BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B70 09002

SUBJECT: Time Specific Apollo Lunar Surface Accessibility for Relaxed Free Return Missions - Computer Program Description - Case 310

DATE: September 1, 1970

FROM: S. F. Caldwell

ABSTRACT

The region of the lunar surface accessible with a specified landing time and spacecraft configuration can be defined in approximately twenty minutes of CPU time on the Univac 1108. Accessibility can be generated for relaxed free return translunar profiles with a post periselene DPS abort constraint as well as for non-free return translunar trajectories.

Program technical descriptions including flowcharts and listings are presented in sufficient detail to allow the potential user to become familiar with the underlying theory as well as the program logic.

(NASA-CR-113509) TIME SPECIFIC APOLLO LUNAR SURFACE ACCESSIBILITY FOR RELAXED FREE RETURN MISSIONS - COMPUTER PROGRAM DESCRIPTION (Bellcomm, Inc.) 51 p

N79-71658

Unclas
00/12 11863

(PAGES) (CODE)
CR-113509
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)
FF No. 602

AVAILABLE TO NASA OFFICES AND NASA RESEARCH CENTERS ONLY

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B70 09002

SUBJECT: Time Specific Apollo Lunar Surface Accessibility for Relaxed Free Return Missions - Computer Program Description - Case 310

DATE: September 1, 1970

FROM: S. F. Caldwell

MEMORANDUM FOR FILE

Introduction

A new method of generating lunar surface accessibility by establishing extremals for a set of mission design parameters including the effect of a sun elevation constraint at lunar landing has been developed. Reference 1 discusses the method used and presents results for April, 1971 with the J-mission spacecraft configuration and constraints. The computer programs which were written to implement the method are described here with flowcharts and listings. Characteristic velocities are computed using patched conic analysis but corrections for the difference between integrated and patched conic trajectories are made before computing the accessibility boundary.

The accessible regions are defined by scanning all possible trajectory configurations within the mission design constraints. The feasible region defined by the limiting configurations or extremals of the trajectory geometry allowed for any specified lunar arrival time is scanned by means of three free variables: the selenocentric flight path azimuth, α, at MSI entrance, which establishes a family of lunar approach hyperbolas and is bounded by the DPS abort constraint; the angle, ϕ, between the approach hyperbola plane and the final parking orbit; and the coast angle, τ, in the parking orbit from LOI to the landing site. Defining REVLO1 to be the minimum number of orbits to be completed prior to LM landing τ is constrained to be between REVLO1 * 360 and (REVLO1 + 1) * 360. The following diagram graphically illustrates the order of scanning that is used.

- desired time of landing (user specified)
  - launch times and energies for obtaining the specified time of landing (subroutine CONTRL)
    - α_{min} to α_{max} in 2° steps (subroutines ANLSIS and STERN)
      - ϕ_{min} to ϕ_{max} in 2° steps (subroutine ANLSIS)
        - τ_{min} to τ_{max} in .25° steps (subroutines ANLSIS and TAULIM)

Figure 1 shows a typical lunar surface accessibility contour for one time of landing.

Launch Date and Translunar Energy Selection

CONTRL has as its inputs the desired date and time of lunar landing. From this the desired time of LOI is calculated based on a τ of (REVLOI + .5) * 360. An internal subroutine GETERG is called to obtain the translunar injection energy for a given launch day with the desired time of LOI. CONTRL considers all possible launch dates that will give a translunar flight time within the mission constraints and calls subroutine ANLSIS for each to obtain the accessible area of the moon. Before returning, CONTRL calls subroutine PLOT to plot the accessible area both on the printout and on the SC4020 plotter.

Appendix A contains a listing of CONTRL.

α Limits

Subroutine ANLSIS finds the minimum and maximum α based on a DPS abort constraint two hours past periselene (DV9MX). For each α considered the spacecraft is targeted to the MSI by subroutine STERN* and the DPS abort ΔV (ΔV_9) is computed. α is stepped from 220° to 320°** in 2° steps until a ΔV_9 is found less than DV9MX or ΔV_9 starts increasing (in the latter case CONTRL selects another launch date and energy as this one will not meet the DPS abort constraint). Once the DV9MX boundary has been crossed, a straight line fit is used to obtain the α (α_{min}) which has a ΔV_9 within a specified tolerance of DV9MX. α_{max} can be obtained in the same manner by stepping α from 320° to α_{min} in increments of -2°.

ϕ Limits

The maximum plane change (ϕ) possible at LOI must be defined for each α considered between α_{max} and α_{min}. Plane changes can be made either to the left or the right relative to the incoming hyperbola. A sign convention has been assumed to

---

*See Appendix B for a discussion and listing of STERN.

**For retrograde lunar orbits α is limited to 180° to 360°. Imposing a DPS abort constraint eight hours past periselene reduces the α range. In all cases an α range of 220° to 320° was found to be sufficient.

distinguish these two directions: a plane change to the left is negative and a plane change to the right is positive. Thus the largest ϕ made to the left can be called ϕ_{min} and the largest to the right, ϕ_{max}.

Propellant requirements are defined by the three mission dependent maneuvers; LOI, plane change prior to rendezvous (PCM), and TEI. The magnitude of ϕ defines the LOI ΔV as there is no flight path angle change at LOI.* Selection of the point under the resulting orbit subsequently defines PCM ΔV and TEI ΔV. One point under each orbit requires no PCM. ϕ_{min} allows landing at that point, with LOI and TEI consuming <u>all</u> the available SPS propellant. ϕ_{max} is the largest <u>right</u> oriented plane change that can be accomplished, similarly landing under the orbit where a CSM plane change prior to rendezvous is unnecessary and LOI and TEI consume all the available propellant. ϕ's between ϕ_{min} and ϕ_{max} generate LOI and TEI ΔV's that do not consume all the propellant; the remaining propellant can be used for the PCM. The ϕ extremals are computed by subroutine ANLSIS.

<u>τ Limits</u>

For each ϕ between ϕ_{min} and ϕ_{max} the τ limits are computed by subroutine TAULIM.** Using one of the limits returned by TAULIM the trajectory is targeted to earth landing*** and the total propellant burned (FUEL) based on the current ΔV budget is computed by subroutine WGTBUD.**** By comparing FUEL to the total propellant available (FUELMX) a CSM plane change ΔV is computed to use all the remaining available propellant. τ limits based on

---

*A parametric study has shown that off perilune deboost into lunar orbit does not appreciably increase the accessible region. In some cases, however, off perilune LOI does reduce ΔV costs to a site within the accessible region.

**Appendix C contains a discussion, listing and flowchart of TAULIM.

***Subroutine GNDHOM is used to generate a fuel optimal transearth trajectory from lunar parking orbit to earth landing. The trajectory is fuel optimized within the constraints with respect to longitude of earth landing and time of flight. FLYHOM is a subroutine used by GNDHOM. Reference 2 contains a discussion of GNDHOM and FLYHOM.

****Reference 3 contains a discussion of the method used in WGTBUD.

this CSM plane change ΔV are then obtained. This iterative procedure is repeated (for both τ_{min} and τ_{max}) until a trajectory is obtained that uses all the available propellant. For each τ between τ_{min} and τ_{max} the point on the lunar surface where landing is possible is calculated and compared against the periphery matrix by subroutine SAVE.

