CT-82-100-41-LR

FAA TECHNICAL CENTER LETTER REPORT

GLOBAL POSITIONING SYSTEM (GPS)

SATELLITE ELEVATION AND AZIMUTH ANGLES

by

Michael Magrogan

U. S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
TECHNICAL CENTER
Atlantic City Airport, N.J. 08405

ABSTRACT

This program was written to plot Global Positioning System (GPS) Satellite elevation and azimuth angles versus Greenwich Mean Time (GMT) in hours. This program is used by Technical Center engineers to predict the GPS satellite visibility window for flight test planning and post flight data analysis. A program flow chart, program listing and sample plots produced by this program are contained in this report.

The author of this letter report is Michael Magrogan, ACT-100B.3 and he may be contacted for additional information at FTS-346-3915, (609) 641-8200, extension 3915.

Key Words

GPS
GMT
Almanac
Satellite window
Flow Chart

Project No. 045-330-130

TABLE OF CONTENTS

ABSTRACT

INTRODUCTION

PURPOSE

BACKGROUND

DISCUSSION

APPENDIX A - GPS Satellite Evaluation and Azimuth Angle Plot Program

APPENDIX B - Almanac Data Collection Program

APPENDIC C - GPS Satellite Position Prediction Program

REFERENCES

LIST OF FIGURES

FIGURE NO.

1 Program Flow Chart

2 Sample Plot of GPS Satellite Elevation Angle vs GMT Time

3 Sample Plot of GPS Satellite Azimuth Angle vs GMT Time

INTRODUCTION

PURPOSE

This program was written to plot Global Positioning System (GPS) Satellite Elevation and Azimuth Angles versus Greenwich Mean Time (GMT) in hours.

BACKGROUND

A program called "A GPS Almanac" was written by Che S. Wong of Texas A&M University in partial fulfillment of his Master of Engineering Degree (Reference 1). This program was modified to run on DEC PDP 11/34 computers at the FAA Technical Center. A listing of this modified program is included in Appendix C. The program predicts rise/fall times, elevation, and azimuth angles of the GPS Satellites.

The inputs to this program are: almanac data file name, identification number of the satellites or satellites excluded from the navigation constellation, the number of satellites in the almanac data file, user position (latitude, longitude, altitude), start time, stop time, computation interval, date of use, option to predict rise/fall times or elevation and azimuth angles. The almanac data file is created by running a program, which is included in Appendix B, that collects the current almanac data from the memory of the GPS Z-set. The modified position prediction program outputs GMT time, satellite constellation and the satellites elevation and azimuth angles to a line printer and a disk file.

The listings of the GPS azimuth and elevation angles were useful, but a graphical representation would provide an easy to use presentation of the satellite visibility window.

DISCUSSION

A program was written to plot the azimuth and elevation angles of the GPS Satellites from a data file created by the GPS Satellite Position prediction program.

The main program logic is displayed in the accompanying flow chart (Figure 1). The user enters the date of use and a plot data file name. The program reads the data file into memory. The first record contains the date of almanac collection. The remaining records in the file consist of GMT time, satellite pseudorandom number (PRN), elevation and azimuth angles.

The data is scaled to an eight inch GMT time axis and a six inch elevation angle axis. An eight and a half by eleven inch border is then drawn around the plot area. The GMT time axis and the elevation angle axis is drawn. A dashed line is drawn at zero elevation. A reference time azis in seconds is also drawn. The elevation angle data is plotted for each satellite. The satellite's PRN number is written after the last elevation angle data point. This elevation angle data plot is finally labeled.

The azimuth angle plot is set up the same as the elevation angle plot. A border is drawn around the plot area. The GMT Time axis and azimuth angle axis are drawn along with a reference time axis in seconds. The azimuth data is plotted for each satellite which is tagged with a PRN number. This azimuth angle plot is labeled and the program terminates. Sample plots are shown in figures 2 and 3. A program listing is contained in Appendix A. All plot subroutine calls are CALCOMP subroutines written for a DEC PDP 11 computer and a CALCOMP 1051 Drum Plotter.

This program is used by the GPS Fixed Wing and Helicopter Navigation groups to predict the satellite visibility window for flight test planning and post flight data analysis.

REFERENCES

Che S. Wong, "A GPS Almanac" Submitted to the Electrical Engineering Department of Texas A&M University in partial fulfillment of the Master of Engineering Degree.

