UR1982-13

1982/13. A FORTRAN program for plotting straight-line cross-sections

R.G. Richardson

Abstract

This program plots arbitrarily oriented straight-line cross-sections through areally distributed data on a rectangular (metric) grid. Both the horizontal and vertical section scales are specified at run time.

THE PROGRAM

XSECTN (Appendix 1)

This program was written for plotting cross-sections of gravity data, but any data in a suitable format can be plotted. The program reads the section parameters from logical unit 5. The data may be read in raw form from logical unit 4 or in gridded form from logical unit 7.

The program reads the start and finish co-ordinates of the section and the maximum perpendicular distance from the section line for which data points will be accepted. A new co-ordinate system is set up with the section line as the X-axis. The data points are transformed to the new co-ordinate system and points having an X co-ordinate along the section line and a sufficiently small Y co-ordinate are accepted. The section labelling is chosen to suit the plotting scale.

Non-standard subroutines are:

FREEIN (ANUM, IVAL, IER, ILUN, ICC) - reads IVAL real numbers in free format from logical unit ILUN and stores them in ANUM. If there is an error, IER is TRUE and ICC is the actual number of values read.

SYSCOM (I, 'AS 0899*') - assign logical unit 8 to the core buffer.

INITAL (ILUN, NSTEP, IW, I, I, I) - initialise the plotter as logical unit ILUN with NSTEP steps per inch and a width of IW inches.

PLOT (X, Y, I) - X and Y are the co-ordinates to which the pen is to be moved (in inches) or the co-ordinate value of the present position, depending on I.

I = 0 No pen movement. The present pen position is redefined as (X, Y)
I = 1 The pen moves to (X, Y) in its present condition
I = 2 The pen drops and moves to (X, Y)
I = 3 The pen lifts and moves to (X, Y)
I = -1, -2, or -3. As for the positive values but after (X, Y) is reached that point is redefined as (0., 0.).

SYMBOL (X, Y, HT, IHOL, THETA, N) - writes an alphanumeric string of N characters in IHOL starting at (X, Y). The character height is HT and they are drawn at an angle THETA counter-clockwise from the X-axis.

NUMBER (X, Y, HT, FLT, THETA, N) - draws the numerals representing the value of FLT by a series of calls to SYMBOL. N specifies the number of decimal places. If N = -1 the decimal point is omitted. If

HT<0.0 the numeric string terminates at (X, Y).

PENUP, PENDN - raise, lower the pen.

RSTR(I) - end the current plot.

I = 1 move the pen beyond the plot and wait for more commands
I = 2 move the pen beyond the plot and cease waiting for commands

MARKER(I) - draw a special symbol centred at the current pen position.

WHERE (X, Y, FAC) - returns the current pen position in X, Y and the current plotting scale factor in FAC.

Control data read from logical unit 5 is:

X_1, Y_1 - the starting co-ordinates of the section line (km)

X_2, Y_2 - the finishing co-ordinates of the section line (km)

DIST - the maximum acceptable perpendicular distance from the section line (km)

AOK - R if raw data, G if gridded data

BAMIN - the minimum value for the vertical axis (in data units)

BAMAX - the maximum value for the vertical axis (in data units)

AINC - the vertical axis increment per cm (in data units)

NPLC - the number of decimal places to be used when labelling the vertical axis

SCALE - the horizontal scale, *e.g.* 50 000
The labelling used depends on the value of SCALE

(a) SCALE >100 000
label each multiple of 10 km

(b) 25 000 <SCALE ≤100 000
label each 1 km

(c) 5 000 <SCALE ≤25 000
label each 250 m

(d) 2 000 <SCALE ≤5 000
label each 100 m

(e) SCALE ≤2 000
label each 100 m and mark each 10 m.

AOK - Y if more plots to follow, N otherwise.

Data input from logical unit 4 is:

X, Y, BA - format (10X, 2F9.1, 36X, F8.2)
X,Y - the east and north co-ordinates of the data point (in metres) (X, Y in the range 0.0 to 700 000.)
BA - the data value

Data input from logical unit 7 is in machine dependent binary format and is:

XMIN - the western boundary of the gridded area (km)

XMAX - the eastern boundary of the area (km)

YMIN - the southern boundary of the area (km)

YMAX - the northern boundary of the area (km)

SPACE - the X and Y spacing of the interpolated points (km)

(TTLE(I), I = 1, 18) - 72 bytes of data to be ignored

(NXNY(I), I = 1, 2) - the number of interpolated points along the southern boundary (NX) and the number of interpolated points along the western boundary (NY) respectively.

