WASHINGTON UNIVERSITY

MEMORANDUM NO. 72/4

OCTOBER, 1972

COMPUTER PROGRAMS FOR PLOTTING SPOT-BEAM COVERAGES FROM AN EARTH-SYNCHRONOUS SATELLITE AND EARTH-STATION ANTENNA ELEVATION ANGLE CONTOURS

THOMAS W. STAGL
JAI P. SINGH

WASHINGTON UNIVERSITY / ST. LOUIS / MISSOURI 63130

42P

PROGRAM ON APPLICATION OF COMMUNICATIONS SATELLITES
TO EDUCATIONAL DEVELOPMENT

Center for Development Technology
Washington University

Memorandum No. 72/4 October, 1972

COMPUTER PROGRAMS FOR PLOTTING SPOT-BEAM COVERAGES FROM AN EARTH-SYNCHRONOUS SATELLITE AND EARTH-STATION ANTENNA ELEVATION ANGLE CONTOURS

THOMAS W. STAGL
JAI P. SINGH

This research is supported by the National Aeronautics and Space Administration under Grant No. Y/NGL-26-008-054. This memorandum is primarily for internal distribution and does not necessarily represent the views of either the research team as a whole or the National Aeronautics and Space Administration.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

This memorandum* contains a description and listings of computer programs for plotting geographical and political features of the world or a specified portion of it, for plotting spot-beam coverages from an earth-synchronous satellite over the computer generated mass, and for plotting polar perspective views of the earth and earth-station antenna elevation contours for a given satellite location. The programs have been prepared in connection with a project on Application of Communication Satellites to Educational Development. This report is intended to be a continuation of the work earlier reported by Stagl and Singh.(1)

A data tape was obtained from the Rand Corporation containing a digitized map of the world. Several programs have been written utilizing this data tape. The first program, called WORLDMAP, simply plots the tape as a map with axes and grid lines. The second program, called MINMAP, plots a specified part of the total map so that larger scale plots of a small area of interest can be obtained. The program described in Memorandum 72/3(1) is used as an overlay on these two plots. This program plots the locus of intersection of quadric cones (narrow-beams from satellites) and a sphere (the earth).

A third program utilizing this data tape, called PERSPECT, plots a polar perspective view of the earth and earth-station antenna elevation angle contours. This is a view of the earth as seen from a satellite in the geosynchronous orbit. Using a plot of this kind, one can make an overlay of the cross-sectional shape of the desired antenna beam and this overlay will be valid over the entire plot of the earth. This program is intended to facilitate determination of the area coverage by satellite-borne shaped beam antennas.

---

*The authors wish to acknowledge the assistance of Mr. Neil Ostrander of the Rand Corporation, Santa Monica in acquiring the computer tape containing the digitized map of the world. They also wish to thank Mrs. Emily S. Pearce and Ms. Donna Barnes for typing the various drafts and the final version of the manuscript.

Also included in this memorandum is a modified antenna coverage program to be used to plot the footprints produced by the two off-axis S-band ETV feeds on Applications Technology Satellite-F (ATS-F). This program can also be used as an overlay for WORLDMAP and MINMAP.

II. DATA TAPE

The point coordinates are stored in strings; latitude, longitude, latitude, longitude. . . . 0.0, 0.0. The range in latitude is from $-\pi/2$ to $\pi/2$ (radian measure), and the longitude range is from $-\pi$ to π. The tape is formatted into 24 blocks. The first block contains 23 integers which specify the number of coordinate pairs in each of the following 23 blocks. Each of the following blocks contain a number of coordinate strings. The strings are of variable length and all strings end with a (0.0, 0.0) coordinate pair. Political boundaries are distinguishable from geographical boundaries in the following way. Signal coordinate pairs appear at the beginning of groups of strings. If the first coordinate pair in a string is (4.0, 0.0) then that string and all following strings are geographical boundaries until a string whose first coordinate pair is (8.0, 0.0) is encountered. This indicates that that string and all following strings are political boundaries until a string whose first coordinate pair is (4.0,0.0) is encountered.

The original tape obtained from the Rand Corporation was an 8 track 1600 bpi tape. Since the IBM 360/50 installation at Washington University Computing Facilities has no facilities for dealing with 1600 bpi tape, the data tape had to be copied onto an 800 bpi (bits per inch) tape which is kept in the Washington University Computing Facilities. A card backup was also obtained in case the tape is lost or inadvertently modified.

III. PLOTTING PROGRAMS

The first program written to plot the data tape, called WORLDMAP, does not take advantage of the ability to distinguish between geographical and political boundaries. This is because the added programming complexity needed to distinguish political boundaries from geographical boundaries on the Calcomp plotter is not warranted by the use of the plot.

Basically the program reads the coordinate strings from the tape and connects the points with straight line segments. The only problem is that the boundary of Siberia "wraps around" the end of the map and those strings that "wrap around" must be broken in order to avoid "retrace" lines across the plot. This is done by checking consecutive points in a string for a separation of over 2 radians. If the separation is greater than 2 radians the points are assumed to "wrap around" the ends of the map. In this case the string is broken there and plotted. The remainder of that string is treated as a new string.

The input to the program, FACT, is a magnifying factor for the map. With an input of FACT = 1.0 a map 50 inches by 26 inches will be drawn. Any number greater or less than 1.0 can be used but one should consider the size of the output plot and the size of the paper available. Figure 1 is a flowchart of the WORLDMAP program. A listing of the program is included in Appendix A.

The second program written for this data tape, called MINMAP, plots only a portion of the total map. The inputs to the program are the upper and lower limits of the longitude and latitude and the magnifying factor as for the WORLDMAP program. For this program, all points of each string are checked against these limits and either accepted or rejected depending on whether or not they are within these boundaries.

The size of this plot is somewhat dependent on the input boundaries. The vertical axis is set at 10 inches. Using this fact and the upper and lower limits on the latitude, a scaling factor is calculated. This scaling factor is then

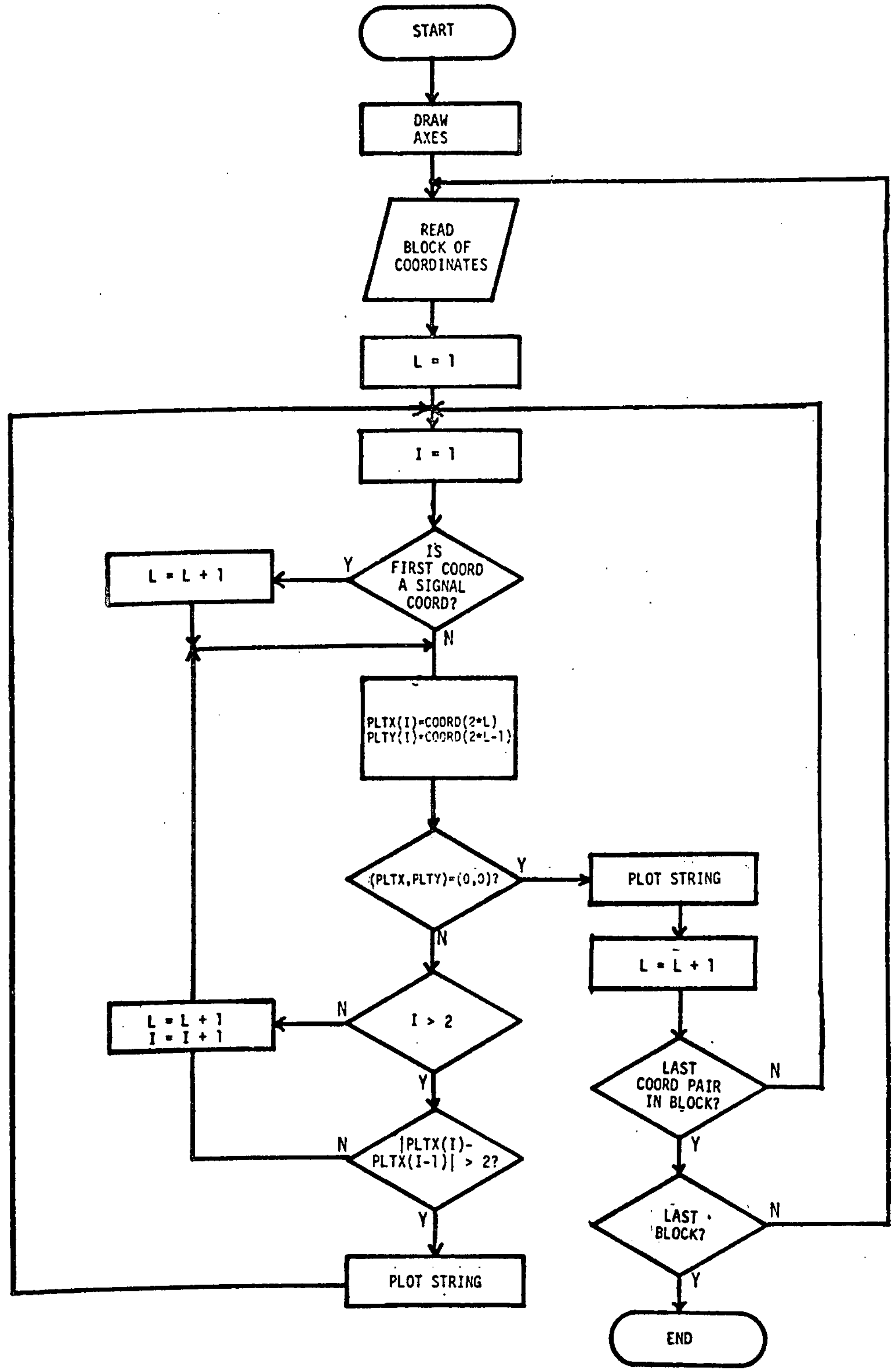

Figure 1. Flowchart of WORLDMAP Program

used, along with the upper and lower limits on the longitude, to calculate the horizontal axis length. As in the WORLDMAP program, the magnifying factor input can be used to change the size of the plot.