Accessibility Periphery

Subroutine SAVE collects the periphery matrix of accessibility between longitudes of ±60°. The periphery is collected for every half degree of longitude. For each longitude the minimum and maximum latitude is saved. The periphery matrix is initialized once for each desired lunar landing time. Upon entering SAVE the latitude of the lunar landing site is compared to the minimum and maximum latitude in the periphery matrix corresponding to the longitude of the lunar landing site. If an improvement is made the periphery matrix is updated. In conjunction with the periphery matrix there is a data matrix so that each point in the periphery matrix can be identified. The data matrix stores the date and time of launch; the date and time of MSI entrance; the date and time of lunar landing; α; whether the α was a minimum or maximum; the translunar injection energy; whether or not the point was generated by ϕ_{min}, ϕ_{max}, τ_{min} or τ_{max}; and whether the whole orbit was accessible.

Subroutine ANLSIS contains four internal subroutines: GNDHOM, FLYHOM, SAVE and WGTBUD. Appendix D contains a flowchart of ANLSIS and a listing of ANLSIS and its internal subroutines.

Plots (Subroutine PLOT)

The input to PLOT is the periphery matrix that was collected for one lunar arrival time. The entire periphery matrix is plotted on the on line printer and a smoothed periphery matrix is plotted on the SC4020 plotter. Figure 1 shows a sample SC4020 plot. The plotting package used was AUPLOT, a description of which can be found in Reference 4.

Appendix E contains a listing of PLOT.

Remarks

On the Univac 1108 the generation of the lunar surface accessible for one landing time takes approximately 20 minutes of CPU time or about 170 charge units. At the present time the program will compute relaxed free return translunar trajectories

with a DPS abort constraint. The DPS abort constraint could be removed by performing minor modifications to the program ANLSIS. Monitoring of the RCS abort capability is not currently included.

For each arrival time the entire periphery matrix and data matrix are punched on cards. These cards can be used to determine the accessible areas for various lighting constraints or the accessible area by launch date. Reference 5 contains the accessible region of the lunar surface for a lighting constraint of 5° to 14° and 20° to 30° and the accessible regions by launch date for both November and December, 1971.

S F Caldwell

2013-SFC-slr S. F. Caldwell

Attachments

REFERENCES

1. Bass, R. A. and Stern, R. J., "Determination of Apollo Lunar Surface Accessibility Subject to DPS Abort and Lighting Constraints," Bellcomm Technical Memorandum TM-70-2013-2, March 31, 1970.

2. Kerr, M. R. and Stern, R. J., "An Improved Determination of Optimal Moon-to-Earth Trajectories for BCMASP," Bellcomm Memorandum for File B70 04050, April 20, 1970.

3. McGarry, C. L., "Spacecraft Weight and ΔV Budget Program on the IBM Terminal," Working Note to D. R. Anselmo, February 18, 1970.

4. Jessup, R. F., "The AUPLOT Graphical Data Processing System," Bellcomm Memorandum for File B69 10117, October 27, 1969.

5. Bass, R. A., "J-Mission Lunar Accessibility During November and December, 1971," Bellcomm Memorandum for File B70 07017, July 7, 1970.

APPENDIX A

CONTRL

Calling Sequence: CALL CONTRL (DATEAR, TIMEAR)

Purpose: CONTRL determines all possible launch times and translunar injection energies for a specified time of lunar landing.

Input:

DATEAR - Desired date of lunar landing
TIMEAR - Desired time of lunar landing

```
C
      SUBROUTINE CONTRL(DATEAR,TIMEAR)
C
      COMMON/OLDNEW/IPLOT
      COMMON/C4/TWV(400)/C5/FLT(130)/C7/PHC(50)
      EQUIVALENCE          (FLT(  7),DATEF ),     (FLT(  8),TIMEF ),
     1(FLT( 24),RM9R  ),   (FLT( 32),RLO   ),     (FLT( 33),REVLO1),
     2(PHC( 15),RM    ),   (PHC( 25),GMM   ),     (PHC( 33),PI    ),
     3(PHC( 37),HRDAY ),   (PHC( 40),FTMILE),     (PHC( 43),SECHR ),
     4(TWV( 35),DATELP),   (TWV( 36),TIMELP),     (TWV( 38),TPFLT ),
     5(TWV( 40),TEOP  ),   (TWV( 45),ERG7P ),     (TWV( 46),DATBP ),
     6(TWV( 47),TIMBP ),   (TWV( 57),RMBPX ),     (TWV( 61),VMBPX ),
     7(TWV( 71),RM9RP ),   (TWV(259),DAT20P),     (TWV(260),TIM20P),
     8(TWV(262),SNHRTL),   (TWV(381),ALPHA )
      DIMENSION RM20PX(3), VM20PX(3), RMBPX(3), VMBPX(3),D(3),E(3),T(3)
C
      ALPHA=270.
      RM9RP=RM9R
      PERLOP=2.*PI*RLO*SQRT(RLO/GMM)
      CALL ADDT(DATEAR,TIMEAR,0.,-(REVLO1+.5)*PERLOP,DAT20R,TIM20R)
      DATEF=DAT20R-4.
      TIMEF=TIM20R
      CALL GETERG
      I=1
      D(I)=DATEF
      T(I)=TIMEF
      E(I)=ERG7P
C
      IF(HRTLP-24..LT.60.)GO TO 10
      I=I+1
      D(I)=DATEF+1.
C
      IF(HRTLP-48..LT.60.)GO TO 10
      I=I+1
      D(I)=DATEF+2.
C
 10   IF(HRTLP+24.GT.110)GO TO 20
      I=I+1
      D(I)=DATEF-1.
C
      IF(HRTLP+48.GT.110)GO TO 20
      I=I+1
      D(I)=DATEF-2.
      GO TO 20
C
 20   DO 30 J=2,I
      DATEF=D(J)
      CALL GETERG
      IF(K20.EQ.20.AND.ABS(HRTLP-HRTLR).GE..1)GO TO 40
 30   E(J)=ERG7P
      T(J)=TIMEF
      GO TO 50
 40   I=I-1
 50   CALL INIT
```

```
      IPLOT=0                                                           005500A1
      DO 60 I2=1,I                                                      005600A1
      DATEF=D(I2)                                                       005700A1
      TIMEF=T(I2)                                                       005800A1
      ERG7P=E(I2)                                                       005900A1
 60   CALL ANLSIS                                                       006000A1
      IF(IPLOT.NE.0)CALL PLOT                                           006100A1
      RETURN                                                            006200A1
C                                                                       006300A1
C                                                                       006400A1
C                                                                       006500A1
      SUBROUTINE GETERG                                                 006600A1
      K20=1                                                             006700A1
      ERG7P=-8000000.                                                   006800A1
      DERG7P=-500000.                                                   006900A1
 10   CALL STERN                                                        007000A1
      CALL CONIC(RM8PX,VM8PX,1,1,RLO,RM20PX,VM20PX,TEMP50)              007100A1
      CALL ADDT(DAT8P,TIM8P,0.,TEMP50,DAT20P,TIM20P)                    007200A1
      HRTLP=(DAT20P-DATELP)*HRDAY+(TIM20P-TIMELP-TPFLT-TEOP)/SECHR      007300A1
      HRTLR=(DAT20R-DATELP)*HRDAY+(TIM20R-TIMELP-TPFLT-TEOP)/SECHR      007400A1
      CALL PRINTR(0,'DATEF TIMEF ERG7P HRTLP HRTLR ',                   007500A1
     .DATEF,TIMEF,ERG7P,HRTLP,HRTLR)                                    007600A1
C                                                                       007700A1
      IF(ABS(HRTLP-HRTLR).LT..1.OR.K20.EQ.20)RETURN                     007800A1
      IF(K20.EQ.1)GO TO 25                                              007900A1
      SHRTL=(HRTLP-SHRTLP)/DERG7P                                       008000A1
      DERG7P=(HRTLR-HRTLP)/SHRTL                                        008100A1
      IF(ABS(DERG7P).LT.1.E6)GO TO 25                                   008200A1
      DERG7P=SIGN(1.E6,DERG7P)                                          008300A1
 25   SHRTLP=HRTLP                                                      008400A1
      ERG7P=ERG7P+DERG7P                                                008500A1
      K20=K20+1                                                         008600A1
      GO TO 10                                                          008700A1
C                                                                       008800A1
      END                                                               008900A1
```