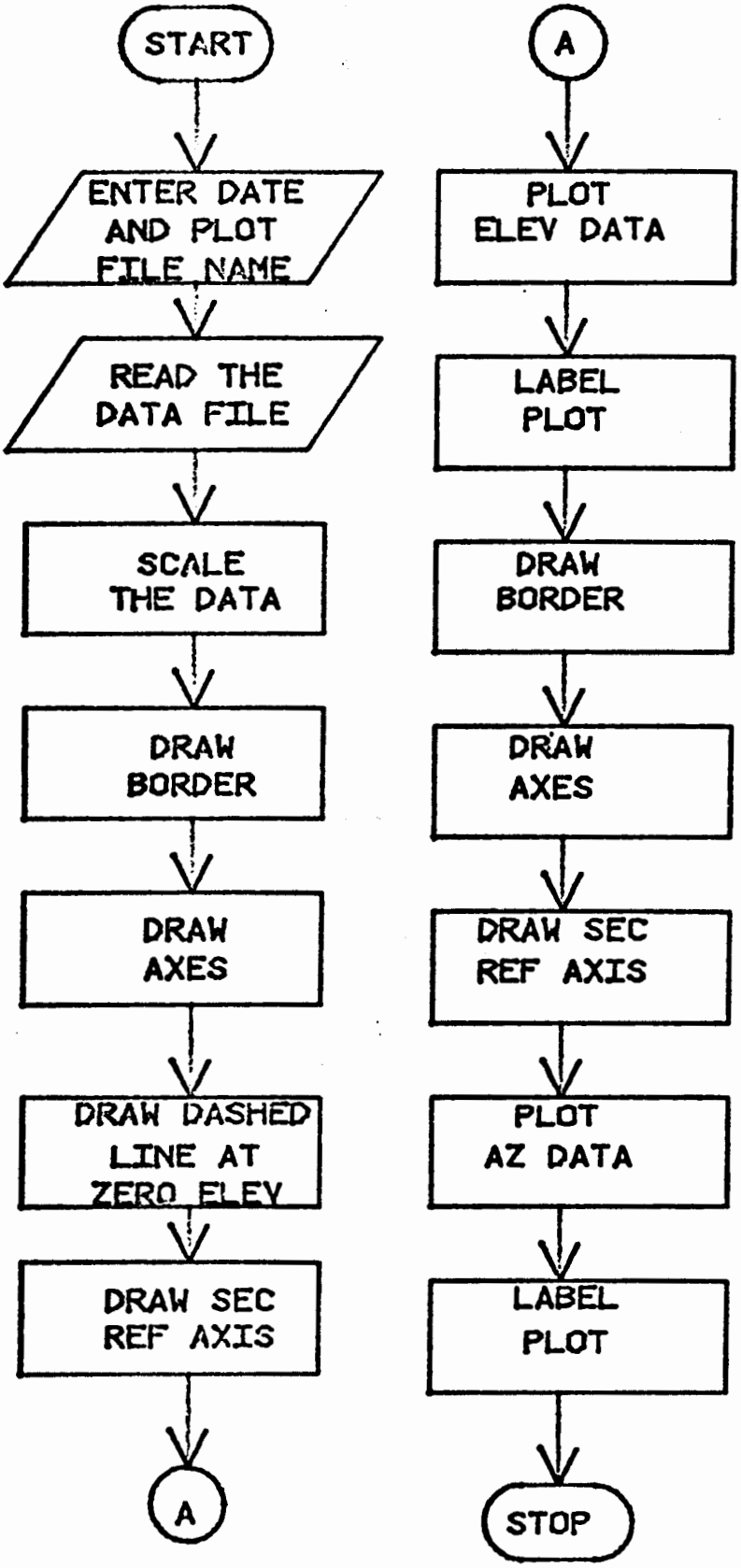

FIGURE 1 - PROGRAM FLOW CHART

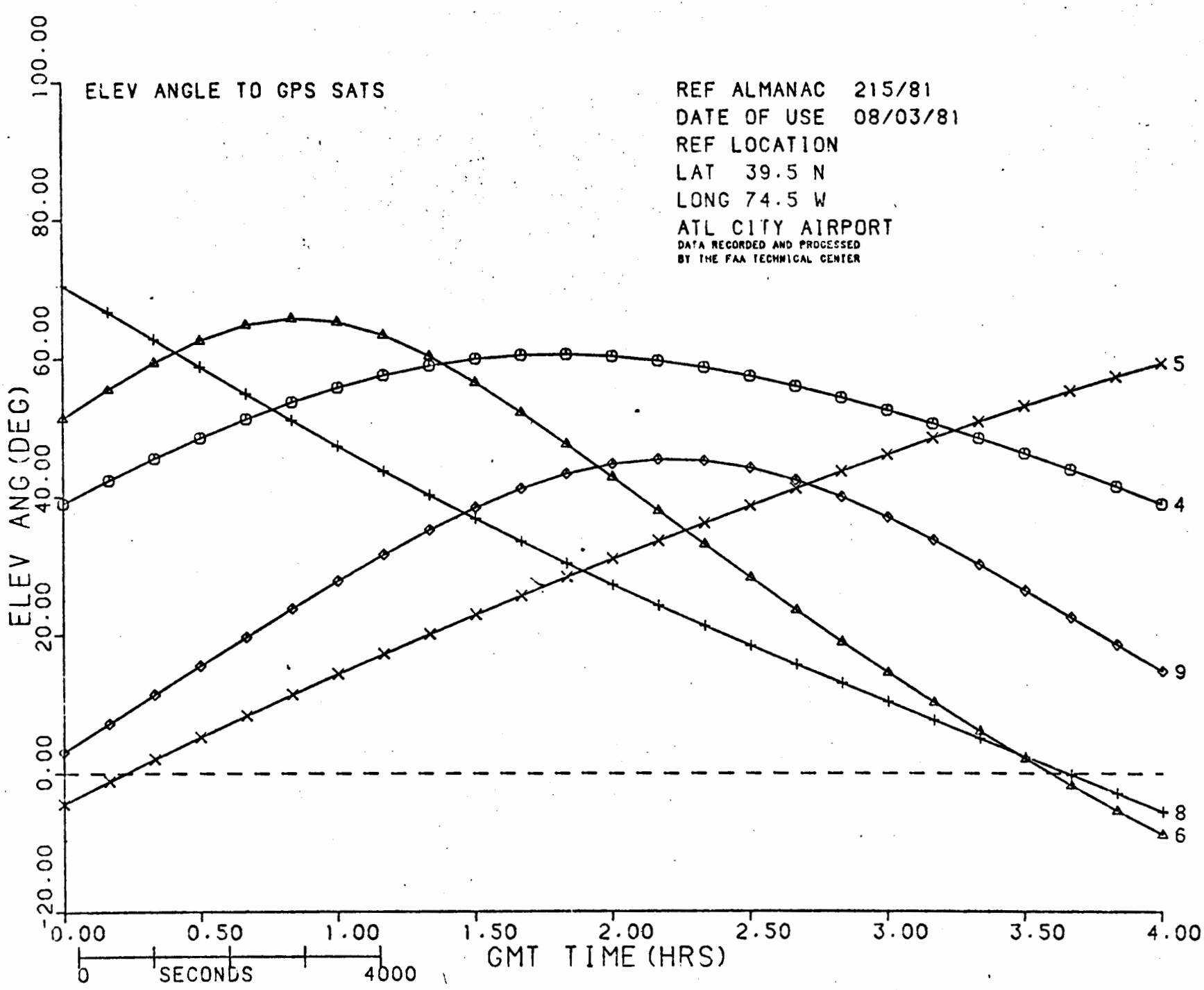

FIGURE 2 - SAMPLE PLOT OF GPS SATELLITE ELEVATION ANGLE VS GMT TIME

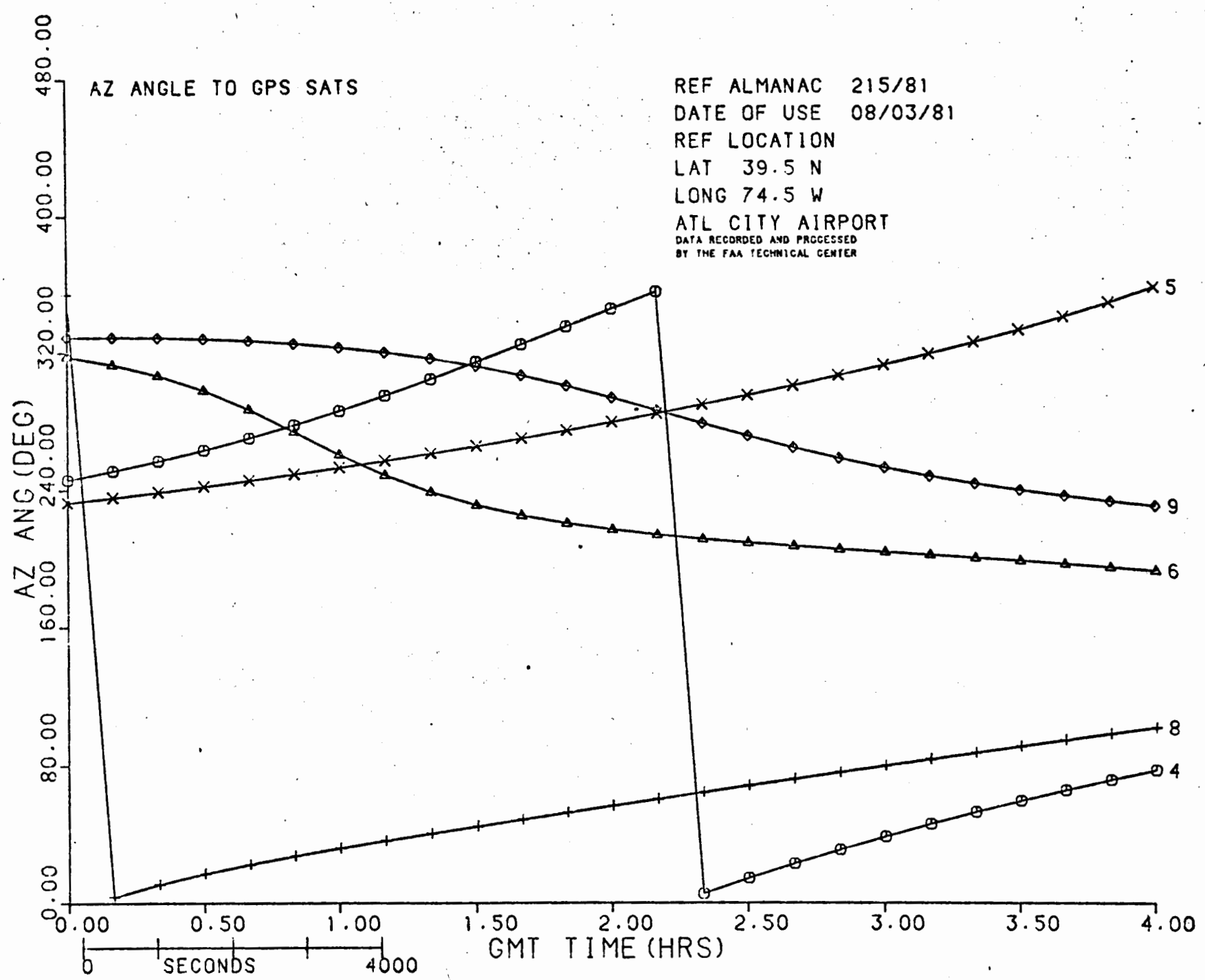

FIGURE 3 - SAMPLE PLOT OF GPS SATELLITE AZIMUTH ANGLE VS GMT TIME

```
      C FILE NAME: AZELPL.FOR
      C LINK WITH CALLIB
      C PROGRAM TO PLOT AZIMUTH AND ELEVATION ANGLES OF THE GPS SATS.
      C MICHAEL J. MAGROGAN CO-OP ACT-1008 FEB 1982
0001          DIMENSION X(135),YEL(135),YAZ(135),ISAT(125),ISYMB(9)
0002          DIMENSION IDATE(4),DARRAY(2)
0003          DATA IOPT/0/,ISYMB/36,38,40,37,41,4*0/,DARRAY/-.1,.1/
0004          WRITE(5,98)
0005       98 FORMAT(1X,'ENTER DATE OF USE MM/DD/YY:')
0006          READ(5,99) IDATE
0007       99 FORMAT(4A2)
0008          WRITE(5,100)
0009      100 FORMAT('$ENTER INPUT FILE SPEC,PLOT DATA FILE=')
0010          CALL ASSIGN(1,,-1,'RDO')
0011          READ(1,101) DCOL
0012      101 FORMAT(G15.8)
0013          DO 10 I=1,125
0014          READ(1,102) IH,IM,IS,ISAT(I),YEL(I),YAZ(I)
0015      102 FORMAT(I3,1X,I2,I2,1X,I3,1X,F6.2,1X,F6.2)
0016          X(I)=FLOAT(IH)+(FLOAT(IM)/60.)+(FLOAT(IS)/3600.)
0017       10 CONTINUE
0018          CALL PLOTS(I,0,I)
0019          CALL PLOT(0.,0.,-3)
      C SCALE THE DATA.
0020          CALL SCALE(X,8.,125,1)
0021          CALL SCALE(YEL,6.,125,1)
0022          CALL SCALE(YAZ,6.,125,1)
      C DRAW BORDER FOR PLOTS.
0023          CALL PLOT(0.,8.5,2)
0024          CALL PLOT(11.,8.5,2)
0025          CALL PLOT(11.,0.,2)
0026          CALL PLOT(0.,0.,2)
0027          CALL PLOT(1.5,1.25,-3)
      C DRAW AXIS.
0028          CALL AXIS(0.,0.,'GMT TIME(HRS)',-13,8.,0.,X(126),X(127))
0029          CALL AXIS(0.,0.,'ELEV ANG(DEG)',13,6.,90.,YEL(126),YEL(127))
      C DRAW DASHED LINE AT ZERO ELEVATION.
0030          CALL PLOT(0.,1.,3)
0031          CALL DASHS(DARRAY,2)
0032          CALL PLOT(8.,1.,2)
0033          CALL DASHS(DARRAY,0)
      C DRAW SECONDS REFERENCE AXIS.
0034          CALL PLOT(.1,-.25,3)
0035          CALL PLOT(.1,-.4,2)
0036          CALL PLOT(.65,-.4,3)
0037          CALL PLOT(.65,-.25,2)
0038          CALL PLOT(1.2,-.25,3)
0039          CALL PLOT(1.2,-.4,2)
0040          CALL PLOT(1.75,-.4,3)
0041          CALL PLOT(1.75,-.25,2)
0042          CALL PLOT(2.3,-.25,3)
0043          CALL PLOT(2.3,-.4,2)
0044          CALL PLOT(.1,-.325,3)
0045          CALL PLOT(2.3,-.325,2)
```

```
      C PLOT DATA.
 0095         YAZ(131)=YAZ(127)
 0096         DO 30 I=1,5
 0097         YAZ(130+I)=YAZ(131)
 0098         YAZ(125+I)=YAZ(126)
 0099         II=I
 0100         CALL LINE(XC(I),YAZ(I),25,5,1,II)
 0101         CALL WHERE(RXPAGE,RYPAGE,RFACT)
 0102         CALL SYMBOL(RXPAGE+.1,RYPAGE-.05,.1,ISYMB(I),0.,-1)