The computer prints out the values of the first longitude value on the horizontal axis (FVX), the first value on the latitude axis (FVY), and the scaling increment (DA) in degrees per inch. These values will be needed for proper scaling of the antenna coverage program.

A listing of the program is included in Appendix B. Figure 2 is a flowchart of the MINMAP program.

IV. ANTENNA COVERAGE PATTERNS USING DATA TAPE

The antenna coverage patterns program of Memorandum 72/3 was modified for use with the WORLDMAP program. The modification, called ANTOVLY, includes an elimination of the portion of the program that scales the coordinates and draws the axes, as well as a redefinition of the origin and scaling factor to those of the WORLDMAP program. The modified program is listed in Appendix C. When the programs are run, the map is first plotted. Then the antenna coverage patterns are plotted over the map. The resultant plot shows explicitly the areas covered by the antenna beams.

The inputs to the program are the first longitude value on the horizontal axis, FX, the first latitude value on the vertical axis, FY, the horizontal scaling increment, DX, and the vertical scaling increment, DY. The magnifying factor, FACT, is also input.

When used as an overlay to WORLDMAP, the inputs should be: FX = -180., FY = -90., DX = 7.2, DY = 6.923, for matching the size of ANTOVLY plot with the WORLDMAP. The magnifying factor should be the same as that used when WORLDMAP was run. A sample of this type of output is shown in Figure 3.

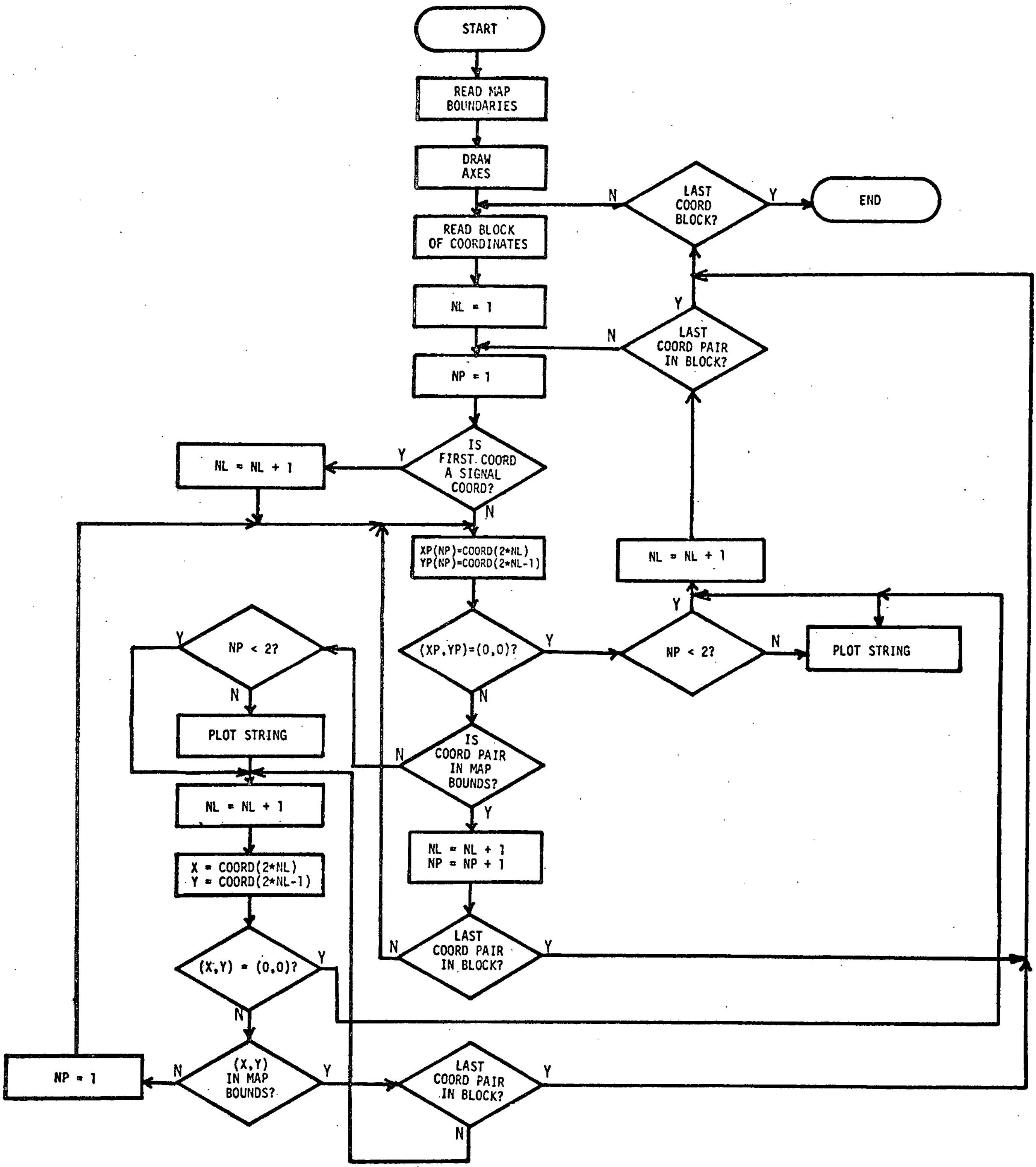

Figure 2. Flowchart of MINMAP Program

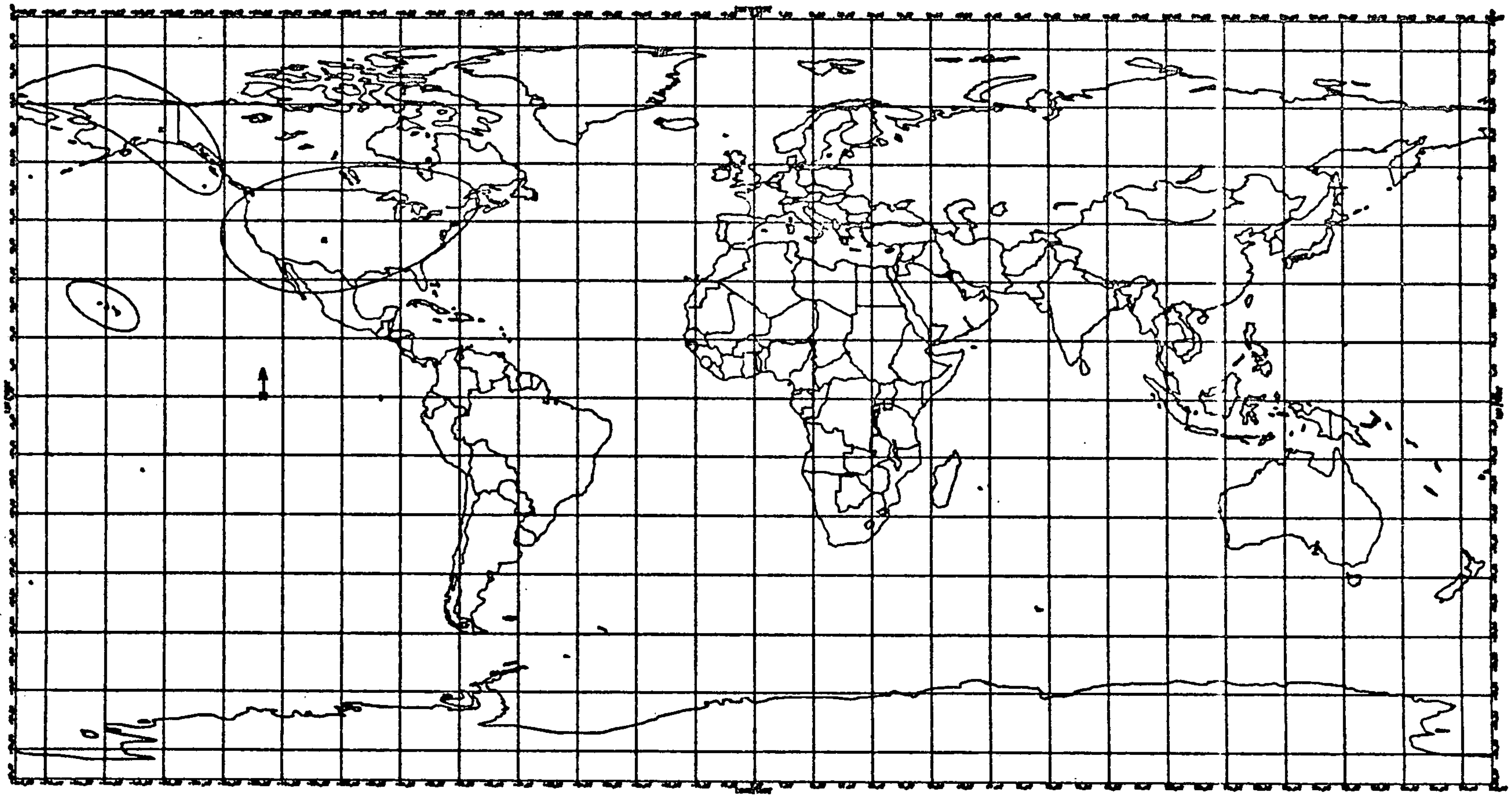

Figure 3. A Typical WORLDMAP Plot With An Antenna Coverage Overlay.
(Subsatellite point = 120°W latitude)

The ANTOVLY program can also be used with the MINMAP program. When used in this way, the inputs should be: FX = FVX (output from MINMAP), FY = FVY (output from MINMAP) and DX = DY = DA (output from MINMAP). The magnifying factor should be the same as that used when MINMAP was run. A sample of this output is shown in Figure 4.

