Centrum för solenergiforskning
Solar Energy Research Center

Solar Position diagram

Eva Lindberg

SERC

Avdelningen för energi.
miljö och byggande
Högskolan Dalarna
781 88 Borlänge

Tel: +46 23 778700
Tel: +46 23 778701

Besöksadress/Street adress:
Forskargatan 8
Borlänge

HÖGSKOLAN
Dalarna

ISSN 1103 - 1816
ISRN HFB-SERC--34--SE
Maj 1990

SOLPOSITIONSDIAGRAM

Datorprogram för uppritande av solens vandring över himlavalvet vid olika tidpunkter och platser

SOLAR POSITION DIAGRAM

Computer program for drawing the solar movement over the hemisphere at different dates and places

Eva Lindberg

ISSN 1103-1816
ISRN HFB-SERC--34--SE
Maj 1990

PROGRAM SOLPOSITIONSDIAGRAM

Program SOLVEJ är ett användarvänligt program som visar solens vandring över himlavalvet vid upp till fem valfria datum och vid valfri ort. Programmet är utvecklat av två skäl. För det första, att demonstreras för en intresserad allmänhet som del av vandringsutställning om solenergi, vilken är initierad och utarbetad av SERC. För det andra, att användas av solenergiintressenter för att snabbt få en uppfattning om solinstrålningen på en ort vid olika tidpunkter på året.

Indata till programmet ges från tangentbordet. Som svar på frågor skrivs för vilken ort diagrammet skall gälla, max fem datum, ortens latitud och longitud, som anges positiv i västlig riktning, samt tidszonen. Varje uppgift avslutas med tryck på tangenten ENTER. Programmet kommer nu att rita ett koordinatsystem på skärmen. Första axeln visar vädersträcken, norr, öster, söder, väster och norr, varje delstreck utgör 10 grader. För södra halvklotet byter norr och söder plats. Andra axeln visar höjden över horisonten i grader, 0 till 90 grader och 10 grader för varje delstreck. Efter några sekunder ritas diagrammet upp med solhöjden som funktion av väderstrecket och varje hel timme markerad. Se fig. 1-4. Slutligen frågas efter om diagrammet skall ritas ut på printer. SOLVEJ avbrytes med att trycka CTRL+BREAK.

SOLVEJ är skrivet i Quick-BASIC (se App. 1) och leveras både som källkod och körklar version. Lämplig dator är IBM-kompatibel AT med EGA- eller VGA-skärmkort (ej Herkules) Lämplig printer är IBM Proprinter eller liknande matrisskrivare, kopplad till LPT1 på kommunikationskortet.

Till grund för beräkningarna har använts artikeln On Calculating the Position of the Sun, publicerad i nr. 1 1988 av The International Journal of Ambient Energy. Fem empiriska ekvationer beträffande beräkningar av solens position har studerats för att undersöka deras tillförlitlighet. Felaktigheter på fem grader eller mer kan uppträda om man använder sig av de enkla ekvationer som kan hittas solenergiböcker och som inte kräver tillgång till dator. FORTRAN-rutinen SUNAE2 (se App. 2) beräknar solpositionen med noggrannast kända metod. Program SOLVEJ är en utveckling av SUNAE2.

SOLAR POSITION DIAGRAM

The computer program SOLVEJ is a user friendly routine, which shows the solar movements over the sky during up to five optional dates and at optional places. The program is developed by two reasons. Firstly, to be demonstrated for an interested public as part of a travelling solar energy exhibition, which is initiated and produced by SERC. Secondly, to be used by people interested in solar energy in order to quickly get an idea about the insolation at a certain place at different yearly times.