APPENDIX B

STERN

Calling Sequence: CALL STERN

Purpose: STERN is used by subroutine ANLSIS to target the spacecraft from launch to the moon's sphere of influence (MSI). The trajectory is constrained to yield a periselene radius of 60 n.mi.*

Input:

PHIMSP - Latitude of previous MSI entrance point
XLMMSP - Longitude of previous MSI entrance point
ALPHA - Selenocentric flight path azimuth at MSI entrance

Output:

PHIMSP - Latitude of MSI entrance point
XLMMSP - Longitude of MSI entrance point
RM8PX(3) - Position at MSI entrance
VM8PX(3) - Velocity at MSI entrance

Method:

STERN calls subroutine HITMSI to target the spacecraft from launch to a point on the moon's sphere of influence. Using the previous latitude (η) and longitude (ζ) of MSI entrance HITMSI is called to obtain the velocity vector ($\bar{V}_{MSI}$) at the MSI. STERN then computes the latitude (η_o) and longitude (ζ_o) of the negative $\bar{V}_{MSI}$ pierce point. Since the translunar injection energy is defined and the periselene radius is fixed the desired MSI pierce point lies on the dotted circle in Figure 2.

The flight path angle, β, can be computed using basic equations for orbital mechanics. The following definitions were used for the semilatus rectum (p), the orbit eccentricity (e), the semimajor axis (a), and the angular momentum (h).

---

*A parametric study has shown that perilune depression to 40 n.mi. does not increase the accessible region.

$$p = \frac{h^2}{\mu} = r_p \ (1+e) \tag{B1}$$

$$e = 1 - \frac{r_p}{a}$$

$$a = \left(\frac{2}{r} - \frac{V^2}{\mu}\right)^{-1}$$

$$h = rV \sin\beta$$

r is the radius of the moon, V is the velocity magnitude at MSI entrance, and r_p is the desired periselene radius. Substituting in Equation (B1) for h, e and a gives

$$\frac{r^2V^2 \sin^2\beta}{\mu} = 2r_p - r_p^{\ 2}\left(\frac{2}{r} - \frac{V^2}{\mu}\right).$$

Dividing both sides of the equation by $\frac{r^2V^2}{\mu}$,

$$\sin^2\beta = \frac{2r_p\mu}{r^2V^2} - r_p^{\ 2}\left(\frac{2\mu}{r^3V^2} - \frac{1}{r^2}\right)$$

collecting terms and taking the square root gives an equation for $\sin\beta$.

$$\sin\beta = \frac{r_p}{r}\left[1 + \frac{2\mu}{r_pV^2}\left(1 - \frac{r_p}{r}\right)\right]^{1/2}$$

The desired pierce point (η, ζ) for a given α can then be calculated using the law of sines and cosines.

$$\cos\left(\frac{\pi}{2}-\eta\right) = \cos\left(\frac{\pi}{2}-\eta_o\right)\cos\beta + \sin\left(\frac{\pi}{2}-\eta_o\right)\sin\beta\cos\alpha$$

or

$$\eta = \sin^{-1}\left(\sin\eta_o\cos\beta + \cos\eta_o\sin\beta\cos\alpha\right)$$

and

$$\frac{\sin\alpha}{\sin\left(\frac{\pi}{2}-\eta\right)} = \frac{\sin(\zeta-\zeta_o)}{\sin\beta}$$

or

$$\zeta = \zeta_o + \sin^{-1}\left(\frac{\sin\alpha\,\sin\beta}{\cos\eta}\right)$$

If this η and ζ are not within some specified tolerance of the ones that were imputed to HITMSI the whole process is repeated.

```
C                                                                       000100A1
      SUBROUTINE STERN                                                  000200A1
C                                                                       000300A1
      COMMON/C1/VAR(375)/C4/TWV(400)/C5/FLT(130)/C7/PHC(50)             000400A1
C                                                                       000500A1
      EQUIVALENCE            (VAR( 97),DATE  ),     (VAR( 98),TIME  ),  000600A1
     1(VAR(247),REMX  ),     (VAR(250),VEMX  ),     (VAR(266),ICOORD)   000700A1
      EQUIVALENCE            (TWV( 22),PHIMSP),     (TWV( 23),XLMMSP),  000800A1
     1(TWV( 46),DATBP ),     (TWV( 47),TIMBP ),     (TWV( 48),DELTB),   000900A1
     2(TWV( 57),RMBPX ),     (TWV( 61),VMBPX ),     (TWV( 60),RMSIP ),  001000A1
     3(TWV( 71),RM9RP ),     (TWV(266),REVEOP),     (TWV(381),ALPHA )   001100A1
      EQUIVALENCE            (FLT( 22),REVEO ),     (FLT( 24),RM9R  )   001200A1
      EQUIVALENCE            (PHC( 25),GMM   ),     (PHC( 28),RMSI  ),  001300A1
     1(PHC( 34),DTOR  ),     (PHC( 35),RTOD  ),     (PHC( 40),FTMILE)   001400A1
C                                                                       001500A1
      DIMENSION VMBPX(3), RMBPX(3), TEMP1(3), TEMP2(3), TEMP3(3),       001600A1
     1REMX(3), VEMX(3)                                                  001700A1
C                                                                       001800A1
      TOL=.001                                                          001900A1
      A=ALPHA*DTOR                                                      002000A1
      K1=0                                                              002100A1
 100  K1=K1+1                                                           002200A1
      PHINEW=PHIMSP                                                     002300A1
      XLMNEW=XLMMSP                                                     002400A1
      CALL HITMSI                                                       002500A1
      DATE=DATBP                                                        002600A1
      TIME=TIMBP                                                        002700A1
      CALL EPHEM                                                        002800A1
      VMBP=VALUE(VMBPX)                                                 002900A1
      CALL CROSS(REMX,VEMX,TEMP1,1)                                     003000A1
      CALL CROSS(VMBPX,TEMP1,TEMP2,1)                                   003100A1
      CALL CROSS(REMX,TEMP1,TEMP3,0)                                    003200A1
      TEMP50=-DOT(TEMP1,VMBPX)/VMBP                                     003300A1
      PHIMSP=RTOD*ATAN(TEMP50/SQRT(1.-TEMP50**2))                       003400A1
      XLMMSP=RTOD*ATAN2(DOT(REMX,TEMP2),DOT(TEMP3,TEMP2))               003500A1
      BETA=ARSIN(RM9R/RMSI*SQRT(1.+2.*GMM/(RM9R*VMBP**2)*(1.-RM9R/RMSI))003600A1
     .)                                                                 003700A1
      PHIMSP=RTOD*ARSIN(SIN(PHIMSP*DTOR)*COS(BETA)+COS(PHIMSP*DTOR)*    003800A1
     .SIN(BETA)*COS(A))                                                 003900A1
      XLMMSP=XLMMSP+RTOD*ARSIN(SIN(A)*SIN(BETA)/COS(PHIMSP*DTOR))       004000A1
      IF(ABS(PHIMSP-PHINEW).LT.TOL.AND.ABS(XLMMSP-XLMNEW).LT.TOL)GOTO200004100A1
      IF(K1.EQ.20)GO TO 150                                             004200A1
      GO TO 100                                                         004300A1
C                                                                       004400A1
 200  IF(ABS(DELTB).GT.1.)WRITE(6,201)DELTB                             004500A1
 201  FORMAT(' ***HITMSI DID NOT CONVERGE',2E16.8)                      004600A1
      RETURN                                                            004700A1
C                                                                       004800A1
 150  WRITE(6,1)PHIMSP,PHINEW,XLMMSP,XLMNEW                             004900A1
 1    FORMAT(' ***K1=20 IN STERN PHIMSP',2E14.8,5X,'XLMMSP',2E14.8)     005000A1
       GO TO 200                                                        005100A1
      END                                                               005200A1
```