 0103         CALL PLOT(0.,0.,3)
 0104      30 CONTINUE
      C LABEL PLOT.
 0105         CALL SYMBOL(.2,5.9,.1,'AZ ANGLE TO GPS SATS',0.,20)
 0106         CALL SYMBOL(4.5,5.9,.1,'REF ALMANAC  ',0.,13)
 0107         CALL NUMBER(999.,999.,.1,DCOL,0.,-1)
 0108         CALL SYMBOL(999.,999.,.1,'81',0.,3)
 0109         CALL SYMBOL(4.5,5.7,.1,'DATE OF USE  ',0.,13)
 0110         CALL SYMBOL(999.,999.,.1,IDATE,0.,8)
 0111         CALL SYMBOL(4.5,5.5,.1,'REF LOCATION',0.,12)
 0112         CALL SYMBOL(4.5,5.3,.1,'LAT  39.5 N',0.,11)
 0113         CALL SYMBOL(4.5,5.1,.1,'LONG 74.5 W',0.,11)
 0114         CALL SYMBOL(4.5,4.9,.1,'ATL CITY AIRPORT',0.,16)
 0115         CALL SYMBOL(4.5,4.8,.05,'DATA RECORDED AND PROCESSED',0.,27)
 0116         CALL SYMBOL(4.5,4.7,.05,'BY THE FAA TECHNICAL CENTER',0.,27)
 0117         CALL PLOT(-1.5,8.,999)
 0118         STOP 'END OF PROGRAM'
 0119         END
```

```
      CALL SYMBOL(.1,-.5,.1,'0      SECONDS          4000',0.,25)
      CALL PLOT(0.,0.,3)
C PLOT DATA.
      X(131)=X(127)
      YEL(131)=YEL(127)
      DO 20 I=1,5
      X(130+I)=X(131)
      YEL(130+I)=YEL(131)
      X(125+I)=X(126)
      YEL(125+I)=YEL(126)
      II=I
      CALL LINE(X(I),YEL(I),25,5,1,II)
      CALL WHERE(RXPAGE,RYPAGE,RFACT)
      CALL SYMBOL(RXPAGE+.1,RYPAGE-.05,0.1,ISYMB(I),0.,-1)
      CALL PLOT(0.,0.,3)
   20 CONTINUE
C LABEL PLOT.
      CALL SYMBOL(.2,5.9,.1,'ELEV ANGLE TO GPS SATS',0.,22)
      CALL SYMBOL(4.5,5.9,.1,'REF ALMANAC  ',0.,13)
      CALL NUMBER(999.,999.,.1,DCOL,0.,-1)
      CALL SYMBOL(999.,999.,.1,'/81',0.,3)
      CALL SYMBOL(4.5,5.7,.1,'DATE OF USE--',0.,13)
      CALL SYMBOL(999.,999.,.1,IDATE,0.,8)
      CALL SYMBOL(4.5,5.5,.1,'REF LOCATION',0.,12)
      CALL SYMBOL(4.5,5.3,.1,'LAT  39.5 N',0.,11)
      CALL SYMBOL(4.5,5.1,.1,'LONG 74.5 W',0.,11)
      CALL SYMBOL(4.5,4.9,.1,'ATL CITY AIRPORT',0.,16)
      CALL SYMBOL(4.5,4.8,.05,'DATA RECORDED AND PROCESSED',0.,27)
      CALL SYMBOL(4.5,4.7,.05,'BY THE FAA TECHNICAL CENTER',0.,27)
      CALL PLOT(-1.5,0.,-3)
C DRAW BORDER FOR PLOTS.
      CALL PLOT(0.,8.5,2)
      CALL PLOT(11.,8.5,2)
      CALL PLOT(11.,0.,2)
      CALL PLOT(0.,0.,2)
      CALL PLOT(-1.5,1.25,-3)
C DRAW AXIS.
      CALL AXIS(0.,0.,'GMT TIME(HRS)',-13,8.,0.,X(126),X(131))
      CALL AXIS(0.,0.,'AZ ANG(DEG)',11,6.,90.,YAZ(126),YAZ(127))
C DRAW SECONDS REFERENCE AXIS.
      CALL PLOT(.1,-.25,3)
      CALL PLOT(.1,-.4,2)
      CALL PLOT(.65,-.4,3)
      CALL PLOT(.65,-.25,2)
      CALL PLOT(1.2,-.25,3)
      CALL PLOT(1.2,-.4,2)
      CALL PLOT(1.75,-.4,3)
      CALL PLOT(1.75,-.25,2)
      CALL PLOT(2.3,-.25,3)
      CALL PLOT(2.3,-.4,2)
      CALL PLOT(.1,-.325,3)
      CALL PLOT(2.3,-.325,2)
      CALL SYMBOL(.1,-.5,.1,'0      SECONDS          4000',0.,25)
      CALL PLOT(0.,0.,3)
```