V. POLAR PERSPECTIVE

The third program to use the data tape, called PERSPECT, was written to provide a view of the earth as seen from a satellite in an earth-synchronous orbit and to plot earth-station antenna elevation angle contours. Using a plot of this kind, one can make an overlay of the cross-sectional shape of the desired antenna beam and this overlay would be valid over the entire plot. The angular beam width would be proportional to the linear scale of the plot.

To realize this type of plot, a projection is first made onto a sphere centered at the satellite as shown in Figure 5. The angle of the horizon, RMAX, is first calculated. A coordinate string is then read from the tape and the angle R is calculated for the first point according to the following formula:

$$R = \mathrm{Cos}^{-1}[\mathrm{Sin(LTSS)}\ \mathrm{Sin(YL)} + \mathrm{Cos(LTSS)}\ \mathrm{Cos(YL)}\ \mathrm{Cos(XL)}] \tag{1}$$

where LTSS is the subsatellite latitude, XL and YL are the longitude relative to the subsatellite longitude and the latitude, respectively, of the point in questions. If $|RMAX-R| < 30°$, that is, if the first point of that string is within 30°, of the horizon, then the angle R is calculated for each point of the string according to equation (1) and it is checked for visibility. If $|RMAX-R| \geq 30°$, that is, if the first point of that string is greater than 30° from the horizon, it is checked for visibility. If that point is

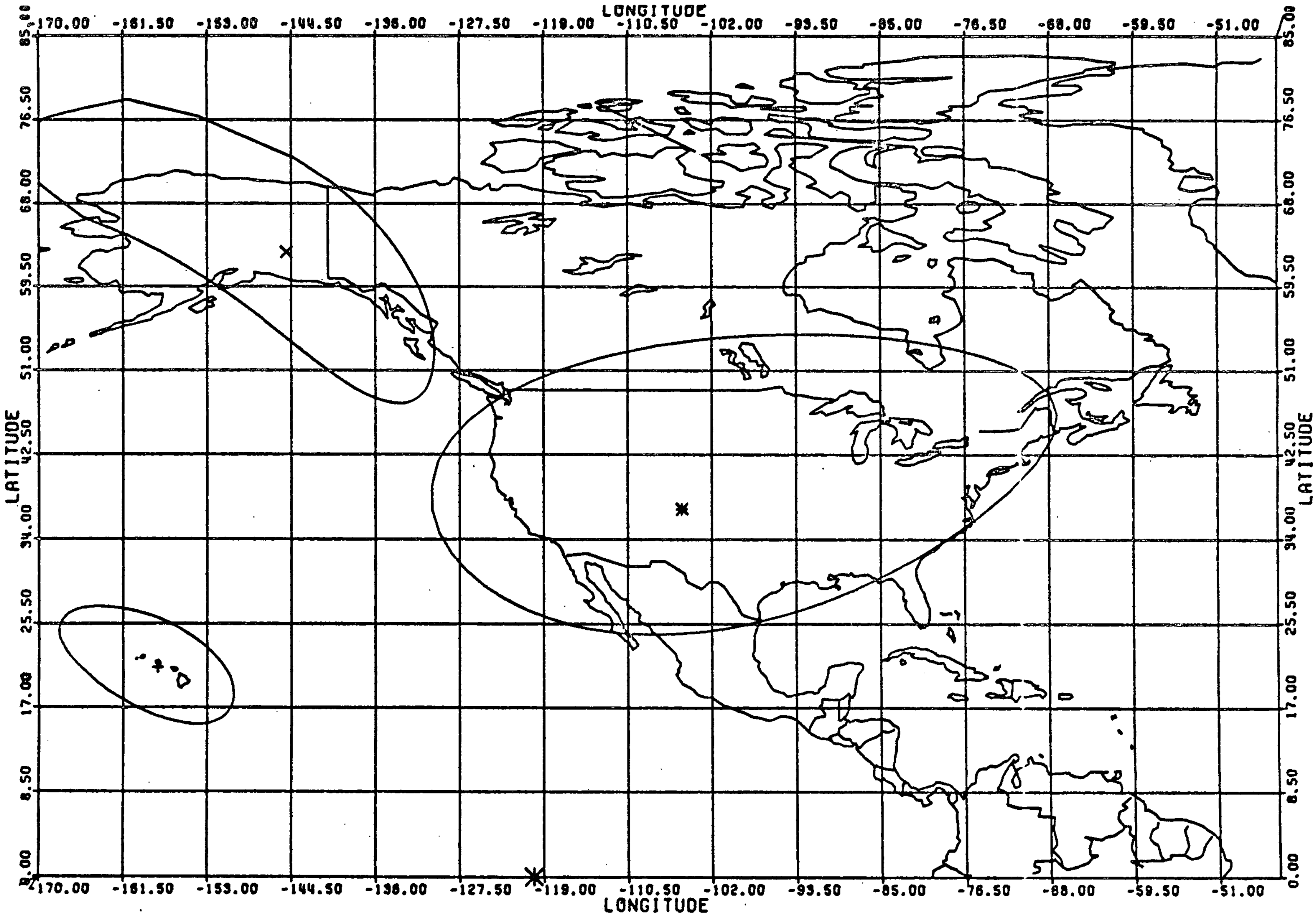

Figure 4. A Typical MINMAP Plot With An Antenna Coverage Overlay.

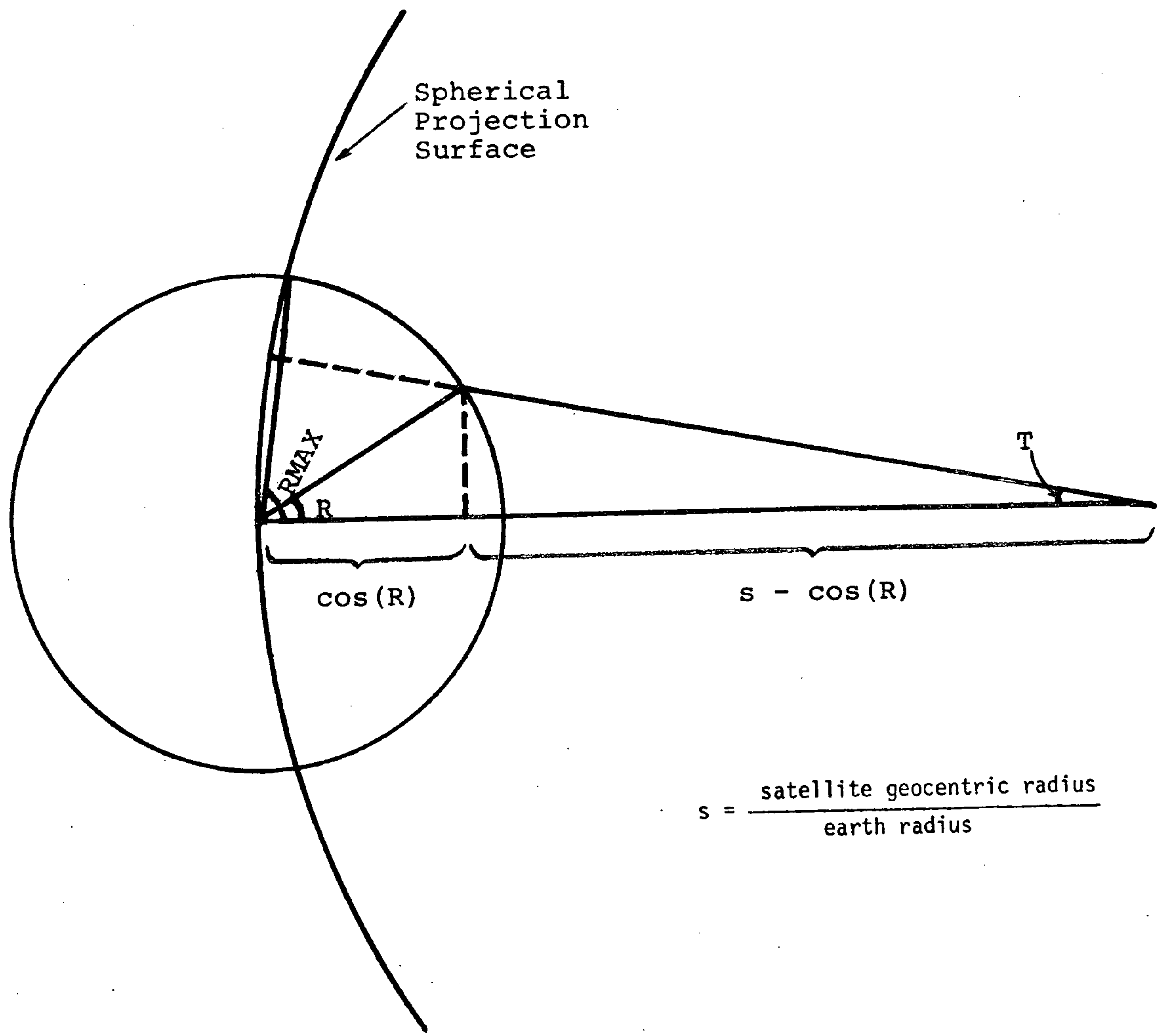

Figure 5. POLAR PERSPECTIVE GEOMETRY
(not to scale)

beyond the horizon then all points of that string are assumed to be beyond the horizon.

