1983/45. A FORTRAN program for plotting straight-line cross-sections (Revision 1)

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 is read from logical unit 4.

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 used for plotting are described in Richardson (1983).

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 (metres)
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 centimetre (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

REFERENCE

RICHARDSON, R.G. 1983. Hard copy plotting on the Geological Survey mini-computer. *Unpubl.Rep.Dep.Mines Tasm.* 1983/38.

[12 September 1983]

APPENDIX 2

Program XSECTN

```
$TITL  XSECTN - PLOT STRAIGHT LINE CROSS-SECTIONS
C FOR PLOTTING CROSS-SECTIONS IN ARBITRARY DIRCETIONS
C THE SECTIONS ARE SPECIFIED BY THE START AND END COORDINATES
C CONTROL DATA READ FROM LU 5
C DATA VALUES READ FROM LU  4 AS RAW DATA
      REAL X(1000),VAR(1000)
      INTEGER*2 AOK,YES,NO,INDEX(1000),ASE,ASC,IOPT
      INTEGER*4 DY,TXSP,SPCE,DFACT
      DOUBLE PRECISION XX1,XX2,YY1,YY2
      LOGICAL IER,LABEL,METRES,NORTH,EAST,LINE
      DATA YES/1HY/,NO/1HN/,ASC/2HSC/,ASE/2HSE/
      OPEN(UNIT=5,FILE='CON:')
C OPEN THE CONSOLE
      YMXPLT=0.
C WIDTH OF PLOTTER USED
   10 WRITE(5,100)
  100 FORMAT(' COORDS. OF START(KM)'/' EASTTTTTT NORTHHHHH')
C READ CROSS SECTION STARTING POINT
      READ(5,*) XX1,YY1
      X1=XX1
      IXX1=XX1*1000.0D00
      Y1=YY1
      IYY1=YY1*1000.0D00
      WRITE(5,101)
  101 FORMAT(' COORDS. OF FINISH(KM)'/' EASTTTTTT NORTHHHHH')
C READ CROSS SECTION ENDING POINT
      READ(5,*) XX2,YY2
      X2=XX2
      IXX2=XX2*1000.0D00
      Y2=YY2
      IYY2=YY2*1000.0D00
C
      NORTH=X1.EQ.X2
      IF (.NOT. NORTH .OR. (NORTH .AND. Y2 .GT. Y1)) GOTO 8
      TEMP=Y1
      Y1=Y2
      Y2=TEMP
C 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
C MUST BE W-E SECTION
   12 WRITE(5,102)
  102 FORMAT(' MAXIMUM DISTANCE FROM SECTION LINE (METRES)?')
      READ(5,*) DIST
      DIST=DIST*0.001
C CONVERT TO KM.
      XD=X2-X1
      YD=Y2-Y1
C 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
      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')
      READ(5,*) BAMINN,BAMAXX
      IF (BAMAXX .LE. BAMINN .OR.  BAMIN .LT. BAMINN
     .   .OR. BAMAX .GT. BAMAXX) GOTO 18
      BAMIN=BAMINN
      BAMAX=BAMAXX
C MUST HAVE THE BIGGEST FIRST
   19 WRITE(5,203)
  203 FORMAT(' INCREMENT/CM ?'/' INCCC')
      READ(5,*) AINC
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')
      READ(5,*) NPLC
      IF (NPLC .GT. 6) GOTO 20
      IF (NPLC .LE. 0) NPLC=-1
C NO DECIMAL PLACES AFTER DECIMAL POINT
   21 WRITE(5,206)
  206 FORMAT(' SCALE? (E.G. 50000.)')
      READ(5,*) SCALE
      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=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
      IXMIN=0.
      IXMAX=X2DIST*1000.0
C DEFAULT DISTANCES ALONG SKEWED SECTION
      IF (NORTH) IXMIN=IYY1
      IF (EAST) IXMIN=IXX1
      IF (NORTH) IXMAX=IYY2
```

```
      IF (EAST) IXMAX=IXX2
C CHOOSE THE RIGHT RANGE
      XMIN=IXMIN/1000.0
      XMAX=IXMAX/1000.0
      IFXMIN=((IXMIN+1)/DFACT)*DFACT
      IFXMAX=((IXMAX+DFACT-1)/DFACT)*DFACT
      FXMIN=IFXMIN/1000.0
      FXMAX=IFXMAX/1000.0
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. 35.0) GOTO 22
C WILL FIT ON PAPER
      WRITE(5,207)
  207 FORMAT(' PLOT WILL NOT FIT'/' DO YOU WISH TO CHANGE SCALE'
     . ' (SC) OR SEARCH AGAIN (SE)?')
   31 READ(5,209) IOPT
  209 FORMAT(A2)
      IF (IOPT .EQ. ASE) GOTO 10
      IF (IOPT .EQ. ASC) GOTO 21
      GOTO 31
      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. 35.0 .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=IFXMIN-TXSP
   26 DY=DY+TXSP
      YTMP=DY*0.001
      CALL XFORM(YTMP,FXMIN,FACT)
      CALL PLOT(XTMP,YTMP,2)
      LINE=MOD(DY,SPCE) .EQ. 0
C CHECK TO SEE IF NEED FULL LABEL ON LINE
      CALL YLABEL(DY,XTMP,YTMP,LABEL,LINE,METRES,EAST,NORTH)
      IF (IFXMAX .GT. DY) 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,IXX2,METRES,2)
      CALL PTNBN(XTMP+1.3,YTMP,IYY2,METRES,1)
      YTMP=XMIN
      CALL XFORN(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,IXX1,METRES,2)
      CALL PTNBN(XTMP+1.3,YTMP,IYY1,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)
      CALL MARKER(1)
C A MARKER
      J=2
   28 CONTINUE
C PLOT POINTS AND JOIN BY A LINE
   29 WRITE(5,200)
  200 FORMAT(' MORE PLOTS?')
      READ(5,104) AOK
  104 FORMAT(A1)
      IF (AOK .EQ. YES) GOTO 10
      IF (AOK .NE. NO) GOTO 29
```

