .LE. IBOT) GOTO 100
      RETURN
      END
      SUBROUTINE INTERP(IA1,IB1,IA2,IB2,V1,V2)
      COMMON/CONTRE/CONTOR,EPS,OLEPS,ICOUNT,OPEPS
      DIMENSION XY(2)
      LOGICAL L1,L2
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      Z1=V1
      IF (Z1 .EQ. CONTOR) Z1=Z1*OPEPS
C THIS VALUE MAY NEED TO BE SET TO SUIT THE DATA
      Z2=V2
      IF (Z2 .EQ. CONTOR) Z2=Z2*OPEPS
C THIS VALUE MAY NEED TO BE CHANGED TO SUIT DATA
      IF (Z1 .EQ. Z2) RETURN
      L1=Z1 .LT. CONTOR
      L2=Z2 .GT. CONTOR
      IF((L1 .AND. .NOT. L2) .OR. (L2 .AND. .NOT. L1)) RETURN
      Z1=(Z1-CONTOR)/(Z1-Z2)
      IF(Z1 .GE. EPS) GOTO 2
      Z1=EPS
      GOTO 1
    2 IF(Z1 .GT. OLEPS) Z1=OLEPS
    1 ICOUNT=ICOUNT+1
      IF (ICOUNT .EQ. 100000) PAUSE 8809
      FIA1=IA1
      FIB1=IB1
      FIA2L1=IA2-IA1
      FIB2L1=IB2-IB1
      X=FIA1+Z1*FIA2L1
      Y=FIB1+Z1*FIB2L1
      WRITE(8) XY
C USE A SEQUENTIAL WRITE HERE
      RETURN
      END
      SUBROUTINE CODRAW(I,II,IB,IT,J,THICK)
C      DRAW AN UN-ANNOTATED CONTOUR...
      DIMENSION XY(2)
      LOGICAL THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      ITHK=1
C NO HEAVY LINE AS DEFAULT
  515 XBASE=0.
      YBASE=0.
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
C OFFSET LINE EAST THEN NORTH
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      IF(II .GT. IB) GOTO 521
```

```
      DO 520 IP=II,IB
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  520 CALL PLOT(X+XBASE,Y+YBASE,1)
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
  530 CALL PLOT(X+XBASE,Y+YBASE,1)
  531 CALL PENUP
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE CLABEL(I,IIP,IB,IT,J,THICK)
C       DRAW, IF POSSIBLE, AN ANNOTATED CONTOUR. IF NOT POSSIBLE DRAW IT
C       WITHOUT AN ANNOTATION
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      COMMON/ROOMYC/II,JIB,JIT,JJ,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
      DIMENSION XCEN(18),YCEN(18),XY(2)
     1 ,IISAVE(18),ISSAVE(18),SXCEN(18),SYCEN(18)
      LOGICAL CYCLE2,ROOM,THICK
      INTEGER*2 STRING
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      CYCLE2=.FALSE.
C       SET UP SOME SCALE INFO... CHARACTERS ARE ISIZE*IX BY ISIZE*IY
C       STEPS = CX BY CY IN COORDS.
      XL=0.
      YL=0.
      CALL XFORM(XL,YL)
      A=1.
      B=1.
      CALL XFORM(A,B)
      CY=SIZELB/ABS(XL-A)
C INTERCHANGE CX AND CY AS COORDS ROTATED**************************
      CX=SIZELB/ABS(YL-B)
      READ(8,REC=I) XY
      XL=X
      YL=Y
      II=IIP
      JIB=IB
      JIT=IT
      JJ=J
      REQX=CX
      REQY=CY
      NSTRI=NSTRIN+2
      ICC=0
      FOM=0.
      N=0
      DO 1 K=1,NSTRI
      CALL GETROO
      IF(.NOT. ROOM) GOTO 50
      ICC=ICC+1
      XCEN(K)=XC
      YCEN(K)=YC
      IISAVE(K)=N
      IF(K .GT. 1) FOM=FOM+ABS(YC-YCEN(K-1))
    1 CONTINUE
C       LABEL AT LEAST FITS ... BEST SPOT?
```