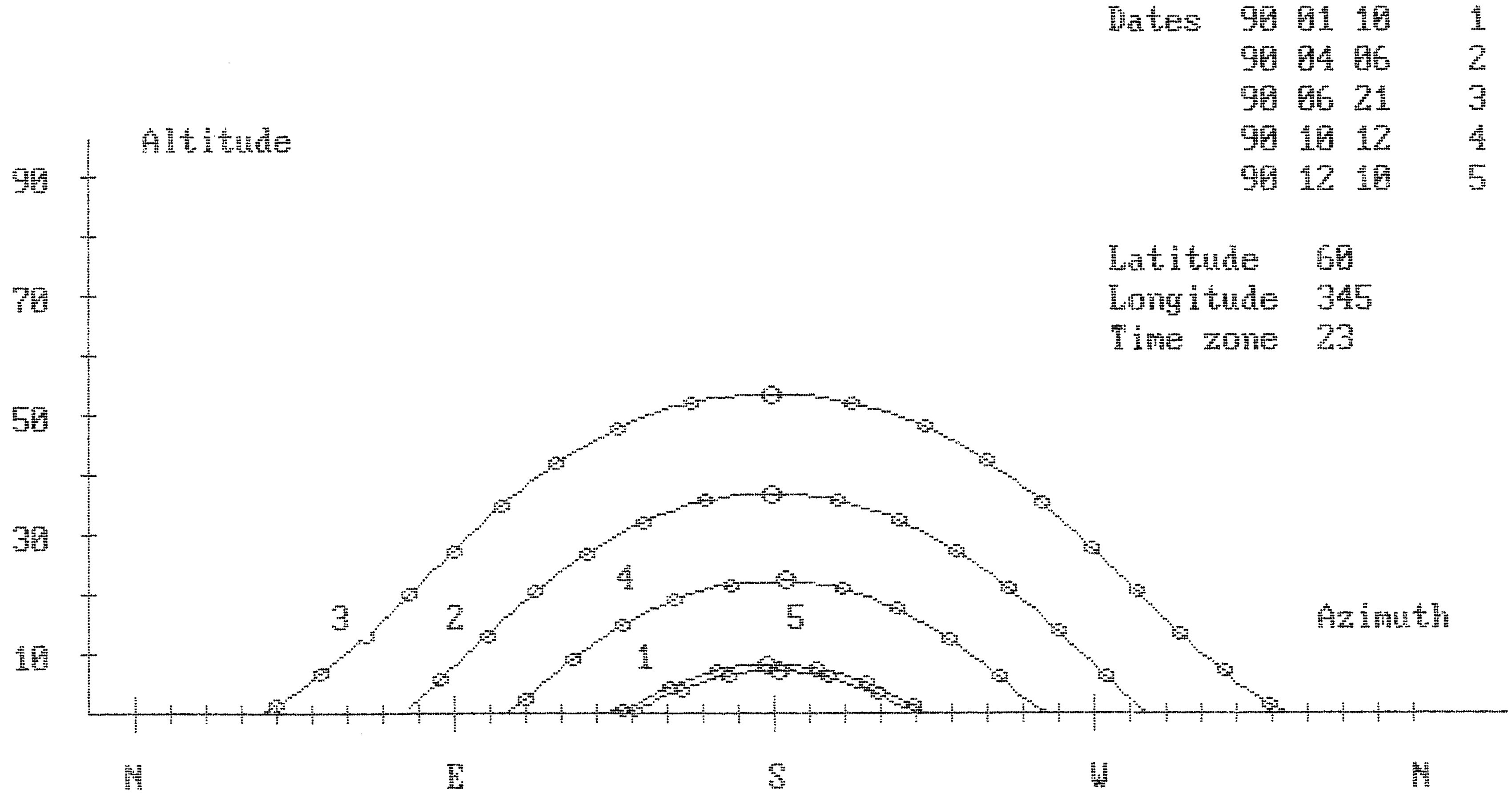
The input parameters to the program are given from the keyboard as answers on questions. The parameters are the actual place, max. five dates, latitude, longitude in west direction (pos) and at last the actual place time zoon. Every input are terminated by pressing the ENTER button. The program will now draw a coordinate system on the screen. The first axis shows the points of the compass; north, east, south, west and north, every mark makes ten degrees. At the southern half of the globe north and south are changed. The second axis shows the solar altitude for every ten degrees. After some seconds the diagram will be written with the sun's altitude as function of the compass point and with every hour marked. The figures 1-4 show examples of diagrams. Then the question is asked whether the diagram shall be printed out or not. SOLVEJ will be interrupted by pressing the buttons CTRL+BREAK.

The program is written in Quick-BASIC (App. 1) and will be delivered in both source code and executable version. A suitable computer is IBM compatible AT with EGA or VGA expansion cards (not Hercules). The printer must be an IBM Proprinter or a similar matrix printer, connected to LPT1 on the communication card.

The article On Calculating the Position of the Sun, published in No. 1 1988 of The International Journal of Ambient Energy, has been used as a base for the calculations. Five different empirical equations have been studied in order to determine their accuracy. It was found that errors of at least five degrees in solar altitude may occur when using the standard "text book procedures", which can be found in solar energy literature and not require use of a computer. The FORTRAN subroutine SUNAE2 (App. 2) calculates the solar position with the most accurate method which is known. The program SOLVEJ is a development of SUNAE2.

Fig. 1

This diagram shows the local solar altitude and azimuth in Borlänge
(in degrees)
Every hour = o
Moon = O



This diagram shows the local solar altitude and azimuth in Prague
(in degrees)