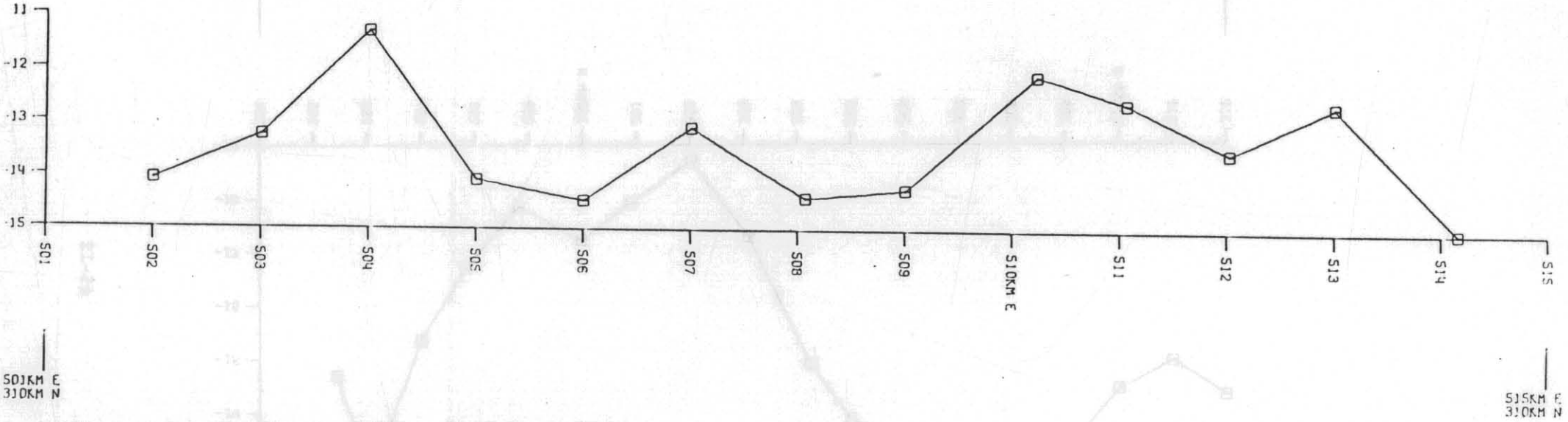
```
      CALL RSTR(2)
      STOP
      END
      SUBROUTINE XFORM(X,XBASE,FACT)
      X=(X-XBASE)*FACT
      RETURN
      END
      SUBROUTINE PTNBN(X,Y,IVAR,METRES,ISEL)
      LOGICAL METRES
      REAL CHAR,ALABL1(2),ALABL2(2)
      DATA ALABL1/4HKM N,4HKM E/,ALABL2/4HM N ,4HM E /
      CHAR=ALABL2(ISEL)
      IX=IVAR
      N=3
      IF (METRES) GOTO 10
      CHAR=ALABL1(ISEL)
      IX=IX/1000
      N=4
   10 CALL NUMPRT(X,Y-0.3,IX,90.0)
      CALL WHERE(XX,YY,AX)
      CALL MOVREL
      CALL SYMBOL(XX,YY,0.07,CHAR,90.,N)
      CALL MOVABS
      RETURN
      END
      SUBROUTINE YLABEL(IVAL,X,Y,LABEL,LINE,METRES,EAST,NORTH)
      LOGICAL LABEL,LINE,METRES,EAST,NORTH
      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
      IX=IVAL
      IF (.NOT. METRES) IX=IX/1000
      CALL NUMPRT(X+0.15,Y-0.035,IX,0.)
      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 MOVREL
      CALL SYMBOL(XX,YY,0.07,CHAR,0.,N)
      CALL MOVABS
   10 CALL PLOT(X,Y,1)
      RETURN
      END
      SUBROUTINE BUBSTR(IR,A,IBASE,N)
      REAL A(N)
      INTEGER*2 IR(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
      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
```