NY sets of:

> (NXNY(I), I = 1, 2) - 4 bytes to be ignored
> NX interpolated values. If the interpolated value is less than -9999.0 the gridding routine has output a dummy value.

[6 May 1982]

4/15

APPENDIX 1

Program XSECTN

```
;TITL  XSECTN - PLOT STRAIGHT LINE CROSS-SECTIONS
: FOR PLOTTING CROSS-SECTIONS IN ARBITRARY DIRCETIONS
: THE SECTIONS ARE SPECIFIED BY THE START AND END COORDINATES
: CONTORL DATA READ FROM LU 5
: DATA VALUES READ FROM LU  4 AS RAW DATA
: DATA VALUES READ FROM LU  7 AS GRIDDED DATA
      REAL X(1000),ANUM(2),TTLE(18),VAR(1000)
      INTEGER*2 NXNY(2),AOK,AR,AG,YES,NO,INDEX(1000)
      IMPLICIT INTEGER*2 (I-N)
      DOUBLE PRECISION DY,TXSP
      LOGICAL IER,LABEL,METRES,NORTH,EAST,RAW,LINE
      DATA YES/1HY/,AR/1HR/,AG/1HG/,NO/1HN/
      YMXPLT=0.
: WIDTH OF PLOTTER USED
      CALL SYSCOM(I,'AS 0899*')
: ASSIGN THE CORE BUFFER
   10 WRITE(5,100)
  100 FORMAT(' COORDS.  OF START(KM)'/' EASTTTTTT NORTHHHHH')
: READ CROSS SECTION STARTING POINT
      CALL FREEIN(ANUM,2,IER,5,ICC)
      IF (IER) GOTO 10
      X1=ANUM(1)
      Y1=ANUM(2)
   11 WRITE(5,101)
  101 FORMAT(' COORDS.  OF FINISH(KM)'/' EASTTTTTT NORTHHHHH')
: READ CROSS SECTION ENDING POINT
      CALL FREEIN(ANUM,2,IER,5,ICC)
      IF (IER) GOTO 11
      X2=ANUM(1)
      Y2=ANUM(2)
:
      NORTH=X1.EQ.X2
      IF (.NOT.  NORTH .OR.  (NORTH .AND.  Y2 .GT.  Y1)) GOTO 8
      TEMP=Y1
      Y1=Y2
      Y2=TEMP
: MUST BE S-N SECTION
    8 EAST=Y1 .EQ.  Y2
      IF (.NOT.  EAST .OR.  (EAST .AND.  X2 .GT.  X1)) GOTO 12
      TEMP=X1
      X1=X2
      X2=TEMP
: MUST BE W-E SECTION
   12 WRITE(5,102)
  102 FORMAT(' MAXIMUM DISTANCE FROM SECTION LINE (KM)?')
      CALL FREEIN(ANUM,1,IER,5,ICC)
      IF (IER) GOTO 12
      DIST=ANUM(1)
   14 WRITE(5,103)
  103 FORMAT(' RAW OR GRIDDED DATA?')
      READ(5,104) AOK
  104 FORMAT(A1)
      IF (AOK .NE.  AR .AND.  AOK .NE.  AG) GOTO 14
: NOT RAW OR GRIDDED
      RAW=AOK .EQ.  AR
: TRUE IF RAW DATA
      XD=X2-X1
      YD=Y2-Y1
: DIFERENCES ALONG MAP AXES
      THETA=ATAN2(YD,XD)
```

```
C WANT ANGLE TO ROTATE TO MAKE LINE BETWEEN X1,Y1 AND X2,Y2 AS NEW X-AXIS
      CTHETA=COS(THETA)
      STHETA=SIN(THETA)
C USE FOR COORD TRANSFORM
C
C XR=(X-XBASE)*CTHETA+(Y-YBASE)*STHETA+X0
C YR=(Y-YBASE)*CTHETA-(X-XBASE)*STHETA+Y0
C XBASE=X1
C YBASE=Y1
C X0,Y0=0.,0.
      XNUM=-X1*CTHETA-Y1*STHETA
      YNUM=-Y1*CTHETA+X1*STHETA
C USED TO MAKE FORMULA LATER
C XR=X*CTHETA+Y*STHETA+XNUM
C YR=Y*CTHETA-X*STHETA+YNUM
      CALL ROT8(X2DIST,X2,Y2,CTHETA,STHETA,XNUM)
C GET DISTANCE OF X2,Y2 FROM X1,Y1 ALONG NEW X-AXIS
C
      BAMAX=-9.9E20
      BAMIN=-BAMAX
C SET RANGES FOR LATER TESTS
      NPT=0
C COUNTER FOR NUMBER OF POINTS NOMINALLY ALONG SECTION LINE
      IF (RAW) GOTO 30
C DEAL WITH GRIDDED DATA FIRST
      REWIND 7
      READ(7) XMIN,XMAX,YMIN,YMAX,SPACE,(TTLE(I),I=1,18),NXNY
C READ RANGES, SPACING, NX, NY FROM FILE
      NX=NXNY(1)
      NY=NXNY(2)
      YY=YMIN
      DO 15 I=1,NY
      XX=XMIN
      READ(7) NXNY
C SKIP LINE NUMBER
      DO 16 J=1,NX
      READ(7) BA
      IF (BA .LT. -9999.) GOTO 16
C DUMMY VALUE
      CALL ROT8(XNEW,XX,YY,CTHETA,STHETA,XNUM)
C GET NEW X COORD
      IF (XNEW .LT. 0. .OR. XNEW .GT. X2DIST) GOTO 16
C NOT BETWEEN THE TWO ENDS OF THE SECTION LINE
      CALL ROT8(YNEW,YY,-XX,CTHETA,STHETA,YNUM)
C GET NEW Y COORD - I.E. PERPENDICULAR DIST. FROM SECTION LINE
      IF (ABS(YNEW) .GT. DIST) GOTO 16
C TOO FAR AWAY
      NPT=NPT+1
      IF (NPT .EQ. 1001) STOP 9998
      X(NPT)=XNEW
      BAMAX=AMAX1(BAMAX,BA)
      BAMIN=AMIN1(BAMIN,BA)
      VAR(NPT)=BA
C STORE AWAY
   16 XX=XX+SPACE
C INCREMENT X
   15 YY=YY+SPACE
C INCREMENT Y
C
      GOTO 18
```