LOCAL VARIABLES, .PSECT $DATA, SIZE = 004044 (1042. WORDS)

NAME	TYPE	OFFSET	NAME	TYPE	OFFSET	NAME	TYPE	OFFSET
BEGL	R*4	003606	I	I*2	003612	IH	I*2	003614
II	I*2	003622	IM	I*2	003616	IOPT	I*2	003560
IS	I*2	003620	RPACT	R*4	003634	RXPAGE	R*4	003624
RYPAGE	R*4	003630						

LOCAL AND COMMON ARRAYS:

NAME	TYPE	SECTION	OFFSET	SIZE	DIMENSIONS
BXARRAY	R*4	$DATA	003550	000010	(4.) (2)
IDATE	I*2	$DATA	003540	000010	(4.) (4)
IGAT	I*2	$DATA	003124	000372	(125.) (125)
ISYMB	I*2	$DATA	003516	000022	(9.) (9)
X	R*4	$DATA	000000	001034	(270.) (135)
YAZ	R*4	$DATA	002070	001034	(270.) (135)
YEL	R*4	$DATA	001034	001034	(270.) (135)

SUBROUTINES, FUNCTIONS, STATEMENT AND PROCESSOR-DEFINED FUNCTIONS:

NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
ASSIGN	R*4	AXIS	R*4	DASHS	R*4	FLOAT	R*4	LINE	I*2
NUMBER	I*2	PLOT	R*4	PLOTS	R*4	SCALE	R*4	SYMBOL	R*4
WHERE	R*4								

```
C*************************************************************************
C FILE NAME: ALCOL.FOR
C LINK WITH ZREAD
C
C       PROGRAM TO COLLECT ALMANAC DATA FROM THE ZSET.
C
C       SEE DOCUMENT:  CP-US-305,PAGE 10-12, 20 JUN 78
C                      TABLE 10-V  PACKED DATA BLOCK III
C
C       MICHAEL J. MAGROGAN  CO-OP  ACT-100D  JANUARY 1981
C       GPS HELICOPTER PROGRAM
C*************************************************************************
      DIMENSION IALMAC(12,24),ISVPRN(24)
      BYTE ALMAC(24,24),IAR(4)
      REAL*4 OMD,OMO,MO
      INTEGER*4 JVAL
      INTEGER H
      EQUIVALENCE (IAR(1),JVAL),(IALMAC(1,1),ALMAC(1,1))
      DATA PI/3.14159265/,ISVPRN/4,7,6,8,5,9,18*0/
C  ISTAD IS THE STARTING ADDRESS OF ALMANAC DATA AND IWCNT IS THE NUMBE
C  OF WORDS IN THE ALMANAC DATA.
      DATA ISTAD/O172700/,IWCNT/288/
C  CALL ZREAD TO READ ALMANAC DATA FROM THE ZSET.
      CALL ZREAD(ISTAD,IWCNT,IALMAC(1,1))
C  ASK THE USER TO ENTER DISK FILE NAME AND DATE OF COLLECTION.
      WRITE(5,100)
      WRITE(5,110)
      CALL ASSIGN(1,,-1)
      WRITE(5,120)
      READ(5,130) DTC
C  WRITE DATE OF COLLECTION TO DISK FILE.
      WRITE(1,130) DTC
      DO 10 J=1,24
C  GET SATELLITE ID(PRN NUMBER) FROM ALMANAC DATA.
        DO 20 I=1,24
        ID=ALMAC(1,I)
C  TEST TO SEE IF SATELLITE IS IN PRN NUMBER ORDER.
        IF(ID .EQ. 0) GO TO 20
        IF(ID .EQ. ISVPRN(J)) GO TO 30
   20 CONTINUE
      GO TO 10
C  GET ECENTRICITY OF SATELLITE ORBIT FROM ALMANAC DATA.
   30 IAR(1)=ALMAC(3,I)
      IAR(2)=ALMAC(2,I)
      IAR(3)=0
      IAR(4)=0
C  ECENTRICITY IS SCALED BY A FACTOR OF 2**-21.
      E=AJFLT(JVAL)*4.7683716E-7
C  GET TIME OF APPLICABILITY OF THE ALMANAC PARAMETERS.
      IAR(1)=ALMAC(4,I)
      IAR(2)=0
      IAR(3)=0
      IAR(4)=0
C  TOA(SECONDS) IS SCALED BY A FACTOR OF 2**12.
      TOA=AJFLT(JVAL)*4096.0
C  GET SATELLITE ORBIT INCLINATION FROM ALMANAC DATA.
      IAR(1)=ALMAC(6,I)
      IAR(2)=ALMAC(5,I)
      IAR(3)=0
      IAR(4)=0
      IF(IAR(2) .LT. 0) IAR(3)=-1
      IF(IAR(2) .LT. 0) IAR(4)=-1
C  DI(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-19 (INCREMENTS
```

```
C  FROM ALMANAC DATA.
      IAR(1)=ALMAC(8,I)
      IAR(2)=ALMAC(7,I)
      IAR(3)=0
      IAR(4)=0
      IF(IAR(2) .LT. 0) IAR(3)=-1
      IF(IAR(2) .LT. 0) IAR(4)=-1
C  OMD(SEMICIRCLES/SEC) IS SCALED BY A FACTOR OF 2**-38 AND
C  MULTIPLIED BY PI TO YIELD RADIANS/SEC.
      OMD=AJFLT(JVAL)*PI*3.6379788E-12
C  GET HEALTH WORD FROM ALMANAC DATA.
      H=ALMAC(9,I)
C  GET THE SQUARE ROOT OF THE SEMI-MAJOR AXIS OF THE ORBIT
C  FROM ALMANAC DATA.
      IAR(1)=ALMAC(12,I)
      IAR(2)=ALMAC(11,I)
      IAR(3)=ALMAC(10,I)
      IAR(4)=0
C  RA(METERS) IS SCALED BY A FACTOR OF 2**-11.
      RA=AJFLT(JVAL)*4.8828125E-4
C  GET THE RIGHT ASCENSION OF THE ORBIT FROM ALMANAC DATA.
      IAR(1)=ALMAC(15,I)
      IAR(2)=ALMAC(14,I)
      IAR(3)=ALMAC(13,I)
      IAR(4)=0
      IF(IAR(3) .LT. 0) IAR(4)=-1
C  OMO(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
      OMO=AJFLT(JVAL)*PI*1.1920929E-7
C  GET THE ARGUMENT OF PERIGEE FROM ALMANAC DATA.
      IAR(1)=ALMAC(18,I)
      IAR(2)=ALMAC(17,I)
      IAR(3)=ALMAC(16,I)
      IAR(4)=0
      IF(IAR(3) .LT. 0) IAR(4)=-1
C  W(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
      W=AJFLT(JVAL)*PI*1.1920929E-7
C  GET MEAN ANOMALY AT TOA FROM ALMANAC DATA.
      IAR(1)=ALMAC(21,I)
      IAR(2)=ALMAC(20,I)
      IAR(3)=ALMAC(19,I)
      IAR(4)=0
      IF(IAR(3) .LT. 0) IAR(4)=-1
C  MO(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
      MO=AJFLT(JVAL)*PI*1.1920929E-7
C  WRITE ALMANAC DATA TO DISK FILE.
      WRITE(1,149) ID
      WRITE(1,150) E,TOA,DI,OMD
      WRITE(1,150) RA,OMO,W,MO
      WRITE(1,140)
   10 CONTINUE
      STOP
  100 FORMAT(' PROGRAM TO EXTRACT ALMANAC DATA FROM ZSET.')
  110 FORMAT('$ENTER FILE SPEC FOR OUTPUT,ALMANAC DATA FILE=')
  120 FORMAT(' ENTER COLLECTION DATE FOR ALMANAC DATA(JULIAN DAY)=')
  130 FORMAT(G15.8)
  140 FORMAT('  ')
  149 FORMAT(' ',I2)
  150 FORMAT(' ',E15.8,',',E15.8,',',E15.8,',',E15.8)
      END
```

```
            .TITLE  ZSET I/O FROM FORTRAN
            .SBTTL ZSET BLOCK READ
            .GLOBL ZREAD,ZWRITE,RADD,DTA,HENBIT,DISHLT,TFREAD
    ; CALL FROM FORTRAN --- CALL ZREAD(ISTAD,IWCNT,IBUFF(1))
    ; WHERE ISTAD IS THE STARTING ADDRESS IN THE ZSET
    ;       IWCNT IS THE NUMBER OF WORDS TO READ
    ;       IBUFF(1) IS THE FIRST ELEMENT OF AN INTEGER*1 DATA AREA
    ;
    ZCS     =       164400
    ZSB     =       ZCS+2
    ZADD    =       ZCS+4
    ZDAT    =       ZCS+6
    LOC     =       164410
    GS      =       LOC+2
    FLAG    =       LOC+4
    ;
    ZREAD:  MOV     (R5)+,R0        ;BUMP ARGUMENT COUNT
            MOV     @(R5)+,R0       ;STARTING ADDRESS
            MOV     @(R5)+,R1       ;WORD COUNT
            MOV     (R5),R2 ;DATA AREA POINTER
    ;
    2$:     TSTB    ZCS     ;TRANSFER COMPLETE
            BPL     2$
    4$:     MOV     R0,ZADD ;WRITE ADDRESS
            MOV     #1,ZSB  ;READ+GO
    3$:     TSTB    ZCS     ;DONE?
            BPL     3$
            MOV     ZDAT,(R2)+      ;GET DATA WORD
            INC     R0
            INC     R0
            DEC     R1
            BNE     4$      ;ALL WORDS DONE?
            RTS     PC
            .PAGE
            .SBTTL ZSET ONE WORD WRITE
    ; CALL FROM FORTRAN --- CALL ZWRITE(IADD,IDAT)
    ; WHERE IADD IS 16-BIT INTEGER ZSET ADDRESS
    ;       IDAT IS 16-BIT INTEGER DATA
    ;
    ZWRITE: MOV     (R5)+,R0        ;BUMP ARGUMENT COUNT
    ;
    1$:     TSTB    ZCS     ;TRANSFER COMPLETE
            BPL     1$
            MOV     @(R5)+,ZADD     ;WRITE ADDRESS
            MOV     @(R5),ZDAT      ;WRITE DATA
            MOV     #3,ZSB  ;WRITE+GO
    2$:     TSTB    ZCS     ;DONE?
            BPL     2$
            RTS     PC
            .PAGE
            .SBTTL ZSET NO CARRY ADD
    ; CALL FROM FORTRAN --- CALL RADD(ISUM,ITERM(I))
    ; WHERE ISUM IS 16-BIT INTEGER SUM
    ;       ITERM(I) IS A 16-BIT WORD TO BE SUMMED
    ;
    RADD:   ADD     @4(R5),@2(R5)   ;ADD TERM TO SUM
            RTS     PC
            .PAGE
            .SBTTL ZSET DIGITAL TO ANALOG CONVERSION
    ; CALL FROM FORTRAN --- CALL DTA(ISCALE,IGLSL,IFLAG)
    ; WHERE ISCALE IS THE SCALED CROSS TRACK
    ;       IGLSL IS THE SCALED GLIDE SLOPE
    ;       IFLAG IS THE TO/FROM FLAG AND WAYPOINT WARNING
```

```
        MOV     @4(R5),GS       ;MOVE GLIDE SLOPE TO DTA
        MOV     @6(R5),FLAG     ;MOVE TO/FROM AND WARNING TO DTA
        RTS     PC
        .PAGE
        .SBTTL ZSET HALT ENABLE BIT IESI
; CALL FROM FORTRAN --- CALL HENBIT(ISW)
; WHERE ISW IS THE STATUS OF THE HALT SWITCH
;
HENBIT: MOV     @2(R5),R0       ;MOVE ISW TO R0
        BIT     #10000,ZCS      ;TEST FOR HALT SWITCH SET
        BEQ     1$
        MOV     #10000,ZCS      ;ENABLE HALT
        MOV     #1,@2(R5)       ;SET ISW
        RTS     PC
; HALT SWITCH NOT SET
1$:     MOV     #0,@2(R5)       ;RESET ISW
        RTS     PC
        .PAGE
        .SBTTL ZSET DISABLE HALT
; CALL FROM FORTRAN --- CALL DISHLT
DISHLT: MOV     #0,ZCS  ;DISABLE HALT
        RTS     PC
        .PAGE
        .SBTTL ZSET TO-FROM WAYPOINT SWITCHES READ
; CALL FROM FORTRAN --- CALL IFREAD(IWPT)
; WHERE IWPT IS A 16-BIT WORD CONTAINING THE FROM WAYPOINT
;       IN THE MOST SIGNIFICANT BYTE AND THE TO WAYPOINT
;       IN THE LEAST SIGNIFICANT BYTE.
;
TFREAD: MOV     @#14454,@2(R5) ;MOVE TO-FROM SWITCHES TO IWPT
        RTS     PC
        .END
```

```
BLOCK DATA
IMPLICIT REAL*4(A-Z)
INTEGER NAC
COMMON /CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2/DB2/AD(14,8)
COMMON/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
DATA PI/3.14159265E0/,RTD/57.2957795E0/,PI2/6.2831853E0/
DATA OMF/.996647220E0/,AE/6.378135E6/,NAC/0/
DATA E2/6.694317778E-3/,OMF2/.993305683E0/
DATA SGC/1.996499E7/,ERATE/7.292115147E-5/
END
```