```
      IPOINT=1
      JPOINT=2
      LPOINT=NSTRI
      KPOINT=NSTRI
      GOTO 25
   20 CALL GETROO
      IF(.NOT. ROOM) GOTO 500
      ICC=ICC+1
      FOM=FOM-ABS(YCEN(JPOINT)-YCEN(IPOINT))+ABS(YC-YCEN(LPOINT))
      XCEN(IPOINT)=XC
      YCEN(IPOINT)=YC
      IPOINT=IPOINT+1
      IF(IPOINT .GT. NSTRI) IPOINT=1
      JPOINT=JPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
      LPOINT=LPOINT+1
      IF(LPOINT .GT. NSTRI) LPOINT=1
      KPOINT=KPOINT+1
      IF(KPOINT .GT. NSTRI) KPOINT=1
      IISAVE(KPOINT)=N
      IF(FOM .GE. FOMS) GOTO 20
C     HERE HAVE A BETTER SITE
   25 FOMS=FOM
      IPS=IPOINT
      KPS=KPOINT
      MX=NSTRI+1
      DO 57 M=1,MX
      SXCEN(M)=XCEN(M)
      SYCEN(M)=YCEN(M)
      ISSAVE(M)=IISAVE(M)
   57 CONTINUE
      GOTO 20
C     LABEL IMPOSSIBLE .....
   50 CALL CODRAW(I,IIP,IB,IT,J,THICK)
      RETURN
C      HERE HAVE OPTIMUM LABEL SITE, TOTAL CHARACTERS AVAILABLE (ICC)
C     CENTER COORDS FOR CHARACTERS AND INTERVAL SSTART/END FOR LABEL.
C      DOES LENGTH EXCEED LABEL LENGTH BY SPECIFIED FACTOR?
  500 NREQ=F*FLOAT(NSTRI)
      IF(ICC .LE. NREQ) GOTO 50
      IPOINT=IPS
      NEND=ISSAVE(KPS)
      KPS=KPS+1
      IF(KPS .GT. NSTRI) KPS=1
      NST=ISSAVE(KPS)
      CALL ANNOT8(I,IIP,IB,IT,J,NST,NEND,SXCEN,SYCEN,IPOINT,THICK)
      RETURN
      END
      SUBROUTINE GETROO
C      FIND ROOM FOR NEXT CHARACTER IF POSSIBLE..
      COMMON/ROOMYC/II,IB,IT,J,XL,YL,XC,YC,REQX,REQY,CYCLE2,ROOM,N
      DIMENSION XY(2)
      LOGICAL CYCLE2,ROOM
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      DX=0.
      DY=0.
      XL0=XL
      YL0=YL
      IF(CYCLE2) GOTO 250
       IF(II .GT. IB) GOTO 150
```

```
      DO 100 IP=II,IB
      READ(8,REC=IP) XY
      DX=DX+X-XL
      DY=DY+Y-YL
      IF(ABS(DX) .GE.  REQX) GOTO 500
      IF(ABS(DY) .GE.  REQY) GOTO 510
      N=N+1
      XL=X
      YL=Y
  100 CONTINUE
  150 II=IT
      CYCLE2=.TRUE.
  250 IF(II .GT.  J) GOTO 251
      DO 200 IP=II,J
      READ(8,REC=IP) XY
      DX=DX+X-XL
      DY=DY+Y-YL
      IF(ABS(DX) .GE.  REQX) GOTO 500
      IF(ABS(DY) .GE.  REQY) GOTO 510
      N=N+1
      XL=X
      YL=Y
  200 CONTINUE
  251 ROOM=.FALSE.
      RETURN
C      FOUND ROOM....
  500 SXL=XL
      ADX=SIGN((ABS(DX)-REQX),(X-XL))
      XL=X-ADX
      YL=Y-ADX*(Y-YL)/(X-SXL)
      GOTO 550
  510 SYL=YL
      ADY=SIGN((ABS(DY)-REQY),(Y-YL))
      YL=Y-ADY
      XL=X-ADY*(X-XL)/(Y-SYL)
  550 XC=(XL0+XL)*0.5
      YC=(YL0+YL)*0.5
      ROOM=.TRUE.
      II=IP
      RETURN
      END
      SUBROUTINE ANNOT8(I,II,IB,IT,J,NST,NEND,XC,YC,IPOINT,THICK)
C      DRAW AN ANNOTATED CONTOUR...
      DIMENSION XC(5),YC(5),XY(2)
      LOGICAL INLABL,THICK
      EQUIVALENCE (XY(1),X),(XY(2),Y)
      ITHK=1
  515 XBASE=0.0
      YBASE=0.
      IF (ITHK .EQ.  2) XBASE=0.01
      IF (ITHK .EQ.  3) YBASE=0.01
C OFFSET LINE
      INLABL=.FALSE.
      N=0
      NEND=NST+1
C WE WILL SET NEND TI ITS PROPER VALUE IN STRUNG AS THE CHARACTERS
C ARE WRITTEN OUT
      READ(8,REC=I) XY
      CALL XFORM(X,Y)
      CALL PENUP
```