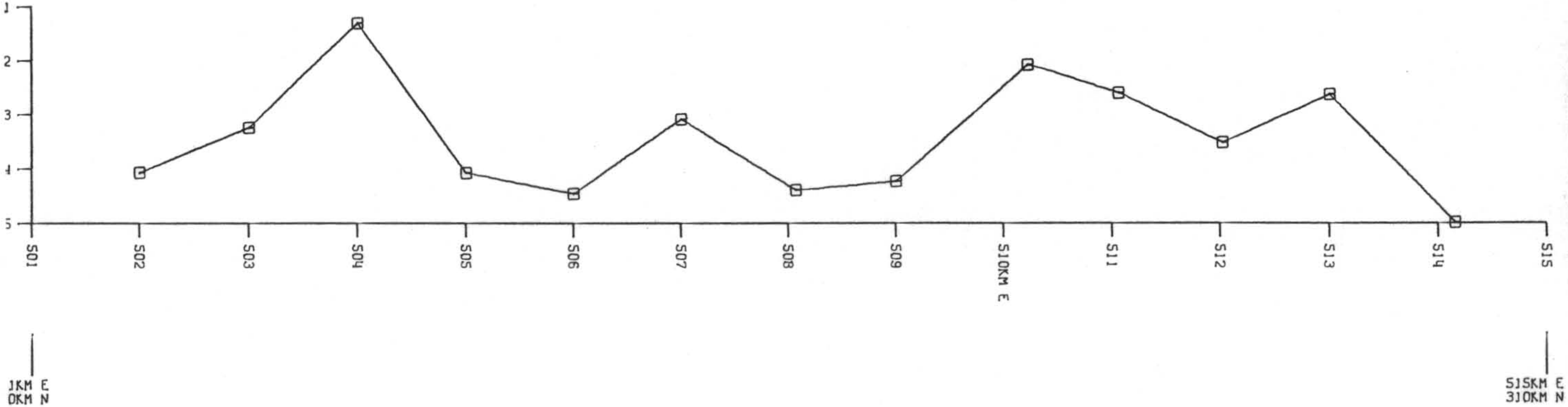
```
C CHECKED GRIDDED DATA
C
   30 CONTINUE
C NOW FOR RAW DATA
      REWIND 4
   17 READ(4,200,END=18) XX,YY,BA
  200 FORMAT(10X,2F9.1,36X,F8.2)
      XX=XX*0.001
      YY=YY*0.001
C TO KM
      CALL ROT8(XNEW,XX,YY,CTHETA,STHETA,XNUM)
C NEW X COORD
      IF (XNEW .LT. 0.0 .OR. XNEW .GT. X2DIST) GOTO 17
C NOT BETWEEN THE ENDS
      CALL ROT8(YNEW,YY,-XX,CTHETA,STHETA,YNUM)
C GET NEW Y COORD
      IF (ABS(YNEW) .GT. DIST) GOTO 17
C TOO FAR AWAY
      NPT=NPT+1
      IF (NPT .EQ. 1001) STOP 9998
      X(NPT)=XNEW
      BAMAX=AMAX1(BAMAX,BA)
      BAMIN=AMIN1(BAMIN,BA)
      VAR(NPT)=BA
C STORE AWAY
      GOTO 17
C
C
   18 CONTINUE
C NOW HAVE ALL POINTS ALONG SECTION
      WRITE(5,201) BAMIN,BAMAX
  201 FORMAT(' RANGE IS FROM',F10.4,' TO',F10.4)
      WRITE(5,202)
  202 FORMAT(' PLOTTING RANGE?'/' MINNNNNNN MAXXXXXXX')
      CALL FREEIN(ANUM,2,IER,5,ICC)
      IF (IER) GOTO 18
      BAMIN=ANUM(1)
      BAMAX=ANUM(2)
   19 WRITE(5,203)
  203 FORMAT(' INCREMENT/CM ?'/' INCCC')
      CALL FREEIN(ANUM,1,IER,5,ICC)
      IF (IER) GOTO 19
      AINC=ANUM(1)
C INCREMENT/CM
C NOW CHECK SIZE
      IF (((BAMAX-BAMIN)/AINC) .LE. 30.) GOTO 20
C LESS THAN 30 CM HIGH
      WRITE(5,205)
  205 FORMAT(' VERTICAL SCALE TOO LARGE')
      GOTO 19
   20 WRITE(5,204)
  204 FORMAT(' HOW MANY DECIMAL PLACES?'/' I')
      CALL FREEIN(ANUM,1,IER,5,ICC)
      IF (IER .OR. ANUM(1) .GT. 6.) GOTO 20
      NPLC=ANUM(1)
      IF @C .LE. 0) NPLC=-1
C NO DECIMAL PLACES AFTER DECIMAL POINT
   21 WRITE(5,206)
  206 FORMAT(' SCALE? (E.G. 50000.)')
      CALL FREEIN(ANUM,1,IER,5,ICC)
      IF (IER) GOTO 21
      SCALE=ANUM(1)
      FACT=1. E+5/(2. 54*SCALE)
C THE FACTOR TO CONVERT 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=0. 1
      SPCE=100. 0
      TXSP=10.
      GOTO 855
C