APPENDIX C-1

```
C THIS SYSTEM CAN HANDLE UP TO 8 SAT. IN THE INPUT DATA
      IMPLICIT REAL*4(A-H,K-Z)
      INTEGER CONSTL(4),PERMUT(8),NAC,NUM,ONE
      LOGICAL SINGL,NEXDAY,FIX
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),DT(8,3)
      DOUBLE PRECISION U(4),LAT,LON,ALT,R
      COMMON/TIME/TDEL,TFIN,TSKIP/NUM/NUM/VIS/SINGL,ONE,FIX
      COMMON/AZM/ELV(8),AZM(8)/FLAG/IFLG,IFLAG,IPATH
      COMMON/POS/LAT,LON,ALT,R/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2/DP2/AD(14,8),ISNU
     1(24)
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),H(4,4),SVIS(8)
      COMMON/SAT/S(4),DA(14)
      COMMON/THR/THR
      DIMENSION IDONT(24),ID(24)
C
C  INITIALIZE OUTPUT DEVICES 1,3,4 AND 6
      WRITE(5,700)
      WRITE(6,700)
      WRITE(5,730)
      CALL ASSIGN(1,,-1,'RDO')
      WRITE(5,760)
      CALL ASSIGN(2,,-1)
      WRITE(5,740)
      READ(5,750) (IDONT(I),I=1,24)
C
C  INITIALIZE VARIABLES
C
      SINGL=.FALSE.
      NUM=1
      TC=0.
      ISCNT=0
C
C  READ THE DATE OF COLLECTION FOR THE ALMANAC BEING USED AS THE
C  NTH DAY OF THE YEAR
C
C   10 WRITE(5,600,ERR=10)
C      READ(5,500,ERR=10)DTC
      REWIND 1
       READ(1,500) DTC
       WRITE(2,500) DTC
C
C READ AND SCALE ALMANIC DATA FROM DISK AND ESTABLISH THE
C  NUMBER OF SAT. USED IN THIS ALMANAC.
C
   20 READ(1,619,END=30,ERR=20) ID(ISCNT+1)
      READ(1,620,END=30,ERR=20) (AD(I,NUM),I=1,8)
      ISCNT=ISCNT+1
      DO 21 I=1,24
      IF(IDONT(I) .EQ. ISCNT) GO TO 20
   21 CONTINUE
      IF (AD(8,NUM).EQ.0.0)GO TO 20
      ISNUM(NUM)=ISCNT
      NUM=NUM+1
      GO TO 20
   30 NUM=NUM-1
   31 WRITE(5,570,ERR=31)NUM
      READ (5,520,ERR=31)NUM
      WRITE(6,610)
      DO 60 J=1,NUM
      ELV(1)=AD(1,J)
      ELV(2)=AD(2,J)
      DO 40 I=3,8
```

```
      ELV(5)=AD(5,J)
      WRITE(6,63)  (ELV(I),I=1,8)
      AD(9,J)=AD(5,J)**2
      AD(10,J)=COS(AD(3,J))
      AD(11,J)=SIN(AD(3,J))
      AD(12,J)=COS(AD(7,J))
      AD(13,J)=SIN(AD(7,J))
   60 AD(14,J)=SGC/(AD(5,J)*AD(9,J))
C  INPUT THE LAT,LON, AND ALT, OF THE DESIRED POSITION.
C  LON WEST AND LAT SOUTH ARE DENOTED BY A NEGATIVE VALUE
C
   70 WRITE(5,580,ERR=70)
      NEXDAY=.FALSE.
      FIX=.FALSE.
      LAT=39.5
      LON=-74.5
      ALT=70.0
      READ(5,520,ERR=70)IFLOW
      IF(IFLOW.EQ.1)GO TO 90
   80 WRITE (5,510)
      READ(5,500,ERR=90)LAT
      READ(5,500,ERR=80)LON
      READ(5,500,ERR=80)ALT
   90 LAT=LAT/RTD
      LON=LON/RTD
      CALL ENCOSE
C
C   ECEF TAKES LAT, LON, AND ALT; IT RETURNS X,Y,Z IN U-VECTOR
C
      CALL ECEF
      WRITE(5,530)
      U(4)=0.
      CALL LLA
      DO 100 I=1,3
  100 DIRCU(I)=U(I)/R
C
C   THERE ARE TWO OPTIONS WHICH CAN BE ENTERED FOR IPATH
C   1  THIS GIVES THE RISE-3, RISE-4, FALL-4, AND FALL-3 TIMES
C      OVER A RANGE OF DAYS INPUT BY THE USER AT THE PROMPTING OF THE
C      PROGRAM.
C   2  THIS GIVES THE ELV. AND AZM. OF THE SAT. ON THE DAY ENTERED BY
C      THE USER. WHEN 4 OR MORE SAT. ARE VISIBLE, THE OPTIMAL GDOP IS
C       GIVEN ALONG WITH THE CONSTELATION USED TO CALCULATE THE GDOP.
C
      READ(5,500,ERR=150)T
      READ(5,500,ERR=150)TFIN
      READ(5,500,ERR=150)TDEL
      READ(5,500,ERR=150)DTU
      READ(5,520,ERR=150)IPATH
      IF(IPATH.EQ.2)WRITE(6,555)DTU,DTC
      IF(IPATH.GT.1) GO TO 120
  110 WRITE(5,590)
      READ(5,500,ERR=110)FDTU
      WRITE(5,710)
  115 READ(5,500,ERR=115)TSKIP
      IF(TSKIP .LT. 1.0)TSKIP=1.0
  120 IF(IPATH .EQ. 2)WRITE(6,560)
      WRITE(6,550)T
      WRITE (6,540)LAT,LON,ALT,R,U
      DTU=(DTU-1.0)*86400.0
      TCY=(DTC-1.0)*86400.0+AMOD(TC,86400.0)
      IF(IPATH.EQ.2)GO TO 140
      FIX =.TRUE.
```

```
  140 CALL VISBLE
      GDOP=0.0
      DO 141 I=1,4
  141 CONSTL(I)=0
      CALL SATSEL
      T=T+TDEL
      IF(T .LT. 86400.0)GO TO 143
      IF(NEXDAY)GO TO 143
      NEXDAY=.TRUE.
      DTU=DTU+86400.0
  143 DO 1410 I=1,NUM
      WRITE(2,770) THR,ID(I),EL(I),AZM(I)
 1410 CONTINUE
      IF (T.LE.TFIN)GO TO 140
  142 WRITE(5,720,ERR=142)
      READ(5,520,ERR=142)IC
      IF (IC .NE. 0)GO TO 70
  150 CALL CLOSE(2)
      STOP
  500 FORMAT(G15.8)
  510 FORMAT(' INPUT : ALMANAC DATA; LAT,LON,ALT OF USER')
  520 FORMAT(I5)
  530 FORMAT(' IO, TFINAL, DELTA T, DTU, IPATH')
  540 FORMAT(2(1H ,4(5X,G15.8)/))
  550 FORMAT(//,12X,'LAT',17X,'LON',17X,'ALT',15X,'RANGE   TIME=',G15.
  555 FORMAT(10X,'DATE OF USE',F8.1,6X,'ALMANAC DATE',F8.1)
  560 FORMAT('   TIME',48X,' ELEVATION/AZIMUTH')
  570 FORMAT(//,6X,'THERE ARE ',I2,' SATS IN THE DATA FILE.',/,8X,
     *'ENTER THE NUMBER OF SATS TO BE USED')
  580 FORMAT(' ENTER: LOCAL POSITION-1, OTHER POSITION-2')
  590 FORMAT(' TERMINATION DATE?')
C  600 FORMAT(///,6X,' DATE OF COLLECTION')
  610 FORMAT(9X,'ECC',12X,' TOA',12X,'INOT',12X,'NRATE',12X,'SRA',13X,
     *'OMEO',13X,'OME',12X,'MNOT')
  619 FORMAT(I3)
  620 FORMAT(2(4E18.8,/))
  630 FORMAT(1H ,SG16.7)
  640 FORMAT(1H ,G16.7)
  700 FORMAT('+')
  710 FORMAT('  TSKIP?  DEFAULT= TDEL')
  720 FORMAT(6X,'CONTINUE?      1-YES 0-NO')
  730 FORMAT('$ENTER FILE SPEC FOR INPUT,ALMANAC DATA FILE=')
  740 FORMAT(' DO YOU WANT TO EXCLUDE A SATELLITE IN THE ALMANAC DATA F
     1LE? (SV NUMBERS(1,2,...,24),DEFAULT=NO SATELLITE EXCLUDED)')
  750 FORMAT(24I3)
  760 FORMAT('$ENTER FILE SPEC FOR OUTPUT,PLOT DATA FILE=')
  770 FORMAT(E8.4,1X,I3,1X,F6.2,1X,F6.2)
      END
```

```
      FUNCTION ACOS(X)
      COMMON/CNST/PI,RTD
C
      IF(X.EQ.0.)GO TO 15
      ACOS=ATAN(SQRT(1./(X*X)-1.))
      IF(X.LT.0.)ACOS=PI-ACOS
      ACOS=ACOS*RTD
      RETURN
   15 ACOS=90.0
      RETURN
      END
```

```
      SUBROUTINE RANGE(I)
      DOUBLE PRECISION U(4)
      COMMON/SAT/S(4)/DTX/U,D(8,4),R(8),RM(8),DELR(8),H(4,4)
C
      R(I)=0.
      DO 5 J=1,3
      D(I,J)=U(J)-S(J)
    5 R(I)=D(I,J)*D(I,J)+R(I)
      R(I)=SQRT(R(I))
      D(I,4)=1.0
C     D(I,4)=R(I)
C     R(I)=R(I)+U(4)
C     DELR(I)=D(I,4)*(RM(I)-R(I))
      RETURN
      END
```

```
C 'SGDOP' CALCULATES THE GDOP OF GIVEN FOUR SATELLITES
C
      SUBROUTINE SGDOP
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),H(4,4),RVIS(8)
      COMMON/HT/HT(4,4),P(4,4),N/THR/THR,GDOP
      DOUBLE PRECISION U(4)
C
      N=4
C TRANSPOSE H IN HT
      DO 5 I=1,4
      DO 5 J=1,4
5     HT(I,J)=H(J,I)
C MULTIPLY H*HT=P &INVERT P MATRIX
      CALL MATMUL
      CALL MATI
      GDOP=P(1,1)+P(2,2)+P(3,3)+P(4,4)
      IF(GDOP.GE.0.0)GO TO 7
      WRITE(4,10)HT,P(1,1),P(2,2),P(3,3),P(4,4)
      GDOP=-GDOP
    7 GDOP=SQRT(GDOP)
   10 FORMAT(' H MATRIX IS',/,4(1X,4G13.6,/),/,' TRACE VALUE',/,(G14.6))
      RETURN
      END
```

```
      SUBROUTINE LLA
      DOUBLE PRECISION S(4),LAT,LON,ALT,R,W,S3,RT,T1,T2,R2,CT1
      IMPLICIT REAL*4(A-H,J-Z)
      COMMON/POS/LAT,LON,ALT,R/DTX/S
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE
C
      W=S(2)*S(2)+S(1)*S(1)
      S3=S(3)*S(3)