```
      CALL PLOT(X+XBASE,Y+YBASE,1)
      CALL PENDN
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
      IF(II .GT. IB) GOTO 521
      DO 520 IP=II,IB
      IF(INLABL) GOTO 510
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
      CALL PLOT(X+XBASE,Y+YBASE,2)
  510 N=N+1
      IF (N .EQ. NEND) INLABL=.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  520 CONTINUE
  521 IF(IT .GT. J) GOTO 531
      DO 530 IP=IT,J
      IF(INLABL) GOTO 525
      READ(8,REC=IP) XY
      CALL XFORM(X,Y)
      CALL PLOT(X+XBASE,Y+YBASE,2)
  525 N=N+1
      IF (N .EQ. NEND) INLABL =.FALSE.
      IF(N .EQ. NST) CALL STRUNG(INLABL,XC,YC,IPOINT,NEND,ITHK)
  530 CONTINUE
  531 CALL PLOT(X+XBASE,Y+YBASE,3)
      IF (.NOT. THICK) RETURN
      ITHK=ITHK+1
      IF (ITHK .LE. 3) GOTO 515
      RETURN
      END
      SUBROUTINE STRUNG(INLABL,XCEN,YCEN,IPOINT,NEND,ITHK)
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      DIMENSION XCEN(5),YCEN(5)
      LOGICAL INLABL
      INTEGER*2 STRING
      XBASE=0.0
      YBASE=0.0
      IF (ITHK .EQ. 2) XBASE=0.01
      IF (ITHK .EQ. 3) YBASE=0.01
      INLABL=.TRUE.
      NSTRI=NSTRIN+2
C     LAST MOVE WITH PEN DOWN
      A0=XCEN(IPOINT)
      B0=YCEN(IPOINT)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,2)
C      IF SLOPE IS TOO STEEP WRITE CHARACTERS FROM S-N,OTHERWISE
C      HORIZONTALLY.
      JPOINT=IPOINT-1
      IF(JPOINT .LT. 1) JPOINT=NSTRI
      DIR=0.0
      IF(ABS(YCEN(JPOINT)-YCEN(IPOINT)) .GT. ABS(XCEN(JPOINT)
     * -XCEN(IPOINT))) DIR=90.0
      XOFFSE=-0.5
      IF (DIR .GE. 89.) XOFFSE=-XOFFSE
C     DO WE SCAN LABEL OUT FORWARDS/BACKWARDS?
      JPOINT=IPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
      KPOINT=JPOINT+1
      IF (KPOINT .GT. NSTRI) KPOINT=1
      IF(XCEN(KPOINT) .GE. XCEN(JPOINT) .AND. DIR .EQ. 0.) GOTO 50
```

```
      IF(YCEN(KPOINT) .GE. YCEN(JPOINT) .AND. DIR .NE. 0.) GOTO 50
C     SCAN OUT BACKWARDS
      INDX=NSTRIN
      INC=-1
      GOTO 51
   50 INDX=1
      INC=1
   51 DO 100 I=1,NSTRIN
      A0=XCEN(JPOINT)+XOFFSE*CX
      B0=YCEN(JPOINT)-0.5*CY
      CALL XFORM(A0,B0)
      IF (ITHK.EQ.1) CALL SYMBOL(A0,B0,SIZELB,STRING(INDX),DIR,1)
      INDX=INDX+INC
      NEND=NEND+1
C USED TO COUNT WHERE THE STRING ENDS - SET INITIALLY TO THE START
C OF THE STRING
      JPOINT=JPOINT+1
      IF(JPOINT .GT. NSTRI) JPOINT=1
  100 CONTINUE
C     AND MOVE PEN OUT OF LABEL AND DOWN...
      A0=XCEN(JPOINT)
      B0=YCEN(JPOINT)
      CALL XFORM(A0,B0)
      CALL PLOT(A0+XBASE,B0+YBASE,3)
      CALL PENDN
      RETURN
      END
      SUBROUTINE SLABEL(VALUE,NPT)
      COMMON/LABELC/SIZELB,F,CX,CY,NSTRIN,STRING(15)
      DIMENSION FMT(2),BUFFR(4)
      INTEGER*2 STRING,BLAH,ZERO,POINT
C
      DATA BLAH,ZERO,POINT/1H ,1H0,1H./
C SET UP FORMAT TO MATCH NUMBER OF DECIMAL POINTS
      IF (NPT .LT. 0) NPT=0
C IN OTHER VERSIONS THAT USE A CALL OF NUMBER NPT=-1 IMPLIES NO
C DECIMAL POINT IN THE NUMBER
      ENCODE(BUFFR,100) NPT
  100 FORMAT('(F15.',I1,')')
      DECODE(BUFFR,101) FMT
  101 FORMAT(2A4)
C CAN DYNAMICALLY SET NO. OF DECIMAL PLACES
      ENCODE(BUFFR,FMT) VALUE
      DECODE(BUFFR,2) (STRING(K),K=1,15)
    2 FORMAT(80A1)
C      EDIT FURTHER
      DO 5 I=1,15
      IF(STRING(I) .NE. BLAH) GOTO 15
    5 CONTINUE
   15 J=1
      IF(I .EQ. 1) GOTO 20
      DO 25 K=J,15
      M=K+I-J
      STRING(K)=STRING(M)
   25 CONTINUE
   20 NSTRIN=15-I+1
C     ALSO REMOVE TRAILING ZEROS AFTER DECIMAL POINT...
      K=NSTRIN
      DO 50 I=1,NSTRIN
      IF(STRING(K) .NE. ZERO) GOTO 60
```

```
      K=K-1
   50 CONTINUE
   60 NSTRIN=K
      IF(STRING(NSTRIN) .EQ.  POINT) NSTRIN=NSTRIN-1
      RETURN
      END
```