C SCALE . LE.  100000, SCALE . GT.  25000
  854 METRES=. FALSE.
      DFACT=1. 0
      TXSP=1.
      SPCE=10.
      GOTO 855
C
C SCALE . LE.  25000, SCALE . GT.  5000
  853 DFACT=4.
      TXSP=0. 25
      SPCE=1. 0
      GOTO 855
C
C SCALE . LE.  5000, SCALE . GT.  2000
  852 DFACT=10.
      TXSP=0. 1
      SPCE=0. 5
      GOTO 855
C
C SCALE . LE.  2000
  851 DFACT=10.
      LABEL=. FALSE.
      TXSP=0. 01
      SPCE=0. 1
  855 CONTINUE
      XMIN=0.
      XMAX=X2DIST
C DEFAULT DISTANCES ALONG SKEWED SECTION
      IF (NORTH) XMIN=Y1
      IF (EAST) XMIN=X1
      IF (NORTH) XMAX=Y2
      IF (EAST) XMAX=X2
C CHOOSE THE RIGHT RANGE
      FXMIN=AINT(XMIN*DFACT+0. 001)/DFACT
      FXMAX=AINT(XMAX*DFACT+0. 999)/DFACT
C GET THE RANGES
      YTMP=FXMAX
      CALL XFORM(YTMP, FXMIN, FACT)
C GET THE WIDTH IN PLOTTER UNITS
      YTMP=YTMP+2.
C GET TOTAL WIDTH INCLUDING LABELS
      IF (YTMP . LE.  35. 5) GOTO 22
```

```
C WILL FIT ON PAPER
      WRITE(5,207)
  207 FORMAT(' PLOT WILL NOT FIT')
      GOTO 21
   22 CONTINUE
C WILL FIT
      BAFACT=1./(2.54*AINC)
C FACTOR TO CONVERT FROM INCREMENT/CM TO PLOTTER INCHES
      IF (YMXPLT+YTMP .LE. 35.5 .AND. YMXPLT .NE. 0.) GOTO 23
C FITS ON PAPER FOLLOWING A PRE-EXISTING PLOT
      IF (YMXPLT .GT. 0.) CALL RSTR(1)
C INDEX TO NEXT PLOT
      IF (YMXPLT .EQ. 0.) CALL INITAL(9,200,3,1,0,0)
C THE FIRST PLOT
      CALL PLOT(0.5,2.,-3)
C MOVE TO 0.5,2. AND CALL IT 0.0,0.0
      YMXPLT=YTMP
      YMXLST=YTMP
      GOTO 24
C
C NOW FOR A PLOT THAT FITS
   23 CALL PLOT(0.0,YMXLST,-3)
C MOVE TO ABOVE LAST PLOT AND CALL IT 0.0,0.0
      YMXPLT=YMXPLT+YTMP
      YMXLST=YTMP
C UPDATE YMXPLT
   24 CONTINUE
C NOW FOR THE BA AXIS
      DX=BAMAX+AINC
   25 DX=DX-AINC
      XTMP=DX
      AX=XTMP
      CALL XFORM(XTMP,BAMAX,-BAFACT)
      CALL PLOT(XTMP,0.,2)
C MOVE TO START OF PIP