APPENDIX C

TAULIM

Calling Sequence: CALL TAULIM (RM21PX, VM21PX, HMLOPX, DLV55P, TAUL, IORBIT)

Purpose: For a given CSM plane change TAULIM computes the minimum and maximum coast angle, τ, in the parking orbit from LOI to the landing site.

Input:

RM21PX(3)	- Position vector after LOI
VM21PX(3)	- Velocity vector after LOI
HMLOPX(3)	- Angular momentum vector
DLV55P	- CSM plane change
REVLO1	- Minimum number of orbits to be completed prior to landing
HRSTAY	- Lunar surface stay time
RTOD	- Conversion factor - radians to degrees
PERLOP	- Period of lunar orbit

Output:

TAUL(2)	- Minimum ($TAUL_{(1)}$) and maximum ($TAUL_{(2)}$) τ
IORBIT	- Flag denoting whether whole orbit is accessible, 1 yes, 0 no

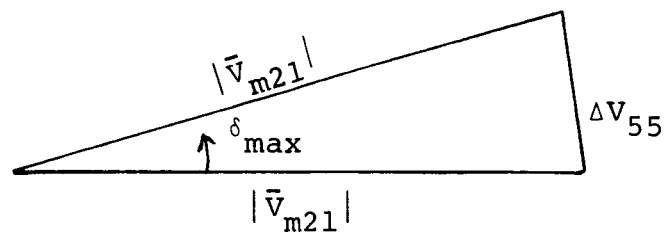
Method:*

In Figure 3 $\bar{R}_{m21}$ and $\bar{V}_{m21}$ are the position and velocity after LOI, $\bar{Z}$ is the selenographic pole in selenographic coordinates, $\bar{R}$ is the position of the CSM above the landing site after LM descent, $\bar{R}'$ is the position of the CSM after the lunar stay time, $\bar{H}_{m1op}$ is $\bar{R}_{m21} \times \bar{V}_{m21}$ and θ_m is the lunar stay time multiplied by the rotation rate of the moon.

---

*Equations were obtained from hand notes of R. J. Stern.

Given a CSM plane change $\Delta V(\Delta V_{55})$ TAULIM can compute the maximum out of plane angle (δ_{max}) after the lunar stay time.



$$\delta_{max} = 2 \sin^{-1} \left[\frac{\Delta V_{55}}{2|\bar{V}_{m21}|} \right]$$

From Figure 3 an equation for $\bar{R}$ in terms of τ can be derived.

$$\frac{\bar{R}}{|\bar{R}|} = \frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \cos\tau + \frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \sin\tau \qquad \text{(C1)}$$

$\bar{R}$ can also be written as the sum of two vectors $\bar{A}$ and $\bar{B}$ where $\bar{A}$ is parallel to $\bar{Z}$ and $\bar{B}$ is perpendicular to $\bar{Z}$.

$$\bar{A} = (\bar{R}\cdot\bar{Z})\bar{Z} \quad \text{and}$$

$$\bar{B} = \bar{R} - \bar{A} \quad \text{or} \quad \bar{R} - (\bar{R}\cdot\bar{Z})\bar{Z}$$

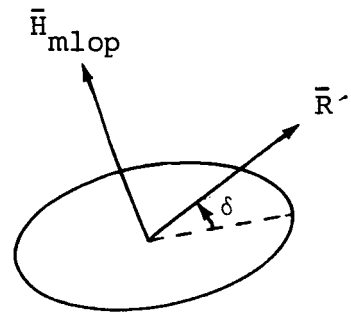
Then using Figure 3

$$\bar{R}' = \bar{B} \cos\theta_m + (\bar{Z}x\bar{B}) \sin\theta_m + \bar{A}$$

or substituting for $\bar{A}$ and $\bar{B}$

$$\bar{R}' = \bar{R}\cos\theta_m + (\bar{R}\cdot\bar{Z})(1-\cos\theta_m)\bar{Z} + (\bar{Z}x\bar{R})\sin\theta_m \qquad \text{(C2)}$$

From the following sketch



it can be seen that

$$\sin\delta_{max} = \bar{H}_{mlop} \cdot \frac{\bar{R}'}{|\bar{R}'|} \qquad \text{(C3)}$$

Substituting equation (C1) and (C2) into equation (C3) gives

$$\bar{H}_{mlop} \cdot \left[\frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \cos\tau \cos\theta_m + \frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \sin\tau \cos\theta_m \right]$$

$$+ \bar{H}_{mlop} \cdot \left[(1-\cos\theta_m) \left\{ \cos\tau \left(\frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \cdot \bar{Z} \right) + \sin\tau \left(\frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \cdot \bar{Z} \right) \right\} \bar{Z} \right]$$

$$+ \bar{H}_{mlop} \times \left[\bar{Z} \times \left(\frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \cos\tau + \frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \sin\tau \right) \right] \sin\theta_m - \sin\delta_{max} = 0$$

Using the scalar triple product and defining α, β, and γ

$$\alpha = (1-\cos\theta_m)(\bar{H}_{mlop} \cdot \bar{Z}) \left(\bar{Z} \cdot \frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \right)$$

$$+ \sin\theta_m \left(\frac{\bar{R}_{m21}}{|\bar{R}_{m21}|} \cdot (\bar{H}_{mlop} \times \bar{Z}) \right)$$

$$\beta = (1-\cos\theta_m)(\bar{H}_{mlop} \cdot \bar{Z}) \left(\bar{Z} \cdot \frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \right)$$

$$+ \sin\theta_m \left(\frac{\bar{V}_{m21}}{|\bar{V}_{m21}|} \cdot (\bar{H}_{mlop} \times \bar{Z}) \right)$$

$$\gamma = \sin\delta_{max}$$

gives an expression in terms of τ

$$\alpha \cos\tau + \beta \sin\tau - \gamma = 0 \tag{C4}$$

Using the identity

$$\sin\tau = \pm(1-\cos^2\tau)^{1/2} \tag{C5}$$

in Equation (C4) and squaring gives an equation for $\cos\tau$ by use of the quadratic equation.