Every hour = o
Noon = O

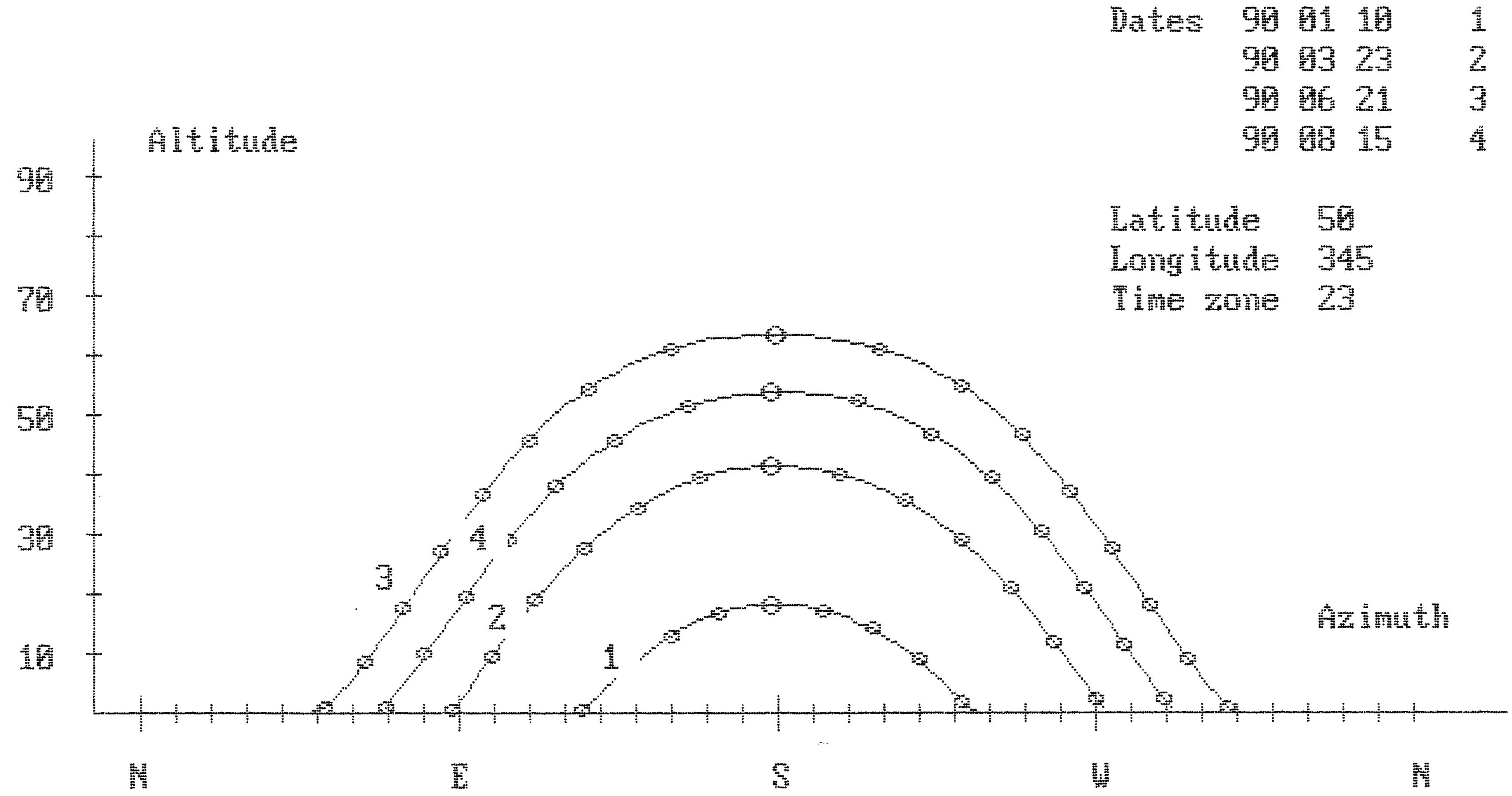


Fig. 2

Fig. 3