10/13

APPENDIX 2

Examples of plots

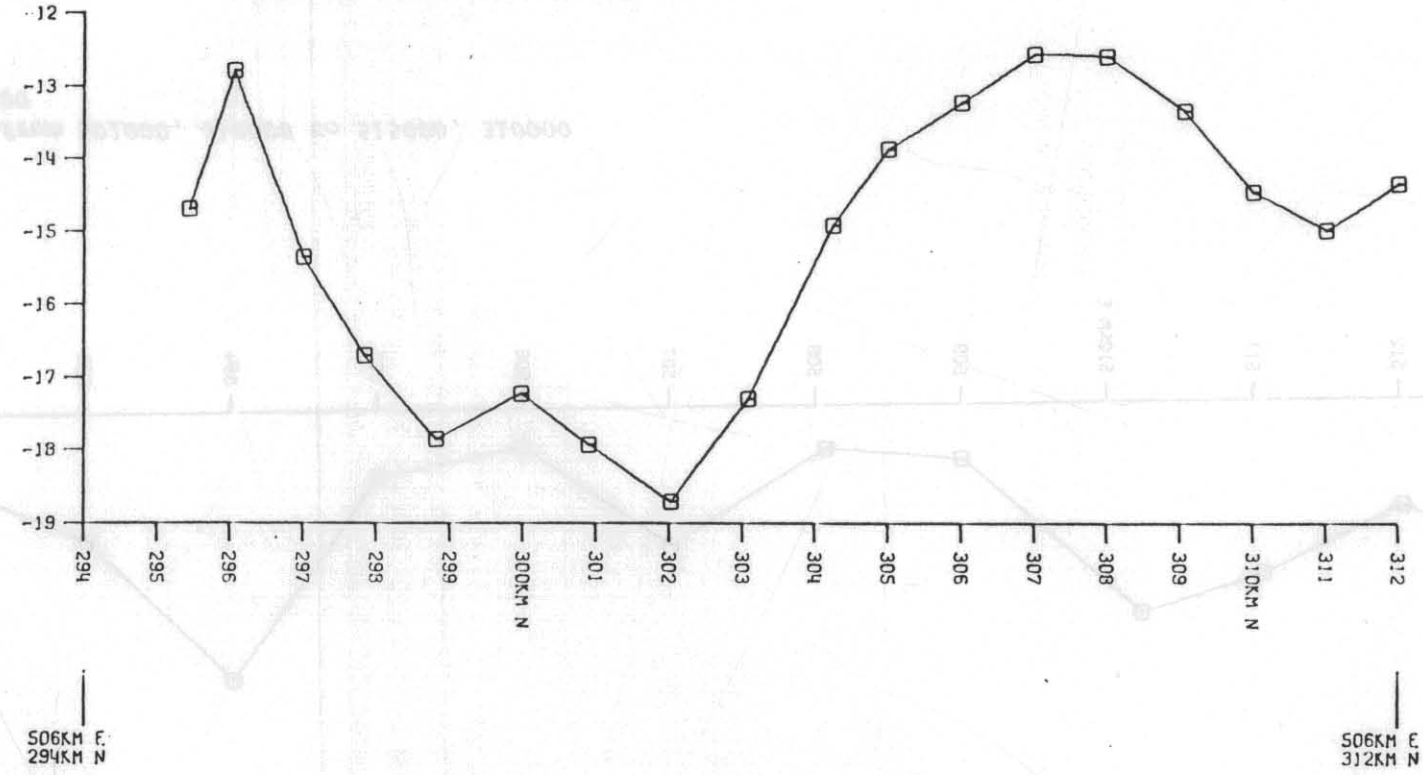


E-W section from 501000, 310000 to 515000, 310000
Scale 1:50 000

5 cm

45-11