For the points that are visible an angle T is calculated according to the following formula:

$$T = \mathrm{Tan}^{-1}\left[\frac{\mathrm{Sin}(R)}{S - \mathrm{Cos}(R)}\right] \tag{2}$$

A polar angle about the subsatellite points is calculated as follows:

$$U = \mathrm{Tan}^{-1}\left[\frac{\mathrm{Cos}(YL)\ \mathrm{Sin}(XL)}{\mathrm{Cos}(LTSS)\ \mathrm{Sin}(YL) - \mathrm{Sin}(LTSS)\ \mathrm{Cos}(YL)\ \mathrm{Cos}(XL)}\right] \tag{3}$$

The Calcomp plotter uses Cartesian coordinates so these polar co-ordinates must be changed to the plotter coordinates, XP, YP, as follows:

$$XP = T \cdot \mathrm{Sin}(U)$$
$$YP = T \cdot \mathrm{Cos}(U) \tag{4}$$

An additional feature of this program is that it will draw the contours of constant elevation on the plot for any set of input elevations.

The only inputs necessary for this program other than the data tape are the subsatellite longitude and latitude, the magnifying factor and the set of elevations for the constant elevation contours. Figure 6 shows a plot of this type with 70°, 45° and 5° elevation contours. Figure 7 is a flowchart of the program.

The program can handle any number of cases in one run. The only limitation is the amount of CPU time specified on the job card. On the IBM 360/50 system a single case takes approximately 45 to 55 seconds, depending on subsatellite location and number of elevation contours to be drawn.

A listing of the program is included in Appendix D.

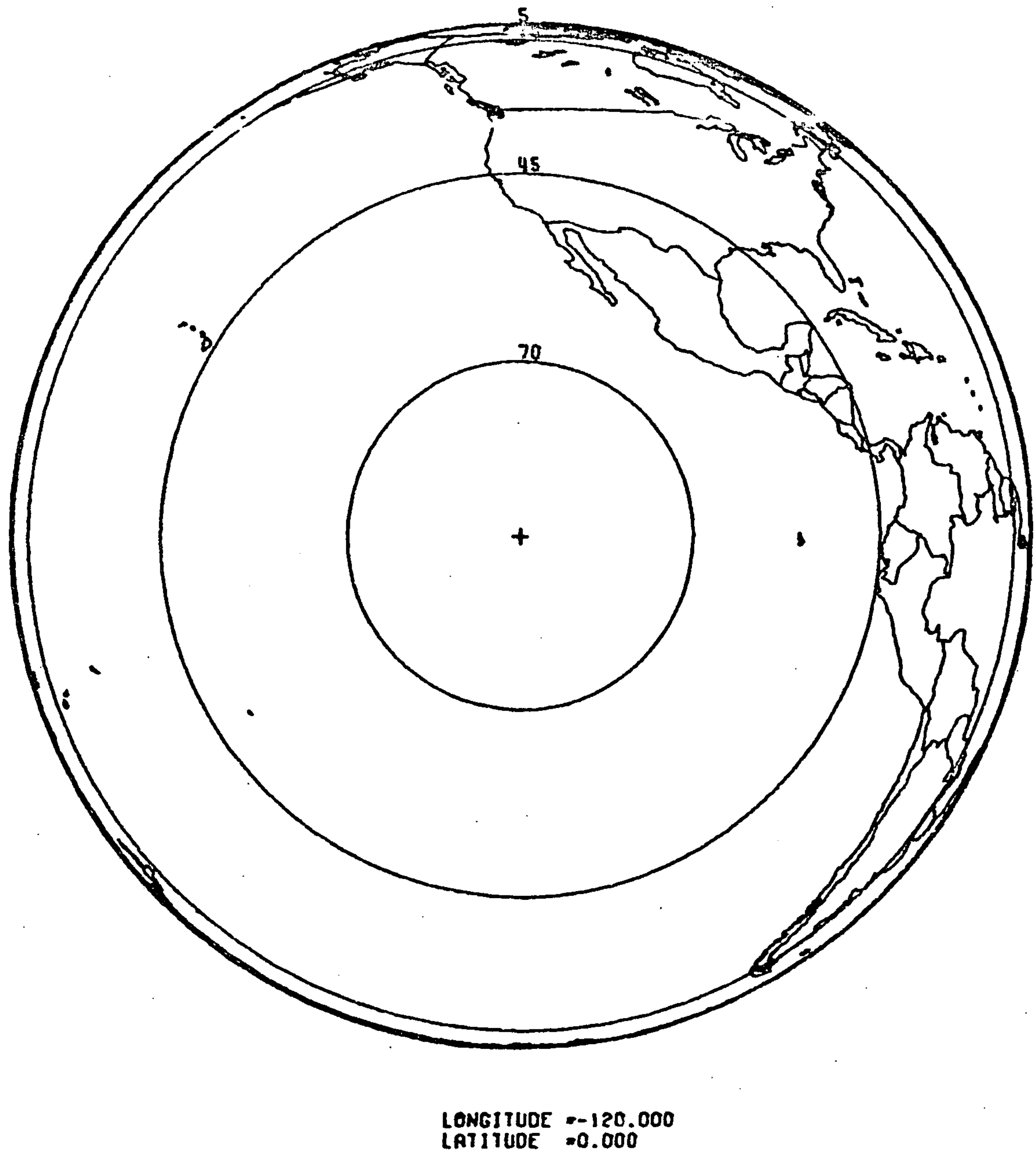

Figure 6. A Polar Perspective of the Earth-From A Satellite at 120°W Longitude and Earth-Station Antenna Elevation Contours.

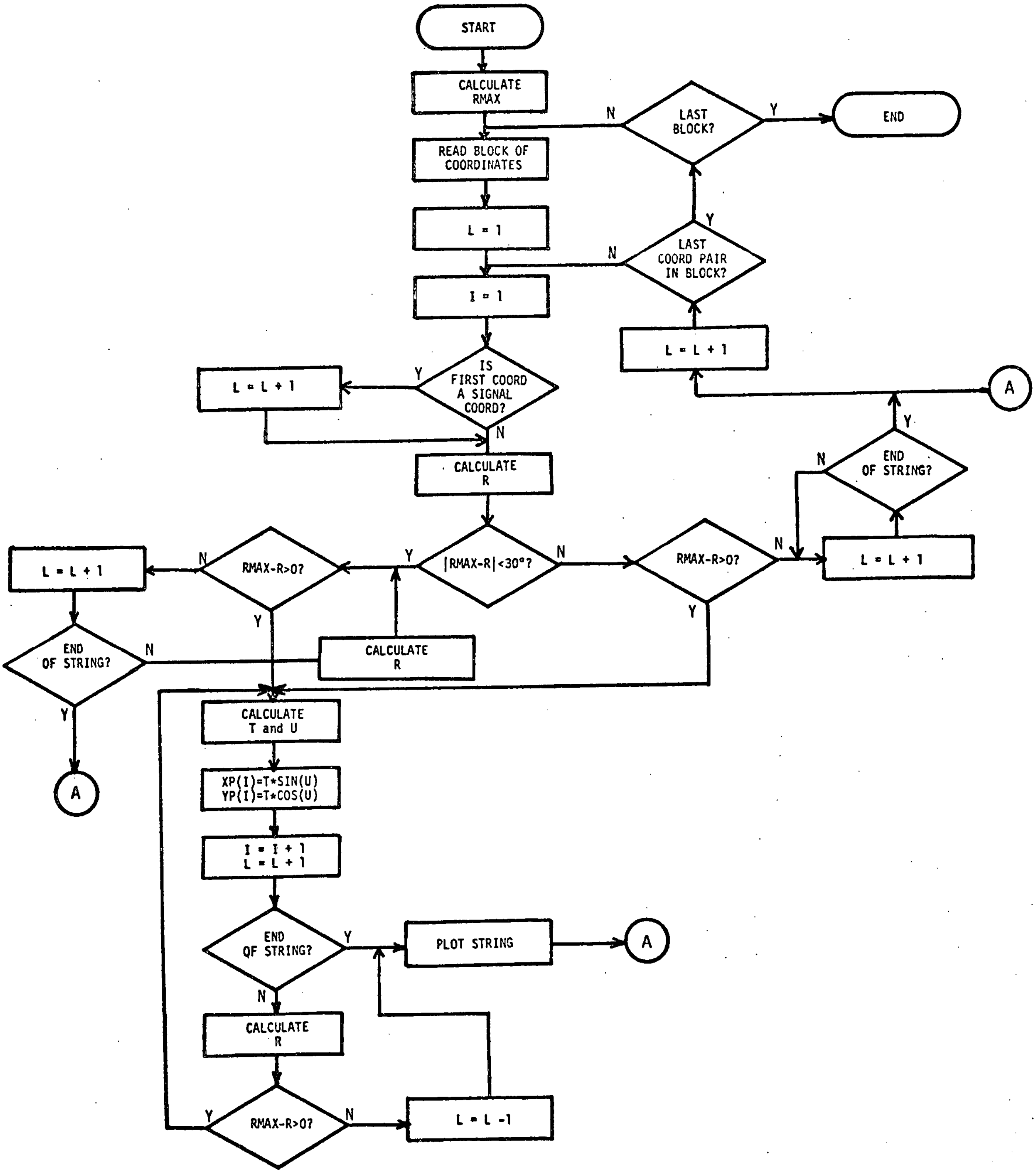

Figure 7. Flowchart for the Polar PERSPECTive Program.

VI. FOOTPRINTS OF ATS-F S-BAND ETV OFF-AXIS DUAL FEEDS

NASA's Applications Technology Satellite-F (ATS-F), scheduled for launch in April, 1974, is going to test the feasibility of the direct delivery of ETV signals to low cost terminals in the newly allocated 2500-2690 MHz band in Alaska, Appalachian region, and the Rocky Mountain States. The antenna coverage program, described earlier in Section IV, has been modified to plot the S-Band footprints produced by the ATS-F spacecraft.

The basic ATS-F spacecraft includes a prime focus feed complex having crossed-feed elements.(2) The two ETV beams are generated by the 30 foot dish onboard the spacecraft from feeds which tie on the satellite North-South axis. Neither feed lies on the antenna boresight, the separation between the boresight and the nearest feed being about 0.7 degree and the beam separation being 1 degree. Since the beamwidth is also approximately one degree, the total coverage at any one time consists of two footprints lying in approximately a North-South relationship, the exact arrangement depending upon the subsatellite point and the boresight location.