      R2=S3+W
      R=DSQRT(R2)
      W=DSQRT(W)
      LON=DATAN2(S(2),S(1))*RTD
      AE2=E2*AE
      LAT=S(3)/(OMF2*W)
      T1=1.E0+OMF2*LAT*LAT
      T2=DSQRT(T1)
      T1=T1*T2
      LAT=LAT+T1/(W*T1-AE2)*((AE2*LAT)/T2-W*LAT+S(3))
      LAT=DATAN(LAT)*RTD
      IF(R.GT.6.65E6)GO TO 20
      ALT=R-AE/DSQRT(E2*S3/(OMF2*R2)+1.E0)
      RETURN
   20 T1=DATAN2(S(3),W)
      DO 25 I=1,2
      CT1=DCOS(T1)
      T2=DATAN2(DSIN(T1),OMF2*CT1)-T1
      S3=AE*OMF2*AE/(1.E0-E2*CT1*CT1)
      W=DSIN(T2)
      ALT=DSQRT(R2-S3*W*W)-DSQRT(S3)*DCOS(T2)
      W=ALT*W/R
   25 T1=T1-DATAN2(W,DSQRT(1.-W*W))
      RETURN
      END
```

```
      SUBROUTINE ECEF
      IMPLICIT REAL*4(A-Z)
      DOUBLE PRECISION TU(4),LAT,LON,ALT,XT,SLAT
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE
      COMMON/DTX/TU/POS/LAT,LON,ALT,XY
C
      SLAT=DSIN(LAT)
      XY=AE/DSQRT(1.E0-E2*SLAT*SLAT)
      TU(3)=(XY*OMF2+ALT)*SLAT
      XY=(XY+ALT)*DCOS(LAT)
      TU(1)=XY*DCOS(LON)
      TU(2)=XY*DSIN(LON)
      RETURN
      END
```

```
      SUBROUTINE MATMUL
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),G(4,4),RVIS(8)
      COMMON/HT/H(4,4),A(4,4)
      DOUBLE PRECISION U(4)
C
      DO 10 I=1,4
      DO 10 J=1,4
      TEMP=0.0E0
      DO 5 K=1,4
 5    TEMP=TEMP+H(I,K)*G(K,J)
 10   A(I,J)=TEMP
      RETURN
      END
```

```
C 'MATI' INVERTS AN A MATRIX AND SUBSTITUTES THE
C RESULTS BACK INTO THE A MATRIX.
C
      SUBROUTINE MATI
      COMMON/HT/HT(4,4),A(4,4),N
      DIMENSION INDEX(4,2),IPVOT(4),PIVOT(4)
      DATA ONE/1.0/,ZERO/0.0/
C
      DET=1.0
C  CLEAR IPVOT VECTOR
      DO 10 I=1,4
      IPVOT(I)=0
   10 CONTINUE
C
      DO 135 I=1,N
C FOLLOWING 12 STATEMENTS FOR SEARCH FOR PIVOT ELEMENT
      T=ZERO
      DO 9 J=1,N
      IF(IPVOT(J)-1)13,9,13
 13   DO 23 K=1,N
      IF(IPVOT(K)-1)43,23,81
 43   IF(ABS(T)-ABS(A(J,K)))83,23,23
 83   IROW=J
      ICOL=K
      T=A(J,K)
 23   CONTINUE
 9    CONTINUE
      IPVOT(ICOL)=IPVOT(ICOL)+1
C FOLLOWING 15 STATEMENTS PUT PIVOT ELEMENT ON DIAGONAL
      IF(IROW-ICOL)73,109,73
 73   DET=-DET
      DO 12 L=1,N
      T=A(IROW,L)
      A(IROW,L)=A(ICOL,L)
 12   A(ICOL,L)=T
 109  INDEX(I,1)=IROW
      INDEX(I,2)=ICOL
      PIVOT(I)=A(ICOL,ICOL)
      DET=DET*PIVOT(I)
C FOLLOWING 6 STATEMENTS TO DIVIDE PIVOT ROW BY
C PIVOT ELEMENT
      A(ICOL,ICOL)=ONE
      DO 205 L=1,N
 205  A(ICOL,L)=A(ICOL,L)/PIVOT(I)
C FOLLOWING 10 STATEMENTS TO REDUCE NON-PIVOT ROWS
      DO 135 LI=1,N
      IF(LI-ICOL)21,135,21
 21   T=A(LI,ICOL)
      A(LI,ICOL)=ZERO
      DO 89 L=1,N
 89   A(LI,L)=A(LI,L)-A(ICOL,L)*T
```

```
 135  CONTINUE
C FOLLOWING 11 STATEMENTS TO INTERCHANGE COLUMNS
      DO 3 I=1,N
      L=N-I+1
      IF(INDEX(L,1)-INDEX(L,2))19,3,19
 19   JROW=INDEX(L,1)
      JCOL=INDEX(L,2)
      DO 549 K=1,N
      T=A(K,JROW)
      A(K,JROW)=A(K,JCOL)
      A(K,JCOL)=T
 549  CONTINUE
 3    CONTINUE
 81   RETURN
      END
```

```
      SUBROUTINE SATGEN
      IMPLICIT REAL*4(A-Z)
      INTEGER I,NAC
      DOUBLE PRECISION TK,EK,OGA,RK
      COMMON/POS/TK,EK,OGA,RK/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2
      COMMON/SAT/X,Y,Z,MK,ECC,TOE,INOT,NRATE,SRAD,OMEO,OME,MNOT,R,CINO,
     *SINO,COME,SOME,C
C
      TKC=TC-TOE
      TK=TUY-TCY+TKC
      OGA=OMEO+(NRATE-ERATE)*TK-ERATE*TOE
      MK=MNOT+C*TK
    6 IF(MK.GE.-PI2)GO TO 7
      MK=MK+PI2
      GO TO 6
    7 IF(MK.LE.PI2)GO TO 8
      MK=MK-PI2
      GO TO 7
    8 EK=MK+ECC*SIN(MK)
      DO 10 I=1,4
      COSE=DCOS(EK)
      SINE=DSIN(EK)
      TK=1.-ECC*COSE
   10 EK=EK-(EK-MK-ECC*SINE)/TK
      COSE=(COSE-ECC)/TK
      SINE=SQRT(1.-ECC*ECC)*SINE/TK
      EK=COSE*COME-SINE*SOME
      MK=SINE*COME+COSE*SOME
      SINE=2.*MK*EK
      COSE=2.*EK*EK-1.
      RK=TK*R
      X=RK*EK
      Y=RK*MK
      Z=Y*SINO
      EK=CINO*Y
      SINE=DSIN(OGA)
      COSE=DCOS(OGA)
      Y=X*SINE+EK*COSE
      X=X*COSE-EK*SINE
      RETURN
      END
```

```
      SUBROUTINE SATSEL
      IMPLICIT REAL*4(A-H,L-Z)
      INTEGER PERMUT(8),NAC,NUM,CONSTL(8)
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3)
      DOUBLE PRECISION U(4),LAT,LON,ALT,R
      COMMON/FLAG/IFLG,IFLAG,IPATH/NUM/NUM/THR/THR,TGDOP
      COMMON/AZM/ELV(8),AZM(8)/SAT/S(4),DA(14)
      COMMON/POS/LAT,LON,ALT,R/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE/DB2/AD(14,8),ISNUM(24
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),H(4,4),RVIS(8)
C
      IF(IFLAG.LT.4)GO TO 149
      IF(IFLAG.EQ.4)GO TO 148
C
C  MORE THAN 4 SAT VISBLE, SELECT OPTIMAL GDOP
C
C  CLEAR PERMUT VECTOR
C
      DO 140 I=1,NUM
      PERMUT(I)=0
  140 CONTINUE
C
C  SELECT HIGHEST SAT
C
      PERMUT(1)=1
      J=1
      DO 1412 I=2,NUM
      IF(EL(I) .LT.0.0)GO TO 1412
      J=J+1
      IF(EL(PERMUT(1)) .GE. EL(I))GO TO 1413
      PERMUT(J)=PERMUT(1)
      PERMUT(1)=I
      IF(EL(PERMUT(J)).LT.0.0) J=J-1
      GO TO 1412
 1413 PERMUT(J)=I
 1412 CONTINUE
C
      IF(IFLAG.EQ.J) GO TO 1414
      WRITE(5,660)NUM,J,IFLAG,PERMUT
C
C  SELECT BEST GDOP
C
 1414 DO 142 J=1,4
  142 R(1,J)=D(PERMUT(1),J)
      GDOP=1.0E10
      CONSTL(1)=PERMUT(1)
      IL=IFLAG-2
      DO 1429 I=2,IL
      IIF=I+1
      IIL=IFLAG-1
      DO 1428 II = IIF,IIL
      IIIF=II+1
      DO 1427 III=IIIF,IFLAG
      DO 1426 J=1,4
      R(2,J)=D(PERMUT(I),J)
      R(3,J)=D(PERMUT(II),J)
 1426 R(4,J)=D(PERMUT(III),J)
      CALL SGDOP
```

```
C      WRITE(6,750)PERMUT(1),PERMUT(),PERMUT(I),PERMUT(III),TGDOP
C
       IF(GDOP.LE.TGDOP) GO TO 1427
       GDOP=TGDOP
       CONSTL(2)=PERMUT(I)
       CONSTL(3)=PERMUT(II)
       CONSTL(4)=PERMUT(III)
 1427  CONTINUE
 1428  CONTINUE
 1429  CONTINUE
       J=4
       GO TO 1495
C
C  ONLY 4 SAT VISIBLE, SELECT THE 4 AND FIND THEIR GDOP
C
  148  JJ=0
       DO 1485 I=1,NUM
       IF(EL(I).LT.0.0)GO TO 1485
       JJ=JJ+1
       DO 1484 J=1,4
 1484  R(JJ,J)=D(I,J)
       PERMUT(JJ)=I
 1485  CONTINUE
C
C  FIND HIGHEST SAT - MAKE OUTPUT UNIFORM
C
       CONSTL(1)=PERMUT(1)
       DO 1488 I=2,4
       IF(EL(CONSTL(1)) .GE. EL(PERMUT(I)))GO TO 1487
       CONSTL(I)=CONSTL(1)
       CONSTL(1)=PERMUT(I)
       GO TO 1488
 1487  CONSTL(I)=PERMUT(I)
 1488  CONTINUE
       IF(IFLAG.EQ.JJ) GO TO 1486
       WRITE(5,670)NUM,JJ,IFLAG,PERMUT
 1486  CALL SGDOP
       GDOP=TGDOP
       GO TO 1495
  149  IF(IFLAG.EQ.0) GO TO 1496
       J=1
       DO 1491 I=1,NUM
       IF(EL(I).LT.0.0)GO TO 1491
       PERMUT(J)=I
       J=J+1
 1491  CONTINUE
C
C  FIND HIGHEST SAT
C
       CONSTL(1)=PERMUT(1)
       IF(IFLAG .EQ. 1)GO TO 1495
       DO 150 I=2,IFLAG
       IF(EL(CONSTL(1)) .GE. EL(PERMUT(I)))GO TO 1501
       CONSTL(I)=CONSTL(1)
       CONSTL(1)=PERMUT(I)
       GO TO 150
 1501  CONSTL(I)=PERMUT(I)
  150  CONTINUE
 1495  CALL HOURS
       WRITE(6,565)THR,((EL(I),AZM(I)),I=1,NUM)
       IF(IFLAG-4)1497,1499,1498