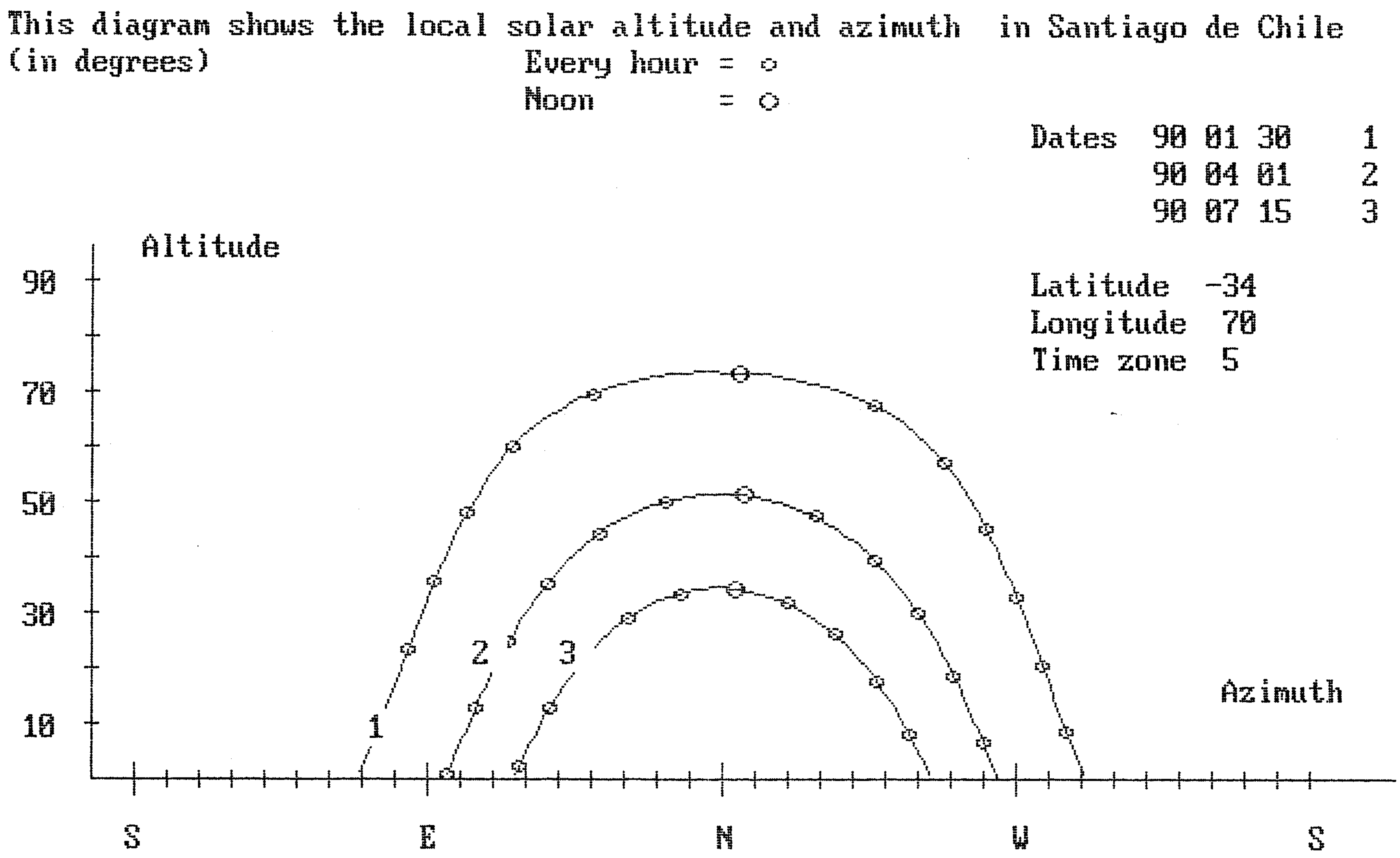
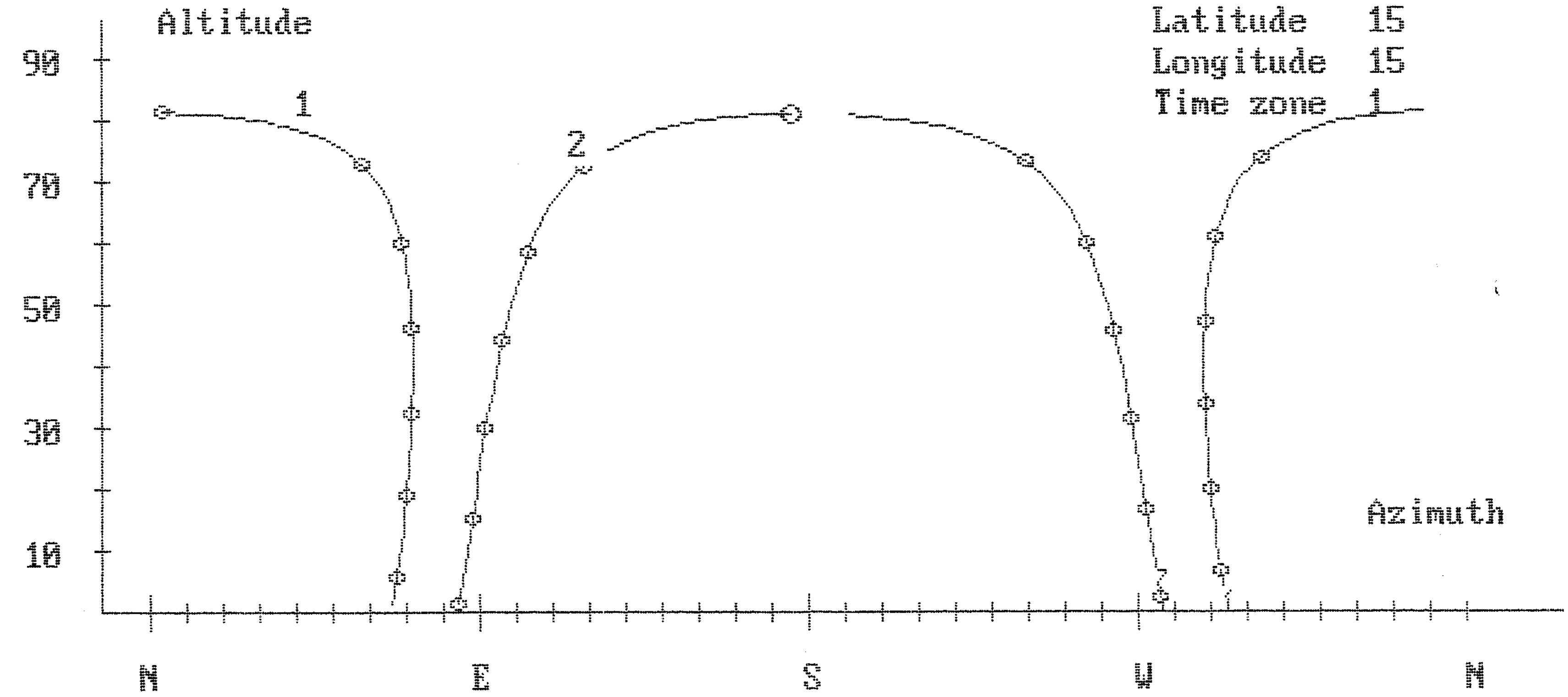