These factors have been programmed into the antenna coverage program described in Section IV. The inputs to the ATS-F S-Band ETV dual-feed coverage program (ATSFS) are the subsatellite and boresight locations, beamwidth dimensions and the plotting parameters. The plotting parameters are FX, FY, DX, DY and FACT, all of which are described above.

The modified coverage program can be used to plot footprints either over the WORLDMAP or the MINMAP. Figure 8 shows the footprints with a subsatellite location of 94°West and boresight pointed towards 81°West and 37°North. A listing of the program is included in Appendix E.

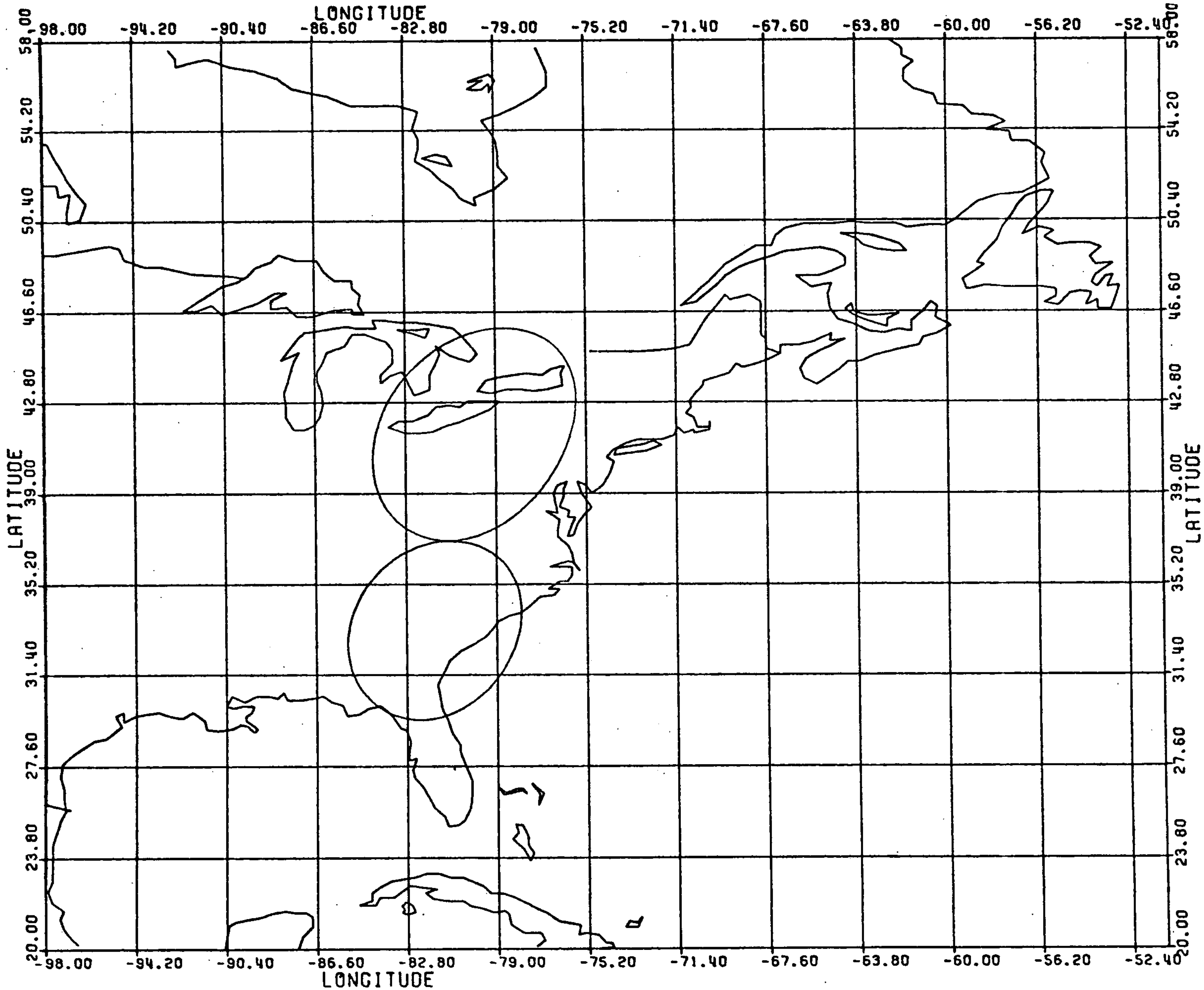

Figure 8. ATS-F S-Band ETV Footprints.
(Satellite at 94°W Longitude)

VII. REFERENCES

1. Thomas W. Stagl, and Jai P. Singh, "A Computer Program for Mapping Satellite-Borne Narrow-Beam Antenna Footprints on Earth", Memorandum No. 72/3, Center for Development Technology, Washington University, Saint Louis, Missouri (March, 1972).