 1497  WRITE(6,655) (ISNUM(CONSTL(I)),)=1,IFLAG)
C 1497 WRITE(6,655)(CONSTL(I),I=1,IFLAG)
```

```
 1498 IFLAG=4
 1499 WRITE(6,650) (ISNUM(CONSTL(I)),I=1,IFLAG),GDOF
C 1499 WRITE(6,650)(CONSTL(I),I=1,IFLAG),GDOF
 1496 RETURN
  565 FORMAT(/,2X,F7.4,6X,7(F6.2,1X,F6.2,3X))
  650 FORMAT(2X,4(I3,2X),F8.2)
  655 FORMAT(2X,4(I3,2X))
  660 FORMAT('  ERROR: IFLAG NE J IN LOOP 1413',/,' NUM, J, IFLAG',/,3I3
     *,/,' VISIBLE SATS',8I3)
  670 FORMAT('  ERROR: IFLAG NE JJ IN LOOP 1485')
  750 FORMAT(2X,'SAT PERMUTATIONS',3X,4(I3,2X),F8.2)
      END
```

```
      SUBROUTINE VISBLE
      IMPLICIT REAL*4(A-H,K-Z)
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),Y(8,3)
      INTEGER NAC,NUM,K,N,ONE
      LOGICAL SINGL,FIX
      COMMON/VIS/SINGL,ONE,FIX/NUM/NUM/SAT/S(4),DA(14)
      DOUBLE PRECISION LAT,LON,ALT,R,U(4)
      COMMON/AZM/ELV(8),AZM(8)/FLAG/IFLG,IFLAG,IPATH
      COMMON/POS/LAT,LON,ALT,R/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE/DB2/AD(14,8)
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),H(4,4),RVIS(8)
C
C  THE LOGICAL VARIABLE SINGL INDICATES THE NUMBER OF SAT POSITIONS
C    TO BE CALCULATED; FALSE=ALL SATS, TRUE=1 SAT - THE SAT
C    NUMBER GIVEN BY THE VARIABLE ONE
C
      IFLG=IFLAG
      K=1
      N=NUM
      IFLAG=0
      TUY=DTU+AMOD(T,86400.0)
      IF(FIX .AND. T .GE. 86400.0)TUY=TUY+86400.0
      IF(.NOT. SINGL)GO TO 10
      K=ONE
      N=ONE
   10 DO 14 I=K,N
      DO 11 J=1,14
   11 DA(J)=AD(J,I)
      CALL SATGEN
      CALL RANGE(I)
      EL(I)=0.
      DO 12 J=1,3
      D(I,J)=-D(I,J)/(RBAR(I)-U(4))
   12 EL(I)=EL(I)+DIRCU(J)*D(I,J)
      EL(I)=ACOS(EL(I))
      EL(I)=90.-EL(I)
      IF(EL(I).GE.0.0)IFLAG=IFLAG+1
      IF(IPATH.NE.2)GO TO 14
      DO 13 II=1,3
      Y(I,II)=0.0
      DO 13 III=1,3
   13 Y(I,II)=Y(I,II)+S(III)*E(II,III)
      Y(I,3)=Y(I,3)-AE
      SY=SQRT(Y(I,1)*Y(I,1)+Y(I,2)*Y(I,2))
      ELV(I)=ATAN2(Y(I,3),SY)*RTD
      AZM(I)=ATAN2(Y(I,1),Y(I,2))*RTD
      IF(AZM(I).LT.0.)AZM(I)=AZM(I)+360.0
   14 CONTINUE
      IF(SINGL)IFLAG=IFLG
      RETURN
      END
```

```
C
C    ROUTINE TO CONVERT SECONDS TO HOURS MIN SEC
       SUBROUTINE HOURS
       COMMON/THR/THR/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,TMIN,FDTU
       INTEGER*4 HR,MIN,SEC
C
       THR=(TMIN/3600.0)+.00005
       HR=INT(THR)
       THR=(THR-HR)*60
       MIN=INT(THR)
       THR=(THR-MIN)*60
       SEC=INT(THR)
       THR=HR+(MIN*.01)+(SEC*.0001)+.00005
       RETURN
       END
```

```
C   ROUTINE FOR USE IN DETERMINING AZIMUTH
C
      SUBROUTINE ENCOSE
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),DT(8,3)
      IMPLICIT REAL*4(A-H,K-Z)
      DOUBLE PRECISION LAT,LON,ALT,R
      COMMON/POS/LAT,LON,ALT,R
C
      SN=DSIN(LON)
      CN=DCOS(LON)
      ST=DSIN(LAT)
      CT=DCOS(LAT)
      E(1,1)=-SN
      E(1,2)=CN
      E(1,3)=0.0
      E(2,1)=-CN*ST
      E(2,2)=-SN*ST
      E(2,3)=CT
      E(3,1)=CN*CT
      E(3,2)=SN*CT
      E(3,3)=ST
      RETURN
      END
```

```
C  ROUTINE TO DETERMINE UP TIME AND DOWN TIME OF 4 SAT
C
      SUBROUTINE VISTIM
      DOUBLE PRECISION LAT,LON,ALT,R,U(4)
      COMMON/NUM/NUM/FLAG/IFLG,IFLAG,IPATH
      COMMON/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),DT(8,3)
      COMMON/LINR/T1T2(2),TEMP(2,8),RISE,TR(2),TWO,SIGN,IPNTR,
     *THREE,FOUR,DECT(2)/TIME/TDEL,TFIN,TSKIP
      COMMON/PTX/U,D(8,4),RBAR(8),RM(8),DELR(8),R(4,4),RVIS(8)
      COMMON/VIS/SINGL,ONE/POS/LAT,LON,ALT,R
      LOGICAL SIGN(2,8),RISE,TWO,SINGL,THREE,FOUR
      INTEGER ONE,OCUR(2)
      REAL TIM(5,5)
      EQUIVALENCE (TIM(1,1),DATE),(TIM(2,1),T1),(TIM(3,1),T2)
C
C  THE LOGICAL VARIABLES ARE USED IN THE FOLLOWING MANNER:
C          SIGN - AN INDICATOR OF THE SIGN OF THE SAT ELEVATION
C                 - FALSE=NEGATIVE, TRUE=POSITIVE
C          RISE - AN INDICATOR OF WHAT CONDITION THE INTERPOLATION
C                 SUBROUTINE SHOULD BE LOOKING FOR
C                 - FALSE=A FALLING CONDITION, A POSITIVE TO NEGATIV
C                         TRANSITION
C                 - TRUE=A RISING CONDITION, A NEGATIVE TO POSITIVE
C                         TRANSITION
C          TWO - AN INDICATOR THAT TWO RISE OR FALL CONDITIONS OF
C                INTEREST HAPPENED IN THE SAME ONE HOUR INTERVAL
C          SINGL - AN INDICATOR TO THE VISBLE SUBROUTINE AS TO THE
C                  NUMBER OF SAT POSITIONS NEEDED TO BE CALCULATED
C                 - TRUE=ONE SAT POSITION, THE SAT POSITION WANTED
C                         IS GIVEN BY THE VALUE OF THE VARIABLE ONE
C                 - FALSE=ALL THE SAT POSITIONS
C          THREE - AN INDICATOR THAT A RISE OF IMPORTANCE OCCURED
C                  FROM A CONDITION OF ONE UP - THROW AWAY FIRST
C                  TIME VALUE FOUND
C          FOUR - AN INDICATOR THAT A RISE OF CONCERN OCCURED FROM
C                 A CONDITION OF ZERO UP - THROW AWAY FIRST TWO
C                 TIME VALUES FOUND
C
      TSKIP=TSKIP*TDEL
      IFLG=0
      DATE=(DTU/86400.0)+1.0
      WRITE(6,1)DATE,FDTU
      WRITE(6,2)LAT,LON
      WRITE(6,3)
      WRITE(6,4)
      SINGL=.FALSE.
   30 THREE=.FALSE.
      FOUR=.FALSE.
      RISE=.TRUE.
      TWO=.FALSE.
C
C CLEAR TIM MATRIX
C ONLY CLEAR VARIABLES THAT WILL BE TESTED LATER
C
      DO 90 I=2,5
      TIM(1,2)=0.0
```

```
      OCUR(2)=1
      IPNTR=1
      IF(3-IFLG)156,105,100
  100 CALL VISBLE
      CALL INSERT
      IF(3-IFLAG)149,110,210
  105 TIM(1,2)=1000000.0
      T1=1000000.0
      GO TO 185
  110 TIM(1,2)=1000000.0
      T1=1000000.0
      GO TO 180
C
C  THERE ARE 3 SATS UP
C
  120 T=T+TDEL
      IF(T .GT.86400.)GO TO 230
      CALL VISBLE
      IF(3-IFLAG)140,130,190
  130 CALL INSERT
      GO TO 120
C
C  RISE-4
C
  140 IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      CALL INSERT
      RISE=.TRUE.
      CALL INTERP
      TIM(OCUR(2),3)=TH(1)
  146 TWO=.FALSE.
      T=T+TSKIP
      IF(T .GT. 86400.0)GO TO 230
      CALL VISBLE
      IF(IFLAG .GE. 4)GO TO  155
      CALL INSERT
      IF(IFLAG .LT. 3)TWO=.TRUE.
  147 T=T-TDEL
      CALL VISBLE
      IF(4-IFLAG)148,148,147
  148 IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      CALL INSERT
      IF(IFLAG .GT. 4)THREE=.TRUE.
      IF(IFLAG .GT. 5)FOUR=.TRUE.
      RISE=.FALSE.
      IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      GO TO 161
C
  149 TIM(1,3)=1000000.0
      TIM(1,2)=1000000.0
      T1=1000000.0
  150 T=T+TDEL
      IF(T .GT. 86400.0)GO TO 230
      CALL VISBLE
      IF(IFLAG .LT. 4)GO TO 160
  155 CALL INSERT
      GO TO 150
  156 TIM(1,2)=1000000.0
      TIM(1,2)=1000000.0
      T1=1000000.0
      CALL VISBLE
```

```
      C
  160 IF(IFLAG .LT. 3)TWO=.TRUE.
      IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      RISE=.FALSE.
      CALL INSERT
  161 CALL INTERP