This diagram shows the local solar altitude and azimuth in Santiago de Chile
(in degrees)
Every hour = ○
Noon = ○
Dates 90 01 30 1
90 04 01 2
90 07 15 3
Latitude -34
Longitude 70
Time zone 5
Altitude
90
70
50
30
10
1
2
3
Azimuth
S
E
N
W
S

Fig. 4

This diagram shows the local solar altitude and azimuth in Dakar
(in degrees)

Every hour = ○
Noon = ○

Dates 90 06 23 1
90 04 05 2



```
DECLARE SUB DisplayPlot (k%)
DECLARE SUB DrawKoordinat (ort$, lat, lng, n, zone%)
DECLARE SUB DispCircle (k%, dotx%(), doty%(), test%())
CLS
DIM SHARED plotelevation%(300)
DIM SHARED azimuth%(300)
DIM SHARED dotx%(300)
DIM SHARED doty%(300)
DIM SHARED test%(300)
DIM SHARED day%(5)
DIM SHARED day$(5)
DIM SHARED mo(5)
DIM SHARED year(5)
DIM SHARED da$(5)
DIM SHARED da(5)
DIM SHARED mo$(5)

  PRINT "program SUN POSITION DIAGRAM"
  INPUT "What place is desired"; ort$
  PRINT "Which dates are desired (i.e. 900611 for 11 of june)
  PRINT "Press ENTER for no more dates"

  n = 1
GETMONTH:
 INPUT day$(n)
   IF day$(n) = "" THEN GOTO CALC
  IF LEN(day$(n)) <> 6 THEN GOTO GETMONTH
  mo$(n) = MID$(day$(n), 3, 2): da$(n) = RIGHT$(day$(n), 2)
  mo(n) = VAL(mo$(n)): da(n) = VAL(da$(n))
  year(n) = VAL(LEFT$(day$(n), 2))
  day%(n) = (mo(n) - 1) * 30 + da(n)

  SELECT CASE mo(n)

  CASE 2
    day%(n) = day%(n) +

  CASE 3
    day%(n) = day%(n) - 1

  CASE 6
    day%(n) = day%(n) + 1

  CASE 7
    day%(n) = day%(n) + 1

  CASE 8
    day%(n) = day%(n) + 2

  CASE 9
    day%(n) = day%(n) + 3

  CASE 10
    day%(n) = day%(n) + 3

  CASE 11
    day%(n) = day%(n) + 4

  CASE 12
    day%(n) = day%(n) + 4

  CASE ELSE

  END SELECT

 n = n + 1: IF n < 6 THEN GOTO GETMONTH
```

```
CALC:
  n = n - 1
  INPUT "Give local latitude in degrees, north is pos. "; lat
  INPUT "Give local longitude in degrees, west of Greenwich "; lng
  PRINT "Give the local international time zone, counted westward from"
  PRINT "Greenwich (0-23). E.g. Stockholm, Sweden is in zone 23."
  zone% = -1
  DO WHILE zone% < 0 OR zone% > 23
  INPUT zone%
  LOOP

CLS
  SCREEN 9

  DrawKoordinat ort$, lat, lng, n, zone%
  FOR mn = 1 TO n
  LOCATE 3 + mn, 67: PRINT LEFT$(day$(mn), 2); " "; mo$(mn); " "; da$(mn); "   "; mn
  NEXT mn

  SKALINDEXX = 54 / 36
  SKALINDEXY = 2.1
  twopi = 6.2831853#
  RAD = .017453293#
  FOR mn = 1 TO n
  delyr = year(mn) - 80
  leap = INT(delyr / 4)

 k% = 0:
  FOR hr% = 0 TO 23
    FOR min% = 0 TO 50 STEP 10
      T = hr% + min% / 60 + zone%
      time = delyr * 365 + leap + day%(mn) - 1 + T / 24
      IF delyr = leap * 4 THEN time = time - 1
      IF delyr < 0 AND delyr <> leap * 4 THEN time = time - 1
      theta = (360 * time / 365.25) * RAD
      g = -.031271 - 4.53963E-07 * time + theta
      el1 = 4.900968 + 3.6747E-07 * time + (.033434 - 2.3E-09 * time) * SIN(g)
      el2 = .000349 * SIN(2! * g) + theta
      el = el1 + el2
      eps = .40914 - 6.2149E-09 * time
      sel = SIN(el)
      a1 = sel * COS(eps)
      a2 = COS(el)
      ra = ATN(a1 / a2)
      IF ra < 0! THEN ra = ra + twopi
      ain = sel * SIN(eps)
      decl = ATN(ain / SQR(-ain * ain + 1))
      st = 1.759335 + twopi * (time / 365.25 - delyr) + 3.694E-07 * time
      IF st >= twopi THEN st = st - twopi
      s = st + (T * 15! - lng) * RAD
      IF s >= twopi THEN s = s - twopi
      h = ra - s
      phi = lat * RAD
      q = SIN(phi) * SIN(decl) + COS(phi) * COS(decl) * COS(h)
      rc = (1! / (.955 + (20.267 * q))) - .047121
      qrc = q + (.0083 * rc)
      IF qrc > 1! THEN