2. "The ATS-F and -G Data Book", Goddard Space Flight Center, Greenbelt, Maryland (October, 1971).

```
C           WORLDMAP
C
      DIMENSION COORD(1024),IBUF(1000),PLTX(514),PLTY(514),N(23)
      CALL PLOTS(IBUF,1000)
C  READ SCALE FACTOR
      READ(5,100) FACT
  100 FORMAT(F10.3)
      CALL FACTOR(FACT)
      CALL PLOT(0.0,-29.5,-3)
      FVX=-3.1415926
      FVY=FVX/2.0
      DAX=2.*3.1415926/50.
      DAY=3.1415926/26.
      FX=-180.
      FY=-90.
      DX=360./50.
      DY=180./26.
      CALL PLOT(2.0,2.5,23)
C  DRAW AXES
      CALL AXIS(0.,0.,'LONGITUDE',-9,50.,0.,FX,DX)
      CALL AXIS(0.0,0.0,'LATITUDE',8,26.,90.,FY,DY)
      CALL AXIS(0.0,26.,'LONGITUDE',9,50.,0.,FX,DX)
      CALL AXIS(50.,0.,'LATITUDE',-8,26.,90.,FY,DY)
      READ(8) N
      K=1
C  READ COORDINATE BLOCK
   70 READ(8)COORD
   81 L=1
   15 I=1
      IF(COORD(2*L-1).GE.4.0)L=L+1
   37 PLTX(I)=COORD(2*L)
      PLTY(I)=COORD(2*L-1)
C  CHECK FOR END OF STRING
      IF(PLTX(I).EQ.0.0.AND.PLTY(I).EQ.0.0)GO TO 47
      IF(I.LT.2)GO TO 31
C  CHECK FOR -180 TO +180 WRAP AROUND
      TEST=ABS(PLTX(I)-PLTX(I-1))
      IF(TEST.GT.2.)GO TO 27
   31 L=L+1
      I=I+1
      GO TO 37
   47 PLTX(I)=FVX
      PLTY(I)=FVY
      PLTX(I+1)=DAX
      PLTY(I+1)=DAY
      NPTS=I-1
C  DRAW STRING OF POINTS
      CALL LINE(PLTX,PLTY,NPTS,1,0,0)
   50 L=L+1
      IF(L.LE.N(K))GO TO 15
      K=K+1
      IF(K.NE.12) GO TO 5
      CALL PLOT(0.,0.,-3)
    5 IF(K.GT.23)GO TO 90
      GO TO 70
   27 PLTX(I)=FVX
      PLTY(I)=FVY
      PLTX(I+1)=DAX
      PLTY(I+1)=DAY
```

```
      NPTS=I-1
C   DRAW STRING OF POINTS
      CALL LINE(PLTX,PLTY,NPTS,1,0,0)
      GO TO 15
   50 CALL PLOT(0.,0.,-3)
C   DRAW GRID LINES
      DO 45 J=1,25
      X=FLOAT(2*J-1)
      NREM=J-(J/2)*2
      IF(NREM.EQ.0) GO TO 46
      CALL PLOT(X,0.,3)
      CALL PLOT(X,26.,2)
      GO TO 45
   46 CALL PLOT(X,26.,3)
      CALL PLOT(X,0.,2)
   45 CONTINUE
      DO 49 J=1,13
      Y=FLOAT(2*J-1)
      NREM=J-(J/2)*2
      IF(NREM.EQ.0) GO TO 48
      CALL PLOT(0.,Y,3)
      CALL PLOT(50.,Y,2)
      GO TO 49
   48 CALL PLOT(50.,Y,3)
      CALL PLOT(0.,Y,2)
   49 CONTINUE
      CALL PLOT(0.,0.,999)
      STOP
      END
```

```
C           MINMAP
C
      DIMENSION IBUF(1000),COORD(1024),N(23),XP(514),YP(514)
      CALL PLOTS(IBUF,1000)
      CALL PLOT(0.,-29.5,-3)
C  READ SCALE FACTOR
      READ (5,99) FACT
   99 FORMAT(F10.3)
      CALL FACTOR(FACT)
      CALL PLOT(1.,.5,23)
      CONVTR=3.1415926535/180.
C  READ BOUNDARIES OF MAP
      READ(5,100) XLL,XUL,YLL,YUL
  100 FORMAT(4F10.3)
      FVX=XLL
      FVY=YLL
      DA=(YUL-YLL)/10.
      WRITE(6,200) FVX,FVY,DA,FACT
  200 FORMAT(1X,'FVX=',F10.4,4X,'FVY=',F10.4,4X,'DA=',F10.4,
     *      4X,'FACT=',F10.4)
      XAXL=(XUL-XLL)/DA
      YAXL=10.
C  DRAW AXES
      CALL AXIS(0.,0.,'LONGITUDE',-9,XAXL,0.,FVX,DA)
      CALL AXIS(0.,0.,'LATITUDE',8,YAXL,90.,FVY,DA)
      CALL AXIS(0.,YAXL,'LONGITUDE',9,XAXL,0.,FVX,DA)
      CALL AXIS(XAXL,0.,'LATITUDE',-8,YAXL,90.,FVY,DA)
      XLL=XLL*CONVTR
      DA=DA*CONVTR
      XUL=XUL*CONVTR
      YLL=YLL*CONVTR
      YUL=YUL*CONVTR
      READ(8) N
      DO 10 K=1,23
      READ(8) COORD
      NL=1
   15 NP=1
      IF(COORD(2*NL-1).LT.4.)GO TO 11
      NL=NL+1
   11 XP(NP)=COORD(2*NL)
      YP(NP)=COORD(2*NL-1)
C  CHECK FOR END OF STRING
      IF(XP(NP).EQ.0..AND.YP(NP).EQ.0.)GO TO 12
C  CHECK FOR POINT IN BOUNDS
      IF(XP(NP).LT.XLL)GO TO 13
      IF(XP(NP).GT.XUL)GO TO 13
      IF(YP(NP).LT.YLL)GO TO 13
      IF(YP(NP).GT.YUL)GO TO 13
      NL=NL+1
      NP=NP+1
      IF(NL.LE.N(K))GO TO 11
      GO TO 10
   12 XP(NP)=XLL
      YP(NP)=YLL
      XP(NP+1)=DA
      YP(NP+1)=DA
      NPTS=NP-1
C  DRAW STRING OF POINTS
      IF(NPTS.LT.2)GO TO 14
```

```
      CALL LINE(XP,YP,NPTS,1,0,0)
   14 NL=NL+1
      IF(NL.GT.N(K))GO TO 10
      GO TO 15
   13 XP(NP)=XLL
      YP(NP)=YLL
      XP(NP+1)=DA
      YP(NP+1)=DA
      NPTS=NP-1
C  DRAW STRING OF POINTS
      IF(NPTS.LT.2)GO TO 17
      CALL LINE(XP,YP,NPTS,1,0,0)
   17 NL=NL+1
      X=COORD(2*NL)
      Y=COORD(2*NL-1)
C  CHECK FOR END OF STRING
      IF(X.EQ.0..AND.Y.EQ.0.)GO TO 14
C  CHECK FOR POINT IN BOUNDS
      IF(X.LT.XLL)GO TO 16
      IF(X.GT.XUL)GO TO 16
      IF(Y.LT.YLL)GO TO 16
      IF(Y.GT.YUL)GO TO 16
      NP=1
      GO TO 11
   16 IF(NL.LT.N(K))GO TO 17
   10 CONTINUE
C  DRAW GRID LINES
      DO 20 I=1,9
      Y=FLOAT(I)
      CALL PLOT(0.,Y,3)
   20 CALL PLOT(XAXL,Y,2)
      NX=INT(XAXL)
      IF(FLOAT(NX).EQ.XAXL)NX=NX-1
      DO 30 J=1,NX
      X=FLOAT(J)
      CALL PLOT(X,0.,3)
   30 CALL PLOT(X,10.,2)
      CALL PLOT(0.,0.,999)
      STOP
      END
```

```
C          ANTOVLY
C
      IMPLICIT REAL*8(A-Z)
      REAL*8 DSIN,DCOS,DSQRT,DARSIN,DARCOS,DATAN
      INTEGER*4 NSTEPS,ICT,I,N,NPLT,IBUF(4000),NP1,NP2,NDB,LDB
      REAL*4 XP(514),YP(514)
      INTEGER*4 J,J1,L,K,NPL
      INTEGER*4 FMT(4)/'(','    ','F7.3',')'/
      INTEGER*4 DIGIT(10)/'1','2','3','4','5','6','7','8','9','10'/
      REAL*4 LON(722),LAT(722),N1,XCT,YCT,FX,FY,DX,DY,XSS,YSS
      INTEGER IM,IP,NP
      INTEGER*4 ISYM(5)
      DATA ISYM/3,4,11,9,10/
      DIMENSION DB(10),DBS(8),RBWS(8)
      DATA DBS/.1,.2,.5,1.,1.5,3.,5.,10./
      DATA RBWS/.18,.26,.4,.56,.7,1.,1.27,1.7/
      DATA PI/3.14159265358979300/,RE/3.96D3/,DIST/2.626D4/
      N=0
      READ(5,205) FX,FY,EX,EY,DX,DY
  205 FORMAT(6F10.3)
      CALL PLOTS(IBUF,4000)
      READ(5,201) FACT
  201 FORMAT(F10.3)
      CALL FACTOR(FACT)
      CONVTR=PI/180.
      BM=DARSIN(RE/DIST)
      CHK=81.3*CONVTR
    2 READ(5,900,END=1000) NPLT
  900 FORMAT(I1)
      DO 33 NPL=1,NPLT
    1 READ(5,500,END=550)DLONSS,DLONCT,DLATCT,DINCR,DDELT
  500 FORMAT(5F9.3)
      READ(5,501)THETA1,BW1A,BW2A,L
  501 FORMAT(3F9.3,I2)
      FMT(2)=DIGIT(L)
      READ(5,FMT)(DB(I),I=1,L)
      NDB=0
      N=N+1
      WRITE(6,600)N,DLONSS,DLONCT,DLATCT,DDELT,BW2A,BW1A,THETA1
  600 FORMAT(//'1DATA SET NO. ',I2/4X,'SUB SAT LONG    =',F9.3/
     1  4X,'BOR SGHT LONG  =',F9.3/4X,'BOR SGHT LAT    =',F9.3/
     2  4X,'DECLINATION    =',F9.3/4X,'MIN BMWDTH',5X,'=',F9.3/
     3  4X,'MAX BMWDTH',5X,'=',F9.3/4X,'ORIENTATION     =',F9.3)
C  CONVERT FROM DEGREES TO RADIANS
      INCR=DINCR*CONVTR
      LONCTR=(DLONCT-DLONSS)*CONVTR
      LATCTR=DLATCT*CONVTR
      DELT=DDELT*CONVTR
      BW1=BW1A*CONVTR/2.0
      BW2=BW2A*CONVTR/2.0
      THETA=THETA1*CONVTR
C  CHECK FOR BORESIGHT LOCATION IN RANGE OF SATELLITE