      CALL NUMBER(XTMP+0.035,-0.15,-0.07,AX,90.,NPLC)
C LABEL IT
      CALL PLOT(XTMP,-0.1,1)
      CALL PLOT(XTMP,0.,2)
C DRAW PIP
      IF (DX-BAMIN .GT. AINC*0.1) GOTO 25
C LABEL THE PROFILE AXIS
C
C NOW FOR THE COORDINATE AXIS
      DY=DBLE(FXMIN)-TXSP
   26 DY=DY+TXSP
      YTMP=DY
      AY=YTMP
      CALL XFORM(YTMP,FXMIN,FACT)
      CALL PLOT(XTMP,YTMP,2)
      LINE=AY-AINT(AY/SPCE+0.05)*SPCE .LE. 0.005
C CHECK TO SEE IF NEED FULL LABEL ON LINE
      CALL YLABEL(AY,XTMP,YTMP,LABEL,LINE,METRES,EAST,NORTH)
      IF (DBLE(FXMAX)-DY .GT. 0.005D00) GOTO 26
C GONE ALONG AXIS
      YTMP=XMAX
      CALL XFORM(YTMP,FXMIN,FACT)
      CALL PLOT(XTMP+0.8,YTMP,3)
      CALL PLOT(XTMP+1.1,YTMP,2)
C DRAW A LINE AT THE END OF THE SECTION
```

```
      CALL PTNBN(XTMP+1.2,YTMP,X2,METRES,2)
      CALL PTNBN(XTMP+1.3,YTMP,Y2,METRES,1)
      YTMP=XMIN
      CALL XFORM(YTMP,FXMIN,FACT)
      CALL PLOT(XTMP+0.8,YTMP,1)
      CALL PLOT(XTMP+1.1,YTMP,2)
C DRAW LINE AT START OF SECTION
      CALL PTNBN(XTMP+1.2,YTMP,X1,METRES,2)
      CALL PTNBN(XTMP+1.3,YTMP,Y1,METRES,1)
C LABELLED BOTH ENDS OF SECTION
C
C NOW SORT DISTANCES FROM X1,Y1
      DO 27 I=1,NPT
      INDEX(I)=I
   27 CONTINUE
      CALL BUBSTR(INDEX,X,1,NPT)
C
C PLOT POINTS
      J=3
      DO 28 I=1,NPT
      INDX=INDEX(I)
      YTMP=X(INDX)
      XTMP=VAR(INDX)
      YTMP=YTMP+XMIN
C OFFSET IF NEEDED FOR NON-ZERO BASE POINT
      CALL XFORM(YTMP,FXMIN,FACT)
C DIST ALONG AXIS
      CALL XFORM(XTMP,BAMAX,-BAFACT)
      CALL PLOT(XTMP,YTMP,J)
      IF (RAW) CALL MARKER(1)
C A MARKER
      J=2
   28 CONTINUE
C PLOT POINTS AND JOIN BY A LINE
   29 WRITE(5,208)
  208 FORMAT(' MORE PLOTS?')
      READ(5,104) AOK
      IF (AOK .EQ. YES) GOTO 10
      IF (AOK .NE. NO) GOTO 29
      CALL RSTR(2)
      STOP
      END
      SUBROUTINE XFORM(X,XBASE,FACT)
      X=(X-XBASE)*FACT
      RETURN