        elevation = twopi / 4
      ELSEIF qrc < -1! THEN
        elevation = -twopi / 4
      ELSE
        elevation = ATN(qrc / SQR(-qrc * qrc + 1))
      END IF
      IF q > 1! THEN
        ets = twopi / 4
      ELSEIF q < -1! THEN
        ets = -twopi / 4
      ELSE
```

```
        ets = ATN(q / SQR(-q * q + 1))
      END IF
      IF (lat - decl) > 0! THEN
        a = 0
      ELSE
        a = 180!
      END IF
      IF NOT ABS(ets - twopi / 4) < .000001 THEN
        A5 = COS(decl) * SIN(h) / COS(ets)
        JSLASK = (-A5 * A5 + 1)
        IF JSLASK <= 0 THEN
          IF A5 = 1 THEN
            a = 90
          ELSE
            a = -90
          END IF
        ELSE
          a = ATN(A5 / SQR(JSLASK)) / RAD
        END IF
      END IF
      IF NOT SIN(elevation) >= SIN(decl) / SIN(phi) THEN
        IF a < 0! THEN a = a + 360!
        a = 180 - a
      END IF
      elevation = elevation / RAD
        IF elevation < 0 THEN GOTO NEXTMIN
        k% = k% + 1
        azimuth%(k%) = CINT((180 - a) * SKALINDEXX + 60)
        plotelevation%(k%) = CINT(292 - elevation * SKALINDEXY)
        IF min% = 0 THEN
              dotx%(k%) = azimuth%(k%)
              doty%(k%) = plotelevation%(k%)
     REM borttagning av cirklar utanför
         IF ABS(plotelevation%(k% - 1) - plotelevation%(k%)) > 20 AND plotelevation%(k% - 1) > 0 THEN
            dotx%(k%) = 0: doty%(k%) = 0
          END IF
     REM märkning av kl 12.00
        IF min% = 0 AND (hr% = 12 OR hr% = 0) THEN test%(k%) = 1
        END IF

NEXTMIN:
      NEXT min%
    NEXT hr%



REM Sortering i x-led

     FOR i% = 1 TO k% - 1: FOR j% = i% + 1 TO k%
       IF azimuth%(i%) > azimuth%(j%) THEN
         x% = plotelevation%(i%): y% = azimuth%(i%)
         plotelevation%(i%) = plotelevation%(j%)
         azimuth%(i%) = azimuth%(j%)
         plotelevation%(j%) = x%: azimuth%(j%) = y%
       END IF
      NEXT j%
     NEXT i%

REM är 0<lat<24 och datum mellan höst- och vårdagjämning?
  IF NOT (ABS(lat) < 24 AND ABS(lat) > 0 AND day%(n) < 267 AND day%(n) > 81) THEN

    IF plotelevation%(2) >= plotelevation%(1) THEN
       plotelevation%(1) = plotelevation%(2)
       azimuth%(1) = azimuth%(2)
    END IF

    IF plotelevation%(k%) <= plotelevation%(k% - 1) THEN
       plotelevation%(k%) = plotelevation%(k% - 1)
       azimuth%(k%) = azimuth%(k% - 1)
```

```
    END IF

REM Borttagning av cirklar utanför x-intervallet
   FOR i% = 1 TO k%
    IF dotx%(i%) < azimuth%(1) OR dotx%(i%) > azimuth%(k%) THEN
       dotx%(i%) = 0: doty%(i%) = 0
    END IF
   NEXT i%

REM Borttagning av "spikar"

IF ABS(lat) > 25 THEN

     FOR i% = 1 TO k% / 2
      IF ABS(plotelevation%(i% + 1) - plotelevation%(i%)) >= 10 THEN
        plotelevation%(i% + 1) = plotelevation%(i%)
        azimuth%(i% + 1) = azimuth%(i%)
      END IF
     NEXT i%

     FOR i% = (k% / 2 + 1) TO k%
     IF ABS(plotelevation%(i%) - plotelevation%(i% + 1)) >= 10 THEN
        plotelevation%(i% + 1) = plotelevation%(i%)
        azimuth%(i% + 1) = azimuth%(i%)
      END IF
     NEXT i%
END IF

END IF


 IF (ABS(lat) < 24 AND ABS(lat) > 5 AND day%(n) < 267 AND day%(n) > 81) THEN
  REM sortering i y-led
    FOR i% = 1 TO k% - 1: FOR j% = i% + 1 TO k%
    IF azimuth%(i% + 1) < 330 AND plotelevation%(i%) > plotelevation%(j%) THEN
     x% = azimuth%(i%): y% = plotelevation%(i%)
     azimuth%(i%) = azimuth%(j%): plotelevation%(i%) = plotelevation%(j%)
     azimuth%(j%) = x%: plotelevation%(j%) = y%
    END IF
    NEXT j%: NEXT i%