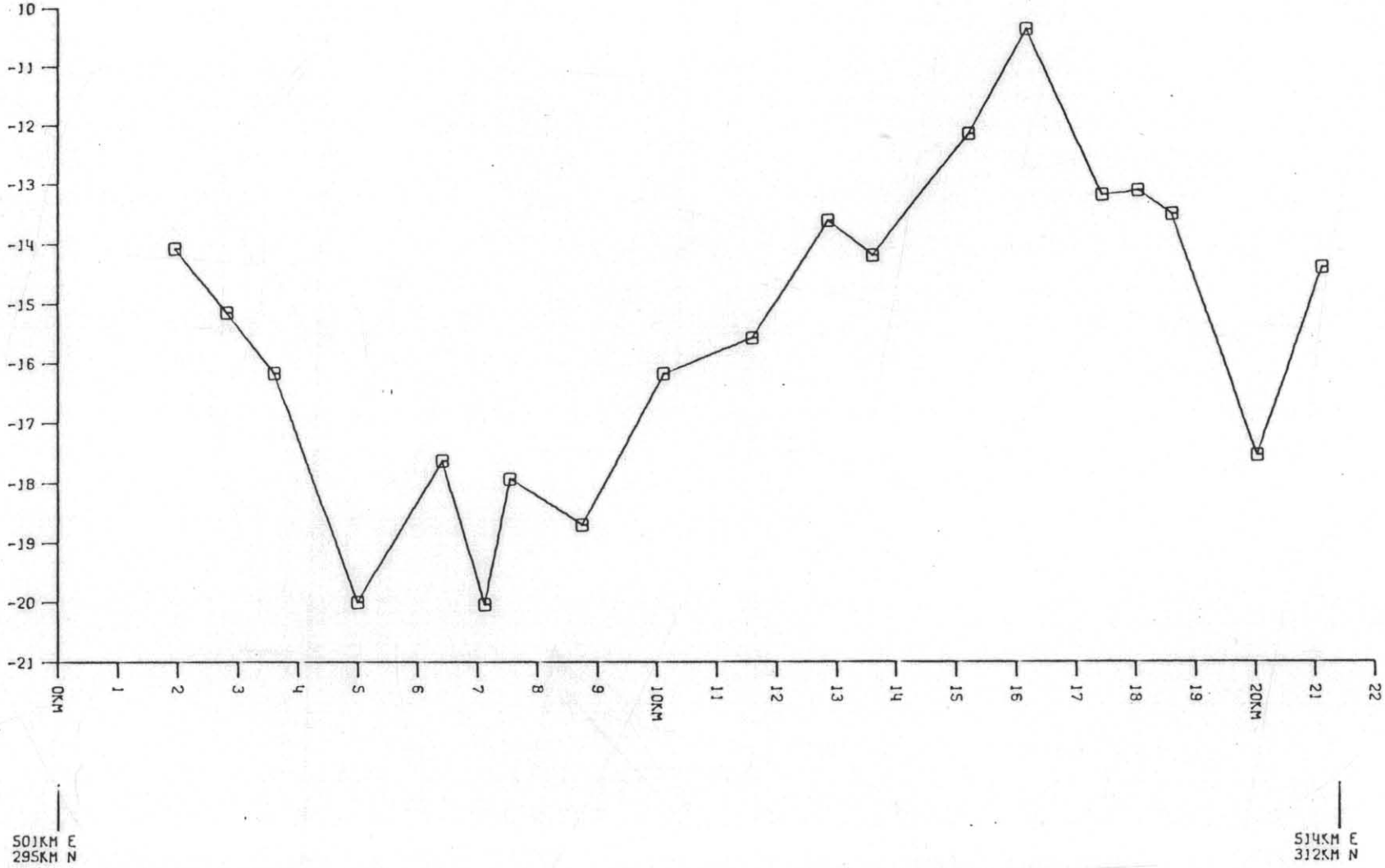
11/13



45-12

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

12/13



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

5 cm