      AL=DARCOS(DCOS(LONCTR)*DCOS(LATCTR+DELT))
      IF(AL.LE.CHK)GO TO 21
   20 WRITE(6,400)
  400 FORMAT(1H0,'BORESIGHT LOCATION IS NOT IN RANGE OF SATELLITE.'/)
      GO TO 1
C  COMPUTE VECTOR FROM SATELLITE TO BORESIGHT LOCATION
   21 RSX=RE*DCOS(LATCTR)*DSIN(LONCTR)
```

```
      RSY=DIST*DCOS(DELT)-RE*DCOS(LATCTR)*DCOS(LONCTR)
      RSZ=DIST*DSIN(DELT)+RE*DSIN(LATCTR)
      RSM=DSQRT(RSX**2+RSY**2+RSZ**2)
      GAM=DARSIN(RE*DSIN(AL)/RSM)
      EL=((PI/2.0)-(AL+GAM))/CONVTR
      WRITE(6,300) EL
  300 FORMAT(4X,'ELEVATION',6X,'=',F9.3)
      DENOM=DSQRT(RSX**2+RSY**2)
      PITCH=DARSIN(RSX/DENOM)
      ROLL=DATAN(RSZ/DENOM)
      CP=DCOS(PITCH)
      CR=DCOS(ROLL)
      SP=DSIN(PITCH)
      SR=DSIN(ROLL)
   12 DO 32 K=1,L
      NDB=NDB+1
      LDB=L-K+1
C  INTERPOLATE FOR RELATIVE BEAMWIDTH
      DO 8 J=1,8
      IF(DB(LDB)-DBS(J))6,7,8
    7 RBW=RBWS(J)
      GO TO 9
    8 CONTINUE
    6 J1=J-1
      RBW=(DB(LDB)-DBS(J1))*(RBWS(J)-RBWS(J1))/(DBS(J)-DBS(J1))
     1          +RBWS(J1)
    9 A=RSM*DTAN(BW1*RBW)
      B=RSM*DTAN(BW2*RBW)
      ALPH1=BW1A*RBW
      ALPH2=BW2A*RBW
      NSTEPS=(360.0/DINCR)+1
      BMWDTH=DALPHA*RBW
      DO 10 ICT=1,NSTEPS
      BETAN=(ICT-1)*INCR
      ANG=BETAN+THETA
      UM=1.0/DSQRT((DCOS(BETAN)/A)**2+(DSIN(BETAN)/B)**2)
C  COMPUTE N-TH M VECTOR FROM SATELLITE TO LOCUS ON EARTH
      MNPX=RSM*CR*SP+UM*DCOS(ANG)*CP-UM*DSIN(ANG)*SR*SP
      MNPY=RSM*CR*CP+UM*DCOS(ANG)*SP-UM*DSIN(ANG)*SR*CP
      MNPZ=RSM*SR+UM*DSIN(ANG)*CR
      MNX=MNPX
      MNY=MNPY*DCOS(DELT)+MNPZ*DSIN(DELT)
      MNZ=MNPY*(-DSIN(DELT))+MNPZ*DCOS(DELT)
      BN=DARCOS(MNY/DSQRT(MNX**2+MNY**2+MNZ**2))
      IF(BN.GT.BM)GO TO 30
      CN=PI-DARSIN(DSIN(BN)*DIST/RE)
      DN=PI-(BN+CN)
      MNL=DSQRT(RE**2+DIST**2-2.*RE*DIST*DCOS(DN))
      DENOM=DSQRT(MNPX**2+MNPY**2)
      PITCHN=DARSIN(MNPX/DENOM)
      ROLLN=DATAN(MNPZ/DENOM)
C  COMPUTE N-TH VECTOR FROM CENTER OF EARTH TO LOCUS
      REI=MNL*DCOS(ROLLN)*DSIN(PITCHN)
      REJ=DIST*DCOS(DELT)-MNL*DCOS(ROLLN)*DCOS(PITCHN)
      REK=MNL*DSIN(ROLLN)-DIST*DSIN(DELT)
      GO TO 40
C  IF N-TH M VECTOR DOES NOT INTERSECT EARTH COMPUTE VECTOR
C   FROM CENTER OF EARTH TO HORIZON SEEN BY SATELLITE
   30 TAUN=DARCOS(MNZ/DSQRT(MNX**2+MNZ**2))
```

```
      IF(MNX.LT.0.0)TAUN=-TAUN
      REI=RE*DCOS(BM)*DSIN(TAUN)
      REJ=RE*(-DCOS(DELT)*DSIN(BM)-DSIN(DELT)*DCOS(BM)*DCOS(TAUN))
      REK=RE*(-DSIN(DELT)*DSIN(BM)+DCOS(DELT)*DCOS(BM)*DCOS(TAUN))
   40 DEN=DSQRT(REI**2+REJ**2)
C     COMPUTE LONGITUDE AND LATITUDE COORDINATES
      LAT(ICT)=SNGL(DATAN(REK/DEN)/CONVTR)
   10 LON(ICT)=SNGL(DARSIN(REI/DEN)/CONVTR+DLONSS)
      WRITE(6,100)DB(LDB),ALPH1,ALPH2
  100 FORMAT(1H-,3X,'AT ',F4.1,' DB LEVEL:'/4X,'MAX BMWDTH= ',F5.2/
     1        4X,'MIN BMWDTH= ',F5.2//
     2        5X,'LONGITUDE',7X,'LATITUDE'/5X,'(DEGREES)',6X,'(DEGREES)'
     3       /)
      DO 80 I=1,NSTEPS
      WRITE(6,200) LON(I),LAT(I)
  200 FORMAT(1H ,4X,F9.3,6X,F9.3)
      IF(LON(I).LT.-180.) LON(I)=360.+LON(I)
      IF(LON(I).LT.FX) LON(I)=FX
      IF(LON(I).GT.EX) LON(I)=EX
      IF(LAT(I).LT.FY) LAT(I)=FY
      IF(LAT(I).GT.EY)LAT(I)=EY
   80 CONTINUE
      CONVTR=SNGL(CONVTR)
      XCT=(SNGL(DLONCT)-FX)/DX
      YCT=(SNGL(DLATCT)-FY)/DY
      CALL SYMBOL(XCT,YCT,0.14,ISYM(N),0.,-1)
C     PLOT COORDINATES OF LOCUS
      IM=1
   50 IP=1
   31 IF(ABS(LON(IM+1)-LON(IM)).GT.100.)GO TO 70
   37 XP(IP)=LON(IM)
      YP(IP)=LAT(IM)
      IM=IM+1
      IP=IP+1
      IF(IM.LE.NSTEPS)GO TO 47
      XP(IP)=FX
      XP(IP+1)=DX
      YP(IP)=FY
      YP(IP+1)=DY
      NP=IP-1
      CALL LINE(XP,YP,NP,1,0,0)
      GO TO 32
   47 IF(IM.EQ.NSTEPS)GO TO 37
      GO TO 31
   70 XP(IP)=LON(IM)
      YP(IP)=LAT(IM)
      XP(IP+1)=FX
      XP(IP+2)=DX
      YP(IP+1)=FY
      YP(IP+2)=DY
      CALL LINE(XP,YP,IP,1,0,0)
      IM=IM+1
      GO TO 50
   32 CONTINUE
      XSS=(SNGL(DLONSS)-FX)/DX
      YSS=(SNGL(-DDELT)-FY)/DY
   33 CALL SYMBOL(XSS,YSS,0.21,ISYM(N),0.,-1)
  550 CALL PLOT(55.,-1.,23)
      GO TO 2
```

```
C     CLOSE PLOTTAPE
 1000 CALL PLOT(0.,0.,999)
      WRITE(6,601)
  601 FORMAT('0PLOTTAPE CLOSED')
      STOP
      END
```

```
C           PERSPECT
C
       DIMENSION IBUF(1000),COORD(1024),N(23),XP(514),YP(514)
       DIMENSION EL(10)
       REAL*4 LNSS,LTSS
       REAL*4 LNSSR
       DATA PI,S/3.1415926,6.6166/
       CALL PLOTS(IBUF,1000)
       CALL PLOT(0.,-29.5,-3)
       READ(5,400) FACT
   400 FORMAT(F10.3)
       CALL FACTOR(FACT)
       CALL PLOT(17.,15.,23)
       CONVTR=PI/180.
       TMAX=ARSIN(1./S)
       FV=0.0
       DA=CONVTR
       RMAX=90.*CONVTR-TMAX
C      READ SUBSATELLITE COORDINATES
     5 READ(5,100,END=1000)LNSS,LTSS
   100 FORMAT(2F10.2)
       LNSSR=LNSS*CONVTR
       DELTR=LTSS*CONVTR
       SIND=SIN(DELTR)
       COSD=COS(DELTR)
       READ(8)N
       K=1
    80 READ(8)COORD
       L=1
    10 I=1
       IF(COORD(2*L-1).GE.4.)L=L+1
       YL=COORD(2*L-1)
       XL=COORD(2*L)-LNSSR
       SINLT=SIN(YL)
       COSLT=COS(YL)
       SINLN=SIN(XL)
       COSLN=COS(XL)
       COSR=SIND*SINLT+COSD*COSLT*COSLN
       R=ARCOS(COSR)
       DIF=RMAX-R
C      IS FIRST POINT IN STRING WITHIN 30 DEGREES OF HORIZON?
       IF(ABS(DIF).LT..5235988)GO TO 20
C      FIRST POINT >30 DEGREES FROM HORIZON...
C      ... IS IT VISIBLE?
       IF(DIF.GT.0.)GO TO 70
    40 L=L+1
       IF(COORD(2*L-1).EQ.0..AND.COORD(2*L).EQ.0.)GO TO 60
       GO TO 40
C      FIRST POINT <30 DEGREES FROM HORIZON...
C      ... CHECK EACH POINT FOR VISIBILITY.
    20 IF(DIF.GT.0.)GO TO 70
C      NOT VISIBLE...CHECK NEXT POINT
       L=L+1
       IF(COORD(2*L-1).EQ.0..AND.COORD(2*L).EQ.0.)GO TO 60
       YL=COORD(2*L-1)
       XL=COORD(2*L)-LNSSR
       SINLT=SIN(YL)
       COSLT=COS(YL)
       SINLN=SIN(XL)
```

```
      COSLN=COS(XL)
      COSR=SIND*SINLT+COSD*COSLT*COSLN
      R=ARCOS(COSR)
      DIF=RMAX-R
      GO TO 20
C     VISIBLE...CHECK NEXT POINT
   70 T=ATAN(SIN(R)/(S-COSR))
      U=ATAN2(COSLT*SINLN,COSD*SINLT-SIND*COSLT*COSLN)
      XP(I)=T*SIN(U)
      YP(I)=T*COS(U)
      I=I+1
      L=L+1
      IF(COORD(2*L-1).EQ.0..AND.COORD(2*L).EQ.0.)GO TO 50
      YL=COORD(2*L-1)
      XL=COORD(2*L)-LNSSR
      SINLT=SIN(YL)
      COSLT=COS(YL)
      SINLN=SIN(XL)
      COSLN=COS(XL)
      COSR=SIND*SINLT+COSD*COSLT*COSLN
      R=ARCOS(COSR)
      DIF=RMAX-R
      IF(DIF.GT.0.)GO TO 70
      L=L-1
   50 XP(I)=FV
      XP(I+1)=DA
      YP(I)=FV
      YP(I+1)=DA
      NPTS=I-1
C     PLOT STRING