    FOR i% = 1 TO k% - 1: FOR j% = i% + 1 TO k%
     IF azimuth%(i%) > 330 AND plotelevation%(i%) < plotelevation%(j%) THEN
      x% = azimuth%(i%): y% = plotelevation%(i%)
      azimuth%(i%) = azimuth%(j%): plotelevation%(i%) = plotelevation%(j%)
      azimuth%(j%) = x%: plotelevation%(j%) = y%
     END IF
    NEXT j%: NEXT i%
 END IF

 REM Borttagning av utanförliggande punkter
   FOR i% = 1 TO k%
      IF azimuth%(i% + 1) - azimuth%(i%) > 150 AND plotelevation%(i% + 2) < plotelevation%(i% + 3) THEN
         plotelevation%(i% + 2) = plotelevation%(i% + 3)
         azimuth%(i% + 2) = azimuth%(i% + 3)
      END IF
   NEXT i%


  DisplayPlot k%
  DispCircle k%, dotx%(), doty%(), test%()

  xp% = INT(azimuth%(mn * 4) / 8): yp% = INT(plotelevation%(mn * 4) / 14)
  LOCATE yp%, xp%: PRINT mn


 NEXT mn
```

```
LOCATE 5, 1: INPUT "Do you wish a printout (Y/N)"; sv$
IF sv$ = "N" OR sv$ = "n" THEN GOTO slut
LOCATE 5, 1: PRINT "                                  "

'Liggande A4
'Skärmen delas upp i 213 rader med 3 pixels i varje

'Koppla bort automatisk radmatning
'REM LPRINT CHR$(27); "5"; CHR$(0);

FOR KOLUMN% = 0 TO 212

'Initiera skrivaren att göra 3 pixels radmatnig och
'att ta emot 350 bytes för 1 plottnings-rad

  LPRINT CHR$(27); "3"; CHR$(9);
  LPRINT CHR$(27); "K"; CHR$(94); CHR$(1);
'Skärmen uppdelas horisontellt i 350 bytes
  FOR RAD% = 350 TO 1 STEP -1
    PLOTBYTE% = 0
'Hämta in 8 pixels i höjd för varje vertikal byte
    FOR PIXEL% = 1 TO 3
      IF POINT(KOLUMN% * 3 + 4 - PIXEL%, RAD%) > 0 THEN
        PLOTBYTE% = PLOTBYTE% OR 2 ^ (PIXEL% - 1)
      END IF
    NEXT
'Skicka ut en byte till skrivaren
    LPRINT CHR$(PLOTBYTE%);
  NEXT
  LPRINT
NEXT
LPRINT CHR$(12);

slut:
END

SUB DispCircle (k%, dotx%(), doty%(), test%())
   FOR L% = 1 TO k%
    IF test%(L%) = 1 AND (dotx%(L%) < 345 AND dotx%(L%) > 315) THEN r% = 4 ELSE r% = 3
     IF dotx%(L%) <> 0 AND doty%(L%) <> 0 THEN
       CIRCLE (dotx%(L%), doty%(L%)), r%
     END IF
   NEXT L%
END SUB

SUB DisplayPlot (k%)
  FOR L% = 1 TO k%
    IF plotelevation%(L%) > 0 THEN
      a$ = STR$(azimuth%(L%)): e$ = STR$(plotelevation%(L%))
      IF L% = 1 THEN DRAW "BM" + a$ + "," + e$
       IF ABS(azimuth%(L% - 1) - azimuth%(L%)) > 100 THEN
          DRAW "BM" + a$ + "," + e$
       ELSE
          DRAW "M" + a$ + "," + e$
      END IF
    END IF
  NEXT L%
END SUB

SUB DrawKoordinat (ort$, lat, lng, n, zone%)
  DRAW "bm40,292 nr600 u202"
  DRAW "bm60,298  u12 bm195,298 u12 bm330,298 u12 bm465,298 u12"
  j% = 60
  FOR i% = 1 TO 36
    j% = j% + 15: i$ = STR$(j%)
    DRAW "bm" + i$ + ",295 u6"
  NEXT i%
  DRAW "bm40,300"
```

```
      i%
         j%           STR$(j%)
    DRAW "bm37."
  NEXT
  LOCATE       PRINT "This diagram shows     local solar altitude     azimuth        ort$
  LOCATE 2,    PRINT "(in degrees)                  Every hour
  LOCATE 3,    PRINT                                Noon
  LOCATE    9: PRINT "Altitude
  LOCATE        PRINT "N"
  LOCATE     25  PRINT "E"
  LOCATE         PRINT
  LOCATE         PRINT
  LOCATE         PRINT
     SGN(lat)     THEN GOTO
  LOCATE     8: PRINT
  LOCATE         PRINT
  LOCATE 23,     PRINT
North:
  LOCATE         PRINT "Azimuth
  LOCATE 8,    PRINT     LOCATE        PRINT
  LOCATE        PRINT     LOCATE        PRINT
  LOCATE        PRINT
  LOCATE 4,     PRINT "Dates";
  LOCATE            PRINT "Latitude
  LOCATE            PRINT "Longitude
  LOCATE            PRINT "Time         zone%
  CIRCLE (351,      3: CIRCLE (351,
```