      TIM(OCUR(2),4)=TH(1)
      OCUR(2)=OCUR(2)+1
      IF(.NOT. TWO)GO TO 180
      C
      C  FALL-3 HAS OCCURED IN THE SAME HOUR INTERVAL AS FALL-4
      C
      TIM(OCUR(1),5)=TH(2)
      T2=DECT(2)
      GO TO 205
      C
      C  LOOK FOR FALL-3 OR RISE-4
      C
  180 T=T+IDEL
      IF(T .GT. 86400.0)GO TO 230
  185 CALL VISBLE
      IF(3-IFLAG)140,130,190
      C
      C  FALL-3
      C
  190 IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      RISE=.FALSE.
      CALL INSERT
      CALL INTERP
      TIM(OCUR(1),5)=TH(1)
      T2=DECT(1)
  205 TWO=.FALSE.
      T1=T2-T1
      IF(T1 .LT. 0.0)T1=-T1
      IF(T1 .GT. 3600.0)GO TO 206
      TIM(OCUR(1),2)=0.0
      TIM(OCUR(1),5)=0.0
      GO TO 207
  206 OCUR(1)=OCUR(1)+1
  207 IF(IFLG .LT. 2)GO TO 225
      GO TO 212
      C
      C  RISE-3
      C
  210 IF(IFLAG .LT. 2)GO TO 225
  212 T=T+IDEL
      IF(T .GT. 86400.0)GO TO 230
      CALL VISBLE
      IF(2-IFLAG)220,211,224
  211 CALL INSERT
      GO TO 212
  220 RISE=.TRUE.
      IF(IPNTR .EQ. 2)IPNTR=0
      IPNTR=IPNTR+1
      IF(IFLAG .GT. 3)TWO=.TRUE.
      CALL INSERT
      CALL INTERP
      TIM(OCUR(1),2)=TH(1)
      T1=DECT(1)
      IF(.NOT. TWO)GO TO 180
      C
```

```
      GO TO 146
C
  224 CALL INSERT
C
C  LESS THAN 2 SATS UP
C
  225 T=T+TDEL
      IF(T .GT. 86400.0)GO TO 230
      CALL VISBLE
      IF(IFLAG .EQ. 0)FOUR=.TRUE.
      IF(2-IFLAG)227,228,226
  226 CALL INSERT
      IF(IFLAG .NE. 0)FOUR=.FALSE.
      GO TO 225
  227 THREE=.TRUE.
      GO TO 220
  228 FOUR=.FALSE.
      CALL INSERT
      GO TO 212
C
C  NEW DAY
C
  230 IFLG=IFLAG
      IF(IFLG .GE. 4)TIM(OCUR(2),4)=1000000.
      IF(IFLG .GE. 3)TIM(OCUR(1),5)=1000000.
      N=1
      IF(OCUR(1) .EQ. 1 .AND. OCUR(2) .EQ. 1)GO TO 240
      N=OCUR(1)
      IF(OCUR(2) .GT. OCUR(1))N=OCUR(2)
      IF(N .EQ. 6)N=5
      IF(TIM(N,2) .EQ. 0.0 .AND. TIM(N,3) .EQ.0.0)N=N-1
  240 WRITE(6,5)T,TIM(1,1),TIM(1,2),TIM(1,3),TIM(1,4),TIM(1,5)
      IF(N .GT. 1)GO TO 260
  250 DTU=DTU+86400.0
      IF(ABS(DATE-FDTU).LT..5)RETURN
      DATE=DATE+1.0
      T=0.
      GO TO 30
C
C  LOGIC FOR OUTPUT FORMAT
C
  260 DO 300 I=2,N
      IF(TIM(I,2) .NE. 0.0 .AND. TIM(I,3) .NE. 0.0)WRITE(6,7)TIM(I,2),
     *TIM(I,3),TIM(I,4),TIM(I,5)
C
      IF(TIM(I,2) .NE. 0.0 .AND. TIM(I,3) .EQ. 0.0)WRITE(6,8)TIM(I,2),
     *TIM(I,5)
C
      IF(TIM(I,2) .EQ. 0.0 .AND. TIM(I,3) .NE. 0.0)WRITE(6,9)TIM(I,3),
     *TIM(I,4)
C
  300 CONTINUE
      GO TO 250
C
    1 FORMAT(5X,'APPROX RISE AND FALL TIMES FROM',F6.0,' TO',F6.0)
    4 FORMAT('0',4X,'DATE',7X,'RISE-3',8X,'RISE-4',8X,
     *'FALL-4',8X,'FALL-3',/)
    2 FORMAT(1H0,4X,'FOR LAT: ',F8.2,' LON: ',F8.2)
    3 FORMAT(1H0,19X,'VISIBLE AT 0 DEGREE HORIZON')
    5 FORMAT(/,4X,F6.2,4X,F8.2,6X,F8.2,6X,F8.2,6X,F8.2)
    7 FORMAT(14X,F8.2,6X,F8.2,6X,F8.2,6X,F8.2)
    8 FORMAT(14X,F8.2,34X,F8.2)
    9 FORMAT(28X,F8.2,4X,F8.2)
```

```
C
C  ROUTINE TO INSERT THE NEW SAT DATA INTO THE WORK MATRIX
C
      SUBROUTINE INSERT
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),DT(8,3)
      COMMON/LINR/T1T2(2),TEMP(2,8),RISE,TH(2),TWO,SIGN,IPNTR,
     *THREE/NUM/NUM
      COMMON/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      LOGICAL SIGN(2,8),RISE,TWO,THREE
C
      DO 100 I=1,NUM
      SIGN(IPNTR,I)=.FALSE.
      IF(EL(I) .GE. 0.0)SIGN(IPNTR,I)=.TRUE.
      TEMP(IPNTR,I)=EL(I)
  100 CONTINUE
      T1T2(IPNTR)=T
      RETURN
      END
```

```
C
C     ROUTINE TO LINERALY INTERPOLATE THE RISE AND FALL TIMES OF SATS
C
      SUBROUTINE INTERP
      COMMON/NUM/NUM/FLAG/IFLG,IFLAG,IPATH
      COMMON/NAC/NAC,TUY,TCY,TKC,DTC,DTU,TC,T,FDTU
      COMMON/POS/LAT,LON,ALT,R/TIME/TDEL,TFIN,TSKIP
      COMMON/ELEV/EL(8),DIRCU(3),DIR(3),E(3,3),DT(8,3)
      COMMON/LINR/T1T2(2),TEMP(2,8),RISE,TH(2),TWO,SIGN,IPNTR,
     *THREE,FOUR,DECT(2)/THR/THR/VIS/SINGL,ONE
      COMMON/DTX/U,D(8,4),RBAR(8),RM(8),DELR(8),H(4,4),RVIS(8)
      LOGICAL SIGN(2,8),RISE,TWO,SINGL,SGN(3,8),THREE,FOUR
      INTEGER ONE,SATNUM(8)
      REAL WORK(4,8),TIMES(8)
      DOUBLE PRECISION LAT,LON,ALT,R,U(4)
C
      IFLG=IFLAG
      KOUNT=0
      IP=1
      IF(T1T2(2) .GT. T1T2(1))IP=2
      DO 100 I=1,NUM
      IF((SIGN(1,I) .AND. SIGN(2,I)) .OR. (.NOT. SIGN(1,I) .AND. .NOT.
     *SIGN(2,I)))GO TO 100
      IF(RISE .AND. .NOT. SIGN(IP,I))GO TO 100
      IF(.NOT. RISE .AND. SIGN(IP,I))GO TO 100
      KOUNT=KOUNT+1
      SATNUM(KOUNT)=I
      WORK(1,KOUNT)=TEMP(1,I)
      WORK(2,KOUNT)=TEMP(2,I)
      SGN(1,KOUNT)=SIGN(1,I)
      SGN(2,KOUNT)=SIGN(2,I)
      SGN(3,KOUNT)=.FALSE.
      WORK(3,KOUNT)=T1T2(1)
      WORK(4,KOUNT)=T1T2(2)
  100 CONTINUE
      SINGL=.TRUE.
      L=KOUNT+1
      IF(L .GT.6)GO TO 300
      DO 200 I=L,8
      TIMES(I)=100000.0
      SATNUM(I)=0
  200 CONTINUE
  300 DO 600 I=1,KOUNT
      M=0
      IF(.NOT. RISE)GO TO 320
      IF(WORK(IPNTR,I) .LT. 0.8 .AND. WORK(IPNTR,I) .GE. 0.0)GO TO 310
      GO TO 350
  310 TIMES(I)=WORK(IPNTR+2,I)
      GO TO 600
  320 IF(WORK(IPNTR,I) .GT. -0.8 .AND. WORK(IPNTR,I) .LE. 0.0)GO TO 310
  350 DO 500 K=1,3
```

```
      ONE=SATNUM(I)
      IF(M .NE. 1)GO TO 360
      M=0
      IP=3
      IF(WORK(4,I) .GT. WORK(3,I))IP=4
      W1=WORK(1,I)
      W2=WORK(2,I)
      IF(W1 .LT. 0.0)W1=-W1
      IF(W2 .LT. 0.0)W2=-W2
      FUDGE=30.0
      IF(IP .EQ. 4)FUDGE=-FUDGE
      IF(W1 .GT. W2)T=WORK(4,I)+FUDGE
      IF(W1 .LT. W2)T=WORK(3,I)-FUDGE
      GO TO 380
  360 T=(WORK(2,I)*WORK(3,I)-WORK(1,I)*WORK(4,I))/(WORK(2,I)-WORK(1,I))
  380 CALL VISBLE
      L=1
      SGN(3,I)=.FALSE.
      IF(EL(SATNUM(I)) .GE. 0.0)SGN(3,I)=.TRUE.
      IF((SGN(2,I) .AND. SGN(3,I)) .OR. (.NOT. SGN(2,I) .AND. .NOT.
     *SGN(3,I)))L=2
      LL=L+2
      WORK(L,I)=EL(SATNUM(I))
      WORK(LL,I)=T
      A=WORK(3,I)-WORK(4,I)
      IF(A .LT.0.0)A=-A
      SGN(L,I)=SGN(3,I)
      IF(.NOT. RISE)GO TO 400
      IF((A .LE. 30.0 .OR. EL(ONE) .LT. 0.8) .AND. EL(SATNUM(I)) .GE.
     *0.0)GO TO 575
      GO TO  500
  400 IF((A .LE. 30.0 .OR. EL(ONE) .GT. -0.8) .AND. EL(SATNUM(I)) .LE.
     *0.0)GO TO 575
  500 CONTINUE
      M=M+1
      IF(.NOT. RISE)GO TO 550
      IF(A .GT. 30.0 .OR. EL(SATNUM(I)) .LT. 0.0)GO TO 350
      GO TO 575
  550 IF(A .GT. 30.0 .OR. EL(SATNUM(I)) .GT. 0.0)GO TO 350
  575 TIMES(I)=T
  600 CONTINUE
      SINGL=.FALSE.
      N=0
  650 N=N+1
  660 T=TIMES(1)
      K=1
      IF(KOUNT .EQ.1)GO TO 800
      DO 700 I=2,KOUNT
      IF(T .LT. TIMES(I))GO TO 700
      K=I
      T=TIMES(I)
```

```
700 CONTINUE
800 DECT(N)=T
    CALL HOURS
    TH(N)=THR
    IF(.NOT. THREE)GO TO 850
    TIMES(K)=100000.0
    THREE=.FALSE.
    GO TO 660
850 IF(.NOT. FOUR)GO TO 950
    TIMES(K)=100000.0
    FOUR=.FALSE.
    GO TO 660
950 T=T1T2(IPNTR)
    IF(N .EQ. 2 .OR. .NOT. TWO)RETURN
    TIMES(K)=100000.0
    GO TO 650
    END
```