$$\cos\tau = \frac{\alpha\gamma \pm \beta(\alpha^2+\beta^2-\gamma^2)^{1/2}}{\alpha^2 + \beta^2} \tag{C6}$$

There are three cases to be considered:

1. There is no CSM plane change prior to rendezvous. In this case ΔV_{55}, δ and hence γ would be zero and Equation (C4) reduces to

$$\tan\tau = -\frac{\alpha}{\beta}$$

2. There is a CSM plane change prior to rendezvous and $\alpha^2 + \beta^2 - \gamma^2 < 0$ (the whole orbit is accessible). When the whole orbit is accessible the largest ΔV_{55} would occur for $\gamma^2 = \alpha^2 + \beta^2$. Substituting this into Equation (C6) gives

$$\cos\tau = \frac{\alpha}{(\alpha^2+\beta^2)^{1/2}}$$

Substituting in Equation (C4) for $\cos\tau$ an equation is derived for $\sin\tau$.

$$\sin\tau = \frac{\beta}{(\alpha^2+\beta^2)^{1/2}}$$

Therefore,

$$\tan\tau = \frac{\beta}{\alpha}$$

3. There is a CSM plane change prior to rendezvous and $\alpha^2 + \beta^2 - \gamma^2 > 0$. Using Equations (C6) and (C5) there are four possible τ's, two of which are extraneous due to Equation (C5). The two τ's that satisfy Equation (C4) are the desired τ's (τ_{a1} and τ_{a2} with $\tau_{a1} < \tau_{a2}$).

For all three cases the τ's for $+\delta_{max}$ have been found. From Equation (C4) it can be seen that the τ's corresponding to $-\delta_{max}$ are 180° away from the τ's corresponding to $+\delta_{max}$. For case #1 where there is no CSM plane change there are only two possible τ's. For this case the τ corresponding to $+\delta_{max}$ is returned. For case #2 where the whole orbit is accessible τ_{min} and τ_{max} are set to the worst possible cases. For case #3 where only portions of the orbit are accessible the τ limits are more difficult to determine. The τ's corresponding to $-\delta_{max}$ are

$$\tau_{b2} = \tau_{a1} + 180°$$

$$\tau_{b1} = \tau_{a2} + 180°$$

The possible τ regions are shown by the shaded areas in Figure 4. The CSM plane change and TEI ΔV's are the same for τ's between τ_{a1} and τ_{b1} as for τ's between τ_{a2} and τ_{b2}, so only one of these regions needs to be considered. TAULIM returns that region which is entirely within the one orbit. If both regions are within the one orbit the earliest one is returned. The τ region returned is the shaded area that is not cross hatched in Figure 4.

```
C
      SUBROUTINE TAULIM(RM21PX,VM21PX,HMLOPX,DLV55P,TAUL,IORBIT)
C
      COMMON/VALUES/SLON(241),SLAT(241,2),ITVAL(6,241)
      COMMON/C4/TWV(400)/C5/FLT(130)/C7/PHC(50)
     EQUIVALENCE            (FLT (33),REVLO1),     (FLT(35 ),HRSTAY),
    1(PHC( 35),RTOD  ),     (TWV(131),PERLOP)
      DIMENSION HMLOPX(3),  RM21PX(3),  VM21PX(3),  TAUA(2),  TAUB(2),
    1A(3),  Z(3),  TEMP1(3),  TAUL(2),  COSTAU(2),  SINTAU(2),  T(4)
C
      OMEGA=.00015250437
      TAUR1=REVLO1*360.
      TSTAY=AINT(HRSTAY*3600./PERLOP+.25)*PERLOP
      THETAM=OMEGA*TSTAY/RTOD
      A(1)=0.
      A(2)=0.
      A(3)=1.
      CALL MFIXP(0,A,Z)
C
      DELMAX=2.*ARSIN(DLV55P/(2.*VALUE(VM21PX)))
      TEMP50=(1.-COS(THETAM))*DOT(HMLOPX,Z)
      CALL CROSS(HMLOPX,Z,TEMP1,0)
      ALFA=TEMP50*DOT(Z,RM21PX)/VALUE(RM21PX)+SIN(THETAM)*DOT(RM21PX,
    1TEMP1)/VALUE(RM21PX)
      BETA=TEMP50*DOT(Z,VM21PX)/VALUE(VM21PX)+SIN(THETAM)*DOT(VM21PX,
    1TEMP1)/VALUE(VM21PX)
      GAMMA=SIN(DELMAX)
C
      IORBIT=0
      IF(DLV55P.GT.0)GO TO 100
      TAUL(1)=RTOD*ATAN2(-ALFA/BETA,1.)
      IF(TAUL(1).LT.0.)TAUL(1)=TAUL(1)+360.
      TAUL(1)=TAUL(1)+TAUR1
      RETURN
C
 100  TEMP50=ALFA**2+BETA**2
      TEMP51=TEMP50-GAMMA**2
      IF(TEMP51.GT.0.)GO TO 200
      IORBIT=1
      TAUL(1)=RTOD*ATAN2(BETA,ALFA)
      IF(TAUL(1).LT.0.)TAUL(1)=TAUL(1)+180.
      TAUL(1)=TAUL(1)+TAUR1
      TAUL(2)=TAUL(1)+180.
      RETURN
C
 200  TEMP52=ALFA*GAMMA/TEMP50
      TEMP53=BETA*SQRT(TEMP51)/TEMP50
      COSTAU(1)=TEMP52+TEMP53
      COSTAU(2)=TEMP52-TEMP53
      SINTAU(1)=SQRT(1.-COSTAU(1)**2)
      SINTAU(2)=SQRT(1.-COSTAU(2)**2)
      T(1)=ABS(ALFA*COSTAU(1)+BETA*SINTAU(1)-GAMMA)
      T(2)=ABS(ALFA*COSTAU(1)-BETA*SINTAU(1)-GAMMA)
      T(3)=ABS(ALFA*COSTAU(2)+BETA*SINTAU(2)-GAMMA)
```

```
      T(4)=ABS(ALFA*COSTAJ(2)-BETA*SINTAJ(2)-GAMMA)                     005500A1
      J=1                                                               005600A1
      DO 210 I=2,4                                                      005700A1
210   IF(T(I).LT.T(J))J=I                                               005800A1
      K=1                                                               005900A1
      IF(J.EQ.1)K=2                                                     006000A1
      DO 220  I=2,4                                                     006100A1
220   IF(T(I).LT.T(K).AND.I.NE.J)K=I                                    006200A1
      IANS=1                                                            006300A1
      GO TO (240,250,260,270),J                                         006400A1
240   TAJA(IANS)=RTOD*ATAN2(+SINTAJ(1),COSTAJ(1))                       006500A1
      GO TO 280                                                         006600A1
250   TAJA(IANS)=RTOD*ATAN2(-SINTAJ(1),COSTAJ(1))                       006700A1
      GO TO 280                                                         006800A1
260   TAJA(IANS)=RTOD*ATAN2(+SINTAJ(2),COSTAJ(2))                       006900A1
      GO TO 280                                                         007000A1
270   TAJA(IANS)=RTOD*ATAN2(-SINTAJ(2),COSTAJ(2))                       007100A1
280   IF(IANS.EQ.2)GO TO 290                                            007200A1
      IANS=2                                                            007300A1
      GO TO (240,250,260,270),K                                         007400A1
290   IF(TAJA(1).LT.0.)TAJA(1)=TAJA(1)+360.                             007500A1
      IF(TAJA(2).LT.0.)TAJA(2)=TAJA(2)+360.                             007600A1
      IF(TAJA(1).LT.TAJA(2))GO TO 300                                   007700A1
      TEMP50=TAJA(1)                                                    007800A1
      TAJA(1)=TAJA(2)                                                   007900A1
      TAJA(2)=TEMP50                                                    008000A1
300   TAJB(1)=AMOD(TAJA(2)+180.,360.)+TAJR1                             008100A1
      TAJB(2)=AMOD(TAJA(1)+180.,360.)+TAJR1                             008200A1
      TAJA(1)=TAJA(1)+TAJR1                                             008300A1
      TAJA(2)=TAJA(2)+TAJR1                                             008400A1
C                                                                       008500A1
      IF(TAJA(2)-TAJA(1).LT.180.)GO TO 400                              008600A1
      TAJL(1)=TAJA(1)                                                   008700A1
      TAJL(2)=TAJB(1)                                                   008800A1
      RETURN                                                            008900A1
C                                                                       009000A1
400   IF(ABS(TAJA(1)-TAJB(1)).GT.180.)GO TO 500                         009100A1
      TAJL(1)=TAJB(1)                                                   009200A1
      TAJL(2)=TAJA(1)                                                   009300A1
      RETURN                                                            009400A1
C                                                                       009500A1
500   TAJL(1)=TAJA(2)                                                   009600A1
      TAJL(2)=TAJB(2)                                                   009700A1
      RETURN                                                            009800A1
C                                                                       009900A1
      END                                                               010000A1
```