App. 2

APPENDIX

```
      SUBROUTINE SUNAE2(YEAR,DAY,HR,MIN,SEC,ZONE,DASVTM,LAT,LONG,A,E)
C
C ****************************************************************
C
C THIS SUBROUTINE CALCULATES THE LOCAL AZIMUTH AND ELEVATION OF
C THE SUN AT A SPECIFIED LOCATION AND TIME, USING AN APPROXIMATION
C TO THE EQUATIONS USED TO GENERATE THE NAUTICAL ALMANAC.
C
C INPUT PARAMETERS:
C     YEAR    - THE YEAR NUMBER (E.G., 1984)
C     DAY     - DAY NUMBER OF THE YEAR, STARTING WITH 1 FOR JANUARY 1,
C               EXCEPT IN LEAP YEARS WHEN 1 SHOULD BE SUBTRACTED
C               FROM THE DAY NUMBER
C     HR,MIN,SEC  - THE TIME OF DAY
C     ZONE    - THE LOCAL INTERNATIONAL TIME ZONE, COUNTED WESTWARD
C               FROM GREENWICH (E.G. SYDNEY, AUSTRALIA IS IN ZONE 14.)
C     DASVTM  - = 1. IF DAYLIGHT SAVING IN EFFECT, ELSE = 0.
C     LAT     - THE LOCAL LATITUDE IN DEGREES (NORTH IS POSITIVE)
C     LONG    - THE LOCAL LONGITUDE IN DEGREES WEST OF GREENWICH
C
C OUTPUT PARAMETERS:
C     A       - AZIMUTHAL ANGLE OF THE SUN, DEGREES EAST OF NORTH
C     E       - ELEVATION OF THE SUN, DEGREES
C     DECL  - DECLINATION OF THE SUN, DEGREES
C
C ****************************************************************
C
      REAL MIN,LAT,LONG
      DATA TWOPI,RAD/6.2831853,0.017453293/
      DELYR=YEAR-1980.
      LEAP=IFIX(DELYR/4.)
      T=HR+(MIN+SEC/60.)/60.+ZONE-DASVTM
      TIME=DELYR*365.+LEAP+DAY-1.+T/24.
      IF (DELYR.EQ.LEAP*4.) TIME=TIME-1.
      IF ((DELYR.LT.0).AND.(DELYR.NE.LEAP*4.)) TIME=TIME-1.
      THETA=(360.*TIME/365.25)*RAD
      G=-0.031271-4.53963E-7*TIME+THETA
      EL=4.900968+3.6747E-7*TIME+(0.033434-2.3E-9*TIME)*SIN(G)
     +  +0.000349*SIN(2.0*G)+THETA
      EPS=0.409140-6.2149E-9*TIME
      SEL=SIN(EL)
      A1=SEL*COS(EPS)
      A2=COS(EL)
      RA=ATAN2(A1,A2)
      IF (RA.LT.0.0) RA=RA+TWOPI
      DECL=ARSIN(SEL*SIN(EPS))
      ST=1.759335+TWOPI*(TIME/365.25-DELYR) + 3.694E-7*TIME
      IF (ST.GE.TWOPI) ST=ST-TWOPI
      S=ST+(T*15.0-LONG)*RAD
      IF (S.GE.TWOPI) S=S-TWOPI
      H=RA-S
      PHI=LAT*RAD
      Q=SIN(PHI)*SIN(DECL)+COS(PHI)*COS(DECL)*COS(H)
      RC=(1./(0.955+(20.267*Q)))-0.047121
      QRC=Q+(0.0083*RC)
      IF (QRC.GT.1.) QRC=1.
      IF (QRC.LT.-1.) QRC=-1.
      E=ARSIN(QRC)
      IF (Q.GT.1.) Q=1.
      IF (Q.LT.-1.) Q=-1.
      ETS=ARSIN(Q)
      IF ((LAT-DECL).GT.0.) GO TO 11
      A=180.
      GO TO 12
   11 A=0.
   12 IF (ABS(ETS-TWOPI/4.).LT..000001) GO TO 10
      A=ARSIN(COS(DECL)*SIN(H)/COS(ETS))/RAD
      IF (SIN(E).GE.SIN(DECL)/SIN(PHI)) GOTO 10
      IF (A.LT.0.0) A=A+360.0
      A=180.0-A
   10 E=E/RAD
      RETURN
      END
```