      END
      SUBROUTINE PTNBN(X,Y,VAR,METRES,ISEL)
      INTEGER*2 ISEL,N
      LOGICAL METRES
      REAL CHAR,ALABL1(2),ALABL2(2)
      DATA ALABL1/4HKM N,4HKM E/,ALABL2/4HM N ,4HM E /
      CHAR=ALABL1(ISEL)
      AX=VAR
      N=4
      IF (.NOT. METRES) GOTO 10
      CHAR=ALABL2(ISEL)
      AX=AX*1000.
      N=3
   10 CALL NUMBER(X,Y-0.3,0.07,AX,90.0,-1)
      CALL WHERE(XX,YY,AX)
      CALL SYMBOL(XX,YY,0.07,CHAR,90.,N)
      RETURN
      END
      SUBROUTINE YLABEL(VAL,X,Y,LABEL,LINE,METRES,EAST,NORTH)
      LOGICAL LABEL,LINE,METRES,EAST,NORTH
      INTEGER*2 ISEL,N
      REAL CHAR,ALABL1(2),ALABL2(2),ALABL3(2)
      DATA ALABL1/4HKM N,4HM N /,
     . ALABL2/4HKM E,4HM E /,
     . ALABL3/4HKM  ,4HM   /
      CALL PLOT(X+0.1,Y,1)
      IF (.NOT. LABEL .AND. .NOT. LINE) GOTO 10
C ONLY WANT TICK
      AX=VAL
      IF (METRES) AX=AX*1000.
      CALL NUMBER(X+0.15,Y-0.035,0.07,AX,0.,-1)
      IF (.NOT. LINE) GOTO 10
      ISEL=1
      IF (METRES) ISEL=2
      N=4
      IF (METRES) N=3
      CHAR=ALABL3(ISEL)
      IF (EAST) CHAR=ALABL2(ISEL)
      IF (NORTH) CHAR=ALABL1(ISEL)
C GET LABELLING
      CALL WHERE(XX,YY,AX)
      CALL SYMBOL(XX,YY,0.07,CHAR,0.,N)
   10 CALL PLOT(X,Y,1)
      RETURN
      END
      SUBROUTINE BUBSTR(IR,A,IBASE,N)
      REAL A(N)
      INTEGER*2 IR(N),IBASE,N,I,J,IRI,IRIP1,NM1,IP1
      LOGICAL NSWAP
      IF (N .LE. 1) RETURN
C NOTHING TO SORT
      NM1=N-1
      DO 30 J=IBASE,NM1
      NSWAP=.TRUE.
      IRI=IR(1)
      DO 40 I=IBASE,NM1
      IP1=I+1
      IRIP1=IR(IP1)
      IF (A(IRI) .LE. A(IRIP1)) GOTO 40
      NSWAP=.FALSE.
      IR(I)=IRIP1
      IR(IP1)=IRI
      IRIP1=IRI
   40 IRI=IRIP1
      IF (NSWAP) RETURN
   30 CONTINUE
      RETURN
      END
      SUBROUTINE ROT8(XR,X,Y,CTHETA,STHETA,XNUM)
      XR=X*CTHETA+Y*STHETA+XNUM
      RETURN
      END
```