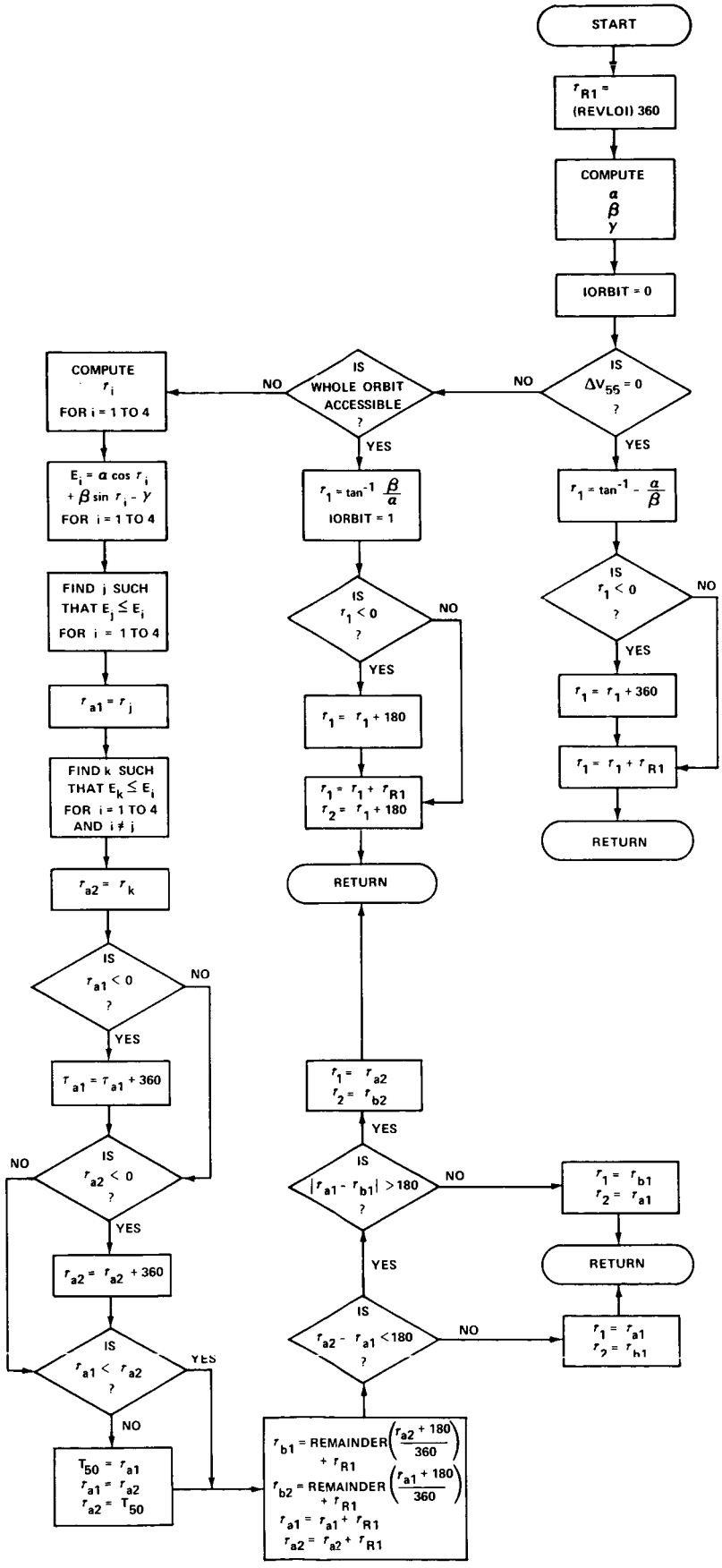


FIGURE C-1: TAULIM FLOWCHART

APPENDIX D

ANLSIS

Calling Sequence: CALL ANLSIS

Purpose: Given a date and time of launch and the translunar injection energy ANLSIS finds the boundary of the accessible region of the moon.

Input:

DATEF - Earliest possible date of launch
TIMEF - Earliest possible time of launch
ERG7P - Translunar injection energy

Output:

SLON(241) - Longitude in .5° from -60° to 60°
SLAT(241,2) - SLAT (I,1) is the minimum latitude and SLAT (I,2) is the maximum latitude corresponding to a longitude of SLON(I)
ITVAL(6,241)- Data matrix associated with the boundary matrix
ITVAL(J,I) - contains date and time of launch and a flag of

0 if $\alpha \neq \alpha_{min}$ and $\alpha \neq \alpha_{max}$
1 if $\alpha = \alpha_{min}$
2 if $\alpha = \alpha_{max}$

ITVAL(J+1,I)- contains date and time of MSI entrance and the value of α
ITVAL(J+2,I)- contains date and time of landing, the translunar injection energy and a flag of