      CALL LINE(XP,YP,NPTS,1,0,0)
   60 L=L+1
      IF(L.LE.N(K))GO TO 10
      K=K+1
      IF(K.LE.23)GO TO 80
C     DRAW HORIZON
      DO 90 J=1,73
      BETA=5.*(J-1)*CONVTR
      XP(J)=TMAX*COS(BETA)
   90 YP(J)=TMAX*SIN(BETA)
      XP(J+1)=FV
      XP(J+2)=DA
      YP(J+1)=FV
      YP(J+2)=DA
      CALL LINE(XP,YP,73,1,0,0)
      CALL SYMBOL(0.,0.,.28,3,0.,-1)
C     DRAW CONSTANT ELEVATION CONTOURS
      READ(5,200) NTS
  200 FORMAT(I1)
      IF(NTS.EQ.0)GO TO 96
      READ(5,300)(EL(I),I=1,NTS)
  300 FORMAT(9F5.3)
      DO 96 K=1,NTS
      ANG2=(90.+EL(K))*CONVTR
      ANG1=ARSIN(SIN(ANG2)/S)
      DO 95 J=1,73
      BETA=5.*(J-1)*CONVTR
      XP(J)=ANG1*COS(BETA)
   95 YP(J)=ANG1*SIN(BETA)
```

```
      XP(J+1)=FV
      XP(J+2)=DA
      YP(J+1)=FV
      YP(J+2)=DA
      CALL LINE(XP,YP,73,1,0,0)
      YN=ANG1/DA+.05
      CALL NUMBER(0.,YN,.21,EL(K),0.,-1)
   90 CONTINUE
      CALL SYMBOL(-1.33,-10.,.21,'LONGITUDE =',0.,11)
      CALL NUMBER(999.,999.,.21,LNSS,0.,3)
      CALL SYMBOL(-1.33,-10.35,.21,'LATITUDE  =',0.,11)
      CALL NUMBER(999.,999.,.21,LTSS,0.,3)
      CALL PLOT(30.,0.,-3)
      REWIND 8
      GO TO 5
 1000 CALL PLOT(20.,-15.,999)
      STOP
      END
```

```
C       ATS-FS
C
      IMPLICIT REAL*8(A-Z)
      REAL*8 DSIN,DCOS,DSQRT,CARSIN,DARCOS,CATAN
      INTEGER*4 NSTEPS,ICT,I,N,NPLT,IBUF(1000),NP1,NP2,NDB,LDB
      REAL*4 FX,FY,DX,DY
      INTEGER*4 J,J1,L,K
      INTEGER*4 FMT(4)/'(','    ','F7.3',')'/
      INTEGER*4 DIGIT(10)/'1','2','3','4','5','6','7','8','9','10'/
      REAL*4 LON(722),LAT(722),N1,XCT,YCT
      INTEGER IJ
      REAL*8 ROLL1,ANGDIF(2)
      DIMENSION DB(10),DBS(8),RBWS(8)
      DATA DBS/.1,.2,.5,1.,1.5,3.,5.,10./
      DATA RBWS/.18,.26,.4,.56,.7,1.,1.27,1.7/
      DATA PI/3.1415926535897930D/,RE/3.96D3/,DIST/2.626D4/
      READ(5,99) FACT
   99 FORMAT(F10.3)
      N=0
      NPLT=0
      CONVTR=PI/180.
      ANGDIF(1)=.5*CONVTR
      ANGDIF(2)=-.5*CONVTR
      BM=DARSIN(RE/DIST)
      CHK=81.3*CONVTR
    1 READ(5,500,END=1000)DLONSS,DLONCT,CLATCT,DINCR,DDELT
  500 FORMAT(5F9.3)
      READ(5,501) BW1A,BW2A
  501 FORMAT(2F9.3)
      N=N+1
      WRITE(6,600)N,DLONSS,DLONCT,CLATCT,DDELT,BW2A,BW1A,THETA1
  600 FORMAT(//'1DATA SET NO. ',I2/4X,'SUB SAT LONG   =',F9.3/
     1  4X,'BOR SGHT LONG  =',F9.3/4X,'BOR SGHT LAT   =',F9.3/
     2  4X,'DECLINATION    =',F9.3/4X,'MIN BMWDTH',5X,'=',F9.3/
     3  4X,'MAX BMWDTH',5X,'=',F9.3/4X,'ORIENTATION    =',F9.3)
C  CONVERT FROM DEGREES TO RADIANS
      INCR=DINCR*CONVTR
      LONCTR=(DLONCT-DLONSS)*CONVTR
      LATCTR=CLATCT*CONVTR
      DELT=DDELT*CONVTR
      BW1=BW1A*CONVTR/2.0
      BW2=BW2A*CONVTR/2.0
C  CHECK FOR BORESIGHT LOCATION IN RANGE OF SATELLITE
      AL=DARCOS(DCOS(LONCTR)*DCOS(LATCTR+DELT))
      IF(AL.LE.CHK)GO TO 21
   20 WRITE(6,400)
  400 FORMAT(1H0,'BORESIGHT LOCATION IS NOT IN RANGE OF SATELLITE.'/)
      GO TO 1
C  COMPUTE VECTOR FROM SATELLITE TO BORESIGHT LOCATION
   21 RSX=RE*DCOS(LATCTR)*DSIN(LONCTR)
      RSY=DIST*DCOS(DELT)-RE*DCOS(LATCTR)*DCOS(LONCTR)
      RSZ=DIST*DSIN(DELT)+RE*DSIN(LATCTR)
      RSM=DSQRT(RSX**2+RSY**2+RSZ**2)
      GAM=DARSIN(RE*DSIN(AL)/RSM)
      EL=((PI/2.0)-(AL+GAM))/CONVTR
      WRITE(6,300) EL
  300 FORMAT(4X,'ELEVATION',6X,'=',F9.3)
      DENOM=DSQRT(RSX**2+RSY**2)
      PITCH=DARSIN(RSX/DENOM)
```

```
      RCLL1=DATAN(RSZ/DENCM)
      DC 31 IJ=1,2
      RCLL=RCLL1+ANGDIF(IJ)
      CP=DCOS(PITCH)
      CR=DCOS(ROLL)
      SP=DSIN(PITCH)
      SR=DSIN(ROLL)
      NPLT=NPLT+1
      IF(NPLT.GT.1)GO TO 15
C   FIRST PLCT, OPEN PLOTTAPE AND SET ORIGIN
      CALL FACTOR(FACT)
      CALL PLOTS(IBUF,1000)
   15 CCNTINUE
      A=RSM*DTAN(BW1)
      B=RSM*DTAN(BW2)
      ALPH1=BW1A
      ALPH2=BW2A
      UM=A
      IF(IJ.EQ.2)UM=B
      NSTEPS=(360.0/DINCR)+1
      DC 10 ICT=1,NSTEPS
      BETAN=(ICT-1)*INCR
      ANG=BETAN
C   COMPUTE N-TH M VECTOR FRCM SATELLITE TO LOCUS CN EARTH
      MNPX=RSM*CR*SP+UM*DCOS(ANG)*CP-UM*DSIN(ANG)*SR*SP
      MNPY=RSM*CR*CP+UM*DCOS(ANG)*SP-UM*DSIN(ANG)*SR*CP
      MNPZ=RSM*SR+UM*DSIN(ANG)*CR
      MNX=MNPX
      MNY=MNPY*DCOS(DELT)+MNPZ*DSIN(DELT)
      MNZ=MNPY*(-DSIN(DELT))+MNPZ*DCOS(DELT)
      BN=DARCOS(MNY/DSQRT(MNX**2+MNY**2+MNZ**2))
      IF(BN.GT.BM)GO TO 30
      CN=PI-DARSIN(DSIN(BN)*DIST/RE)
      DN=PI-(BN+CN)
      MNL=DSQRT(RE**2+DIST**2-2.*RE*DIST*DCOS(CN))
      DENOM=DSQRT(MNPX**2+MNPY**2)
      PITCHN=DARSIN(MNPX/DENCM)
      RCLLN=DATAN(MNPZ/DENCM)
C   COMPUTE N-TH VECTOR FRCM CENTER OF EARTH TC LOCUS
      REI=MNL*DCCS(RCLLN)*DSIN(PITCHN)
      REJ=DIST*DCOS(DELT)-MNL*DCOS(RCLLN)*DCOS(PITCHN)
      REK=MNL*DSIN(RCLLN)-DIST*DSIN(DELT)
      GC TC 40
C   IF N-TH M VECTCR DOES NOT INTERSECT EARTH COMPUTE VECTOR
C    FRCM CENTER OF EARTH TO HORIZCN SEEN BY SATELLITE
   30 TAUN=DARCOS(MNZ/DSQRT(MNX**2+MNZ**2))
      IF(MNX.LT.0.0)TAUN=-TAUN
      REI=RE*DCOS(BM)*DSIN(TAUN)
      REJ=RE*(-DCOS(DELT)*DSIN(BM)-DSIN(DELT)*DCOS(BM)*DCOS(TAUN))
      REK=RE*(-DSIN(DELT)*DSIN(BM)+DCOS(DELT)*DCOS(BM)*DCOS(TAUN))
   40 DEN=DSQRT(REI**2+REJ**2)
C   COMPUTE LCNGITUDE AND LATITUDE COORDINATES
      LAT(ICT)=SNGL(DATAN(REK/DEN)/CONVTR)
   10 LCN(ICT)=SNGL(DARSIN(REI/DEN)/CCNVTR+DLCNSS)
      DO 80 I=1,NSTEPS
   80 WRITE(6,200) LCN(I),LAT(I)
  200 FORMAT(1H ,4X,F9.3,6X,F9.3)
      FX=-151.
      FY=0.
```

```
      DX=4.
      DY=4.
      NP1=NSTEPS+1
      NP2=NSTEPS+2
      LCN(NP1)=FX
      LCN(NP2)=DX
      LAT(NP1)=FY
      LAT(NP2)=DY
C  PLOT COORDINATES CF LOCUS
      CALL LINE(LCN,LAT,NSTEPS,1,0,0)
   31 CCNTINUE
      CALL PLOT(35.,-1.,23)
      GC TO 1
 1000 IF(NPLT.LE.0) GO TO 1001
C  CLOSE PLOTTAPE
      CALL PLCT(0.0,0.C,999)
 1001 WRITE(6,700) NPLT
  700 FORMAT('1NUMBER OF PLOTS PRODUCED =',I3)
      STOP
      END
```

APPENDIX F: INPUT FORMATS

1. ANTOVLY

1.	FX, FY, FY, DY	4F10.3
2.	Magnification Factor	F10.3
3.	Number of beams	I1
4.	SS long, BS long, BS lat, Circumference increment, Declination	5F9.3
5.	Orientation, BW1, BW2, # of DB levels	3F9.3, I2
6.	DB levels	nF7.3

repeat 4,5 & 6 for each beam

2. MINMAP

1.	Magnification Factor	F10.3
2.	XLL, XUL, YLL, DUL	4F10.3

3. WORLDMAP

1.	Magnification Factor	F10.3

4. PERSPECT

1.	Magnification Factor	F10.3
2.	SS long, SS lat	F10.3
3.	# of constant elevation contours	I1
4.	Const. elev. contours	nF5.3

repeat 2,3 & 4 for each case

5. ATSFS

1.	Magnification Factor	F10.3
2.	SS long, BS long, BS lat, Circumference increment, Declination	5F9.3
3.	BW1, BW2	2F9.3