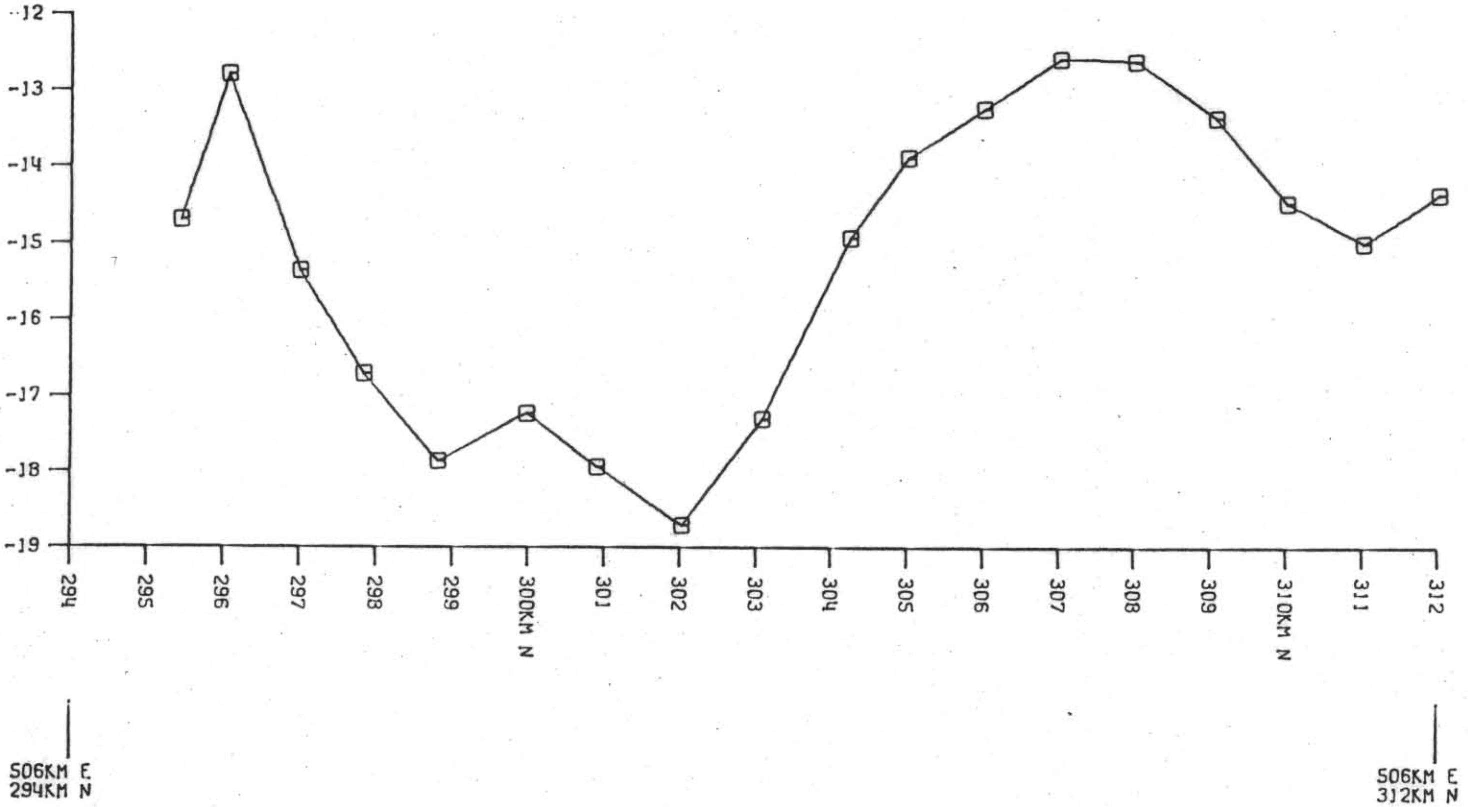
APPENDIX 2

Examples of plots



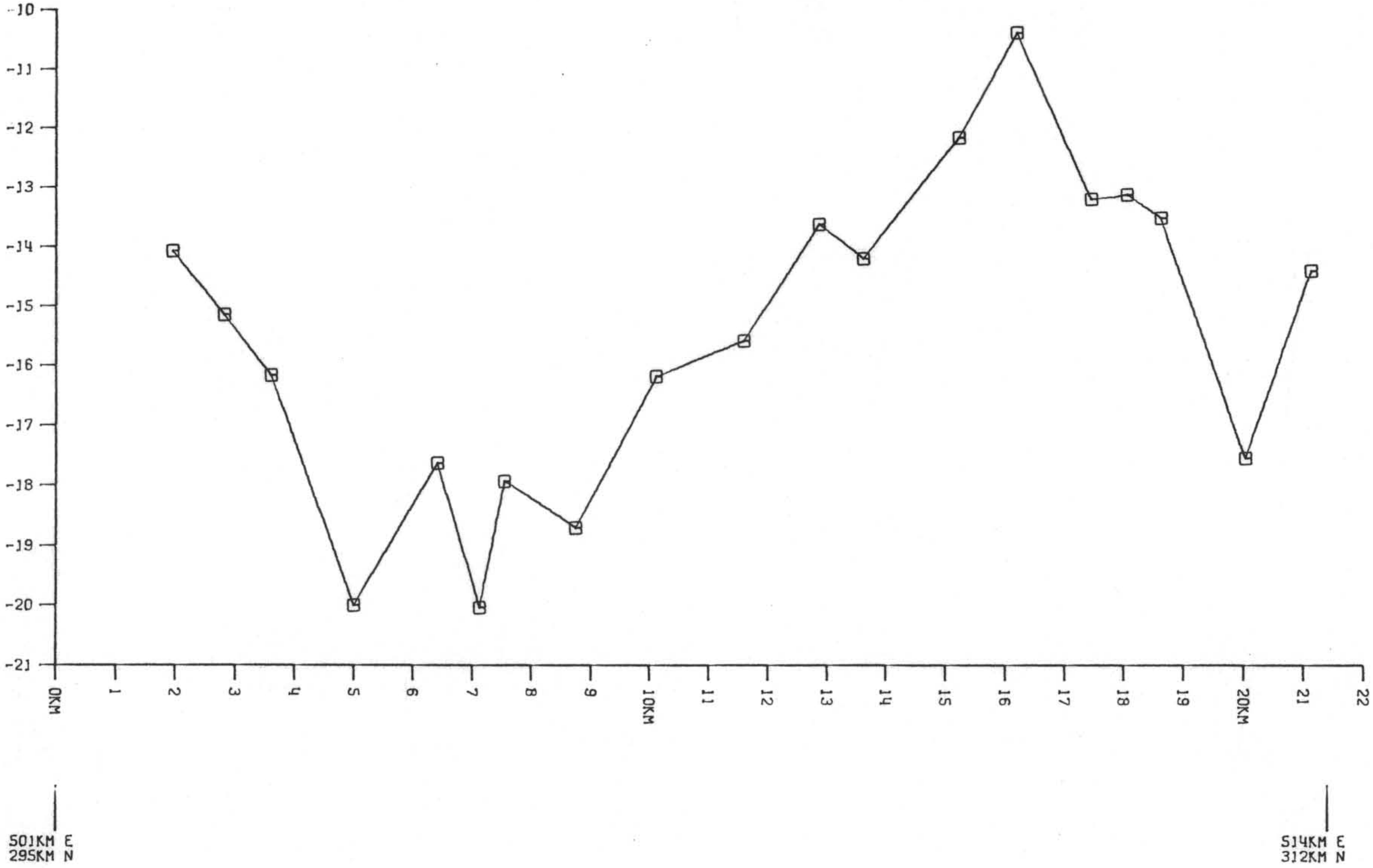
E-W section from 501000, 310000 to 515000, 310000
Scale 1:50 000

5 cm



N-S section from 506000, 294000 to 506000, 312000
Scale 1:100 000

5 cm



Diagonal section from 501000, 295000 to 514000, 312000
Scale 1:100 000

5 cm