1 if $\phi = \phi_{min}$
2 if $\phi = \phi_{max}$

$$\left.\begin{array}{l} 3 \text{ if } \tau = \tau_{min} \\ 4 \text{ if } \tau = \tau_{max} \\ 5 \text{ if } \tau \neq \tau_{min} \text{ and } \tau \neq \tau_{max} \end{array}\right\} \text{whole orbit is not accessible}$$

$$\left.\begin{array}{l} 6 \text{ if } \tau = \tau_{min} \\ 7 \text{ if } \tau = \tau_{max} \\ 8 \text{ if } \tau \neq \tau_{min} \text{ and } \tau \neq \tau_{max} \end{array}\right\} \text{whole orbit is accessible}$$

J = 1 defines the data matrix for the minimum latitude, J = 4 defines the data matrix for the maximum latitude and I denotes the corresponding position on the longitude matrix.

```
C                                ANLSIS
      SUBROUTINE ANLSIS
C
       COMMON/OLDNEW/IPLOT
       COMMON/VALUES/SLON(241),SLAT(241,2),ITVAL(6,241)
      COMMON/C20/DLLMDA,DF30P,PD26DT
      COMMON/C1/VAR(375)/C4/TWV(400)/C5/FLT(130)/C6/VEH(100)/C7/PHC(50)
      EQUIVALENCE              (FLT(  7),DATEF ),     (FLT(  8),TIMEF ),
     1(FLT( 15),ELTMIN),       (FLT( 16),ELTMAX),     (FLT( 18),ELNMIN),
     2(FLT( 19),ELNMAX),       (FLT( 22),REVEO ),     (FLT( 26),BTA11R),
     3(FLT( 32),RLO    ),      (FLT( 33),REVLO1),     (FLT( 34),REVLO2),
     4(FLT( 35),HRSTAY),       (FLT( 36),HRTEMX),     (FLT( 37),BTA29R),
     5(FLT( 38),FRIMAX),       (FLT( 39),RIMAX ),     (FLT( 40),THMMIN),
     6(FLT( 41),THMMAX),       (FLT(128),DLVTL ),     (FLT(129),DLVTE ),
     7(FLT(130),DLVRSC)
      EQUIVALENCE              (VEH( 16),THR20 ),     (VEH( 17),WGT7  ),
     1(VEH( 18),DWGT20),       (VEH( 22),WGT40 ),     (VEH( 55),WGTAST)
      EQUIVALENCE              (PHC(  1),GSTD  ),     (PHC(  2),REGEOD),
     1(PHC(  5),FLAT   ),      (PHC(  8),OMEGA ),     (PHC( 14),ALTATM),
     2(PHC( 15),RM     ),      (PHC( 25),GMM   ),     (PHC( 28),RMSI  ),
     3(PHC( 33),PI     ),      (PHC( 34),DTOR  ),     (PHC( 35),RTOD  ),
     4(PHC( 36),DEGCIR),       (PHC( 40),FTMILE),     (PHC( 43),SECHR )
      EQUIVALENCE              (TWV( 45),ERG7P ),     (TWV( 46),DAT8P ),
     1(TWV( 47),TIM8P ),       (TWV( 49),QDELT8),     (TWV( 60),RMSIP ),
     2(TWV( 61),VM8PX ),       (TWV( 71),RM9RP ),     (TWV(100),BTA11P),
     3(TWV(101),RE11P ),       (TWV(122),DLV20P),     (TWV(128),HMLOPX),
     4(TWV(131),PERLOP),       (TWV(132),RLOP  ),     (TWV(152),DAT25P),
     5(TWV(153),TIM25P),       (TWV(154),RM25PX),     (TWV(158),DAT26P),
     6(TWV(159),TIM26P),       (TWV(162),DLV26P),     (TWV(173),DAT28P),
     7(TWV(174),TIM28P),       (TWV(185),RE28PX),     (TWV(188),RE28P ),
     8(TWV(194),ALF28P),       (TWV(196),QALF28),     (TWV(207),BTA29P),
     9(TWV(209),DAT30P),       (TWV(210),TIM30P),     (TWV(212),QDLT30),
     .(TWV(213),HRTEP ),       (TWV(214),THETAM),     (TWV(215),RENTC1),
     1(TWV(216),RENTC2),       (TWV(217),ELT30R),     (TWV(218),ELN30R),
     2(TWV(219),ELT30P),       (TWV(220),ELN30P),     (TWV(221),QEL30 ),
     3(TWV(226),VM25PX),       (TWV(234),HRTMXP),     (TWV(242),URX   ),
     4(TWV(245),UTX    ),      (TWV(248),UNX   ),     (TWV(252),VE28TP),
     5(TWV(253),VE28RP),       (TWV(254),DF29P ),     (TWV(256),QTHETM),
     6(TWV(259),DAT20P),       (TWV(260),TIM20P),     (TWV(266),REVEOP)
      EQUIVALENCE              (ELTMNP(2),ELTMXP)
      EQUIVALENCE              (TWV( 35),DATELP),     (TWV( 36),TIMELP),
     1(TWV( 57),RM8PX ),       (TWV(202),RE29P ),     (TWV(381),ALPHA ),
     2(VAR( 97),DATE   ),      (VAR(98 ),TIME  ),     (FLT( 24),RM9R  )
      EQUIVALENCE              (TWV(383),REV2MN),
     1(TWV(384),REV2MX),       (TWV(385),IFLY  ),     (TWV(386),TFLMIN),
     2(TWV(387),TFLMAX ),      (TWV(394),TMMAX ),     (TWV(197),DAT29P),
     3(TWV(198),TIM29P)
      DIMENSION      ELTMIN(2),    SLT30P(2),    CLT30P(2),    ELTMNP(2),
     1 VM23PX(3),    RM23PX(3),    URX   (3),    UTX   (3),    UNX   (3),
     2 HMLOPX(3),    RE28PX(3),    VM8PX (3),    VM21PX(3),    RS23PX(3),
     3 RM25PX(3),    VM25PX(3),    TEMP1 (3),    TEMP2 (3),    TEMP3 (3),
     4 TEMP4 (3),    TAUL  (2),    HML   (3),    RM20PX(3),    VM20PX(3)
     5,TL    (5),    ALFA  (2),    TLP   (5),    DW    (9),    DV    (9),
     6 W    (10)
```

```
C
      REVEOP=REVEO
      RMSIP=RMSI
      QDELT8=.1
      QALF28=.01
      QEL30=.1
      TYPREN=1.
      TPFREN=1.
      RE11P=REGEOD+ALTATM
      BTA11P=BTA11R
      RLOP=RLO
      RE29P=REGEOD+ALTATM
      BTA29P=BTA29R
      DO 10 I=1,2
      TEMP50=DTOR*ELTMIN(I)
      ELTMNP(I)=RTOD*ATAN((1.-FLAT)**2*SIN(TEMP50)/COS(TEMP50))
      TEMP50=DTOR*ELTMNP(I)
      SLT30P(I)=SIN(TEMP50)
   10 CLT30P(I)=COS(TEMP50)
      ELT30P=(ELTMNP(1)+ELTMXP)/2.
      ELN30P=-160.
      ELT30R=ELT30P
      ELN30R=ELN30P
      THETAM=33.3
      ILOOP=1
C
      PERLOP=2.*PI*RLOP*SQRT(RLOP/GMM)
C
       VCIR=SQRT(GMM/RLO)
       DVLOIM=3400.
       DLV20P=0.
       DLV55P=0.
       DLV26P=0.
       CALL WGTBOD
       DV9MX=304.7*GSTD*LOG(W(4)/(W(4)-18550.))
       IPRABT=0
       TA9=7200.
       DELPHI=2.
       TOL=25.
       SDV9AP=1000000.
C
       WRITE(6,15)DW,DV
 15    FORMAT('0INPUT DATA'/' DW',9F10.1/' DV',9F10.1)
       CALL PRINTR(0,'ELN30PTA9   DV9MX FUELMXW(1)  ',
     . ELN30P,TA9,DV9MX,FUELMX,W(1))
       CALL PRINTR(0,'HRSTAYREVEO REVLO1REVLO2      ',
     . HRSTAY,REVEO,REVLO1,REVLO2,A)
C
       INJM=0
C
       ALPHA=220.
       IALPHA=1
       ICOUNT=1
C
 2000  CALL STERN
       CALL ABORT(TA9,DV9AP,TIMAB,ABINC,ABPCH,IPRABT)
       WRITE(6,7) ALPHA,DV9AP
```

```
7       FORMAT(10X,'ALPHA=',F3.2,10X,'DV9=',F16.4)                        01120051
        IF(ICOUNT.EQ.1.AND.DV9AP.LT.DV9MX) GO TO 2400                     01130051
        IF(ABS(DV9AP-DV9MX).LT.TOL/2..OR.ICOUNT.EQ.10)GO TO 2400          01140051
        IF(ICOUNT.EQ.1)GO TO 2100                                         01150051
        IF(ICOUNT.GT.2.OR.(DV9MX-DV9AP)/(DV9MX-SDV9AP).LT.0.)GO TO 2300   01160051
C                                                                         01170051
 2100   SALPHA=ALPHA                                                      01180051
        DV9BK=SDV9AP                                                      01190051
        SDV9AP=DV9AP                                                      01200051
        ICOUNT=2                                                          01210051
        ALPHA=ALPHA+2.*IALPHA                                             01220051
        IF((ALPHA.LT.320.1.AND.DV9AP.LT.DV9BK+100.).OR.IALPHA.LT.0)GO TO 01230051
     . 2000                                                               01240051
        WRITE(6,8)  ERG7P                                                 01250051
8       FORMAT('0       ****THIS ENERGY---', E20.8,' IS UNABORTABLE  ***')01260051
        RETURN                                                            01270051
C                                                                         01280051
 2300   TEMP=(DV9MX-DV9AP)*(ALPHA-SALPHA)/(DV9AP-SDV9AP)                  01290051
        SALPHA=ALPHA                                                      01300051
        SDV9AP=DV9AP                                                      01310051
        ALPHA=ALPHA+TEMP                                                  01320051
        ICOUNT=ICOUNT+1                                                   01330051
        GO TO 2000                                                        01340051
C                                                                         01350051
 2400   IF(IALPHA.LT.0)GO TO 2410                                         01360051
        ALFA(1)=ALPHA                                                     01370051
        ICOUNT=1                                                          01380051
        ALPHA=320.                                                        01390051
        IALPHA=-1                                                         01400051
        GO TO 2000                                                        01410051
 2410   ALFA(2)=ALPHA                                                     01420051
        ALPHA=ALFA(1)-2.                                                  01430051
 2420   IF(ABS(ALPHA-ALFA(2)).LT..01)RETURN                               01440051
        ALPHA=ALPHA+2.                                                    01450051
        IF(ALPHA.GT.ALFA(2))ALPHA=ALFA(2)                                 01460051
C                                                                         01470051
        CALL STERN                                                        01480051
C                                                                         01490051
        CALL ABORT(TA9,DV9AP,TIMAB,ABINC,ABPCH,IPRABT)                    01500051
        WRITE(6,2500)DATEF,ERG7P,ALPHA,DV9AP                              01510051
 2500   FORMAT(///' DATEF',F11.1,5X,'ERG7P',F12.0,5X,'ALPHA',F6.0,5X,'DV901520051
     .AP',F7.0//            T14,'PHI',T21,'TAU',T27,'DLV20  DLV55  DLV26015300051
     .   WGT',T57,'FUEL',T65,'SLSLAT',T75,'SLSLON'/)                      01540051
C                                                                         01550051
        CALL CONIC(RMBPX,VMBPX,1,1,RLO,RM20PX,VM20PX,TEMP50)              01560051
        RLO=RLOP                                                          01570051
        CALL ADDT(DATBP,TIMBP,0.,TEMP50,DAT20P,TIM20P)                    01580051
        VM20=VALUE(VM20PX)                                                01590051
        CALL CROSS(VM20PX,RM20PX,TEMP2,1)                                 01600051
        CALL CROSS(RM20PX,TEMP2,TEMP4,1)                                  01610051
        TEMP76=2.*VCIR*DOT(TEMP4,VM20PX)                                  01620051
C                                                                         01630051
        TAU=(REVLO1+.5)*360.                                              01640051
        ICOUNT=1                                                          01650051
        ITEST=1                                                           01660051
C                                                                         01670051
        IF(INJM.EQ.1)GO TO 3000                                           01680051
```

```
      TMAX=TMMAX*24.                                                    01690051
      IF(IFLY.EQ.2)REVLO2=REV2MN                                        01700051
      TEMP51=(REVLO1+.5+AINT(HRSTAY*SECHR/PERLOP+.25)+REVLO2+1.)*PERLOP01710051
      CALL ADDT(DAT20P,TIM20P,-DATELP,-TIMELP,DDAY,TTIME)               01720051
      TO=DDAY*24.+(TTIME+TEMP51)/3600.                                  01730051
      TGO=TMAX-TO                                                       01740051
      WRITE(6,2600)TGO                                                  01750051
 2600 FORMAT('+',T100,'TGO=',F12.4)                                     01760051
      IF(TGO.GE.112.)GO TO 2900                                         01770051
      IF(TGO.LT.45.)RETURN                                              01780051
      HRTMXP=TGO-2.                                                     01790051
      PHI=0.                                                            01800051
      GO TO 3015                                                        01810051
 2700 ALF28P=90.                                                        01820051
      ELN30P=-160.                                                      01830051
      CALL ADDT(DAT23P,TIM23P,0.,TSTAY+(REVLO2+1.)*PERLOP,DAT25P,TIM25P01840051
     .)                                                                 01850051
      TEMP51=DEGCIR*(REVLO2-AINT(REVLO2))                               01860051
      CALL ROTATE (RM23PX,HMLOPX,TEMP51,RM25PX)                         01870051
      CALL ROTATE (VM23PX,HMLOPX,TEMP51,VM25PX)                         01880051
      CALL HOME                                                         01890051
      CALL ADDT(DAT30P,TIM30P,-DATELP,-TIMELP,DDAY,TTIME)               01900051
      TL(1)=DDAY*24.+TTIME/3600.                                        01910051
      TL(2)=TL(1)+24.                                                   01920051
      TL(3)=TL(1)-24.                                                   01930051
      TL(4)=TL(1)-48.                                                   01940051
      TMAX=AMAX1(TMAX,TO+112.)                                          01950051
      DO 2800 I=1,4                                                     01960051
 2800 IF(TL(I).LE.TMAX.AND.TL(I).GE.(TO+45.))GO TO 2900                 01970051
      RETURN                                                            01980051
 2900 INJM=1                                                            01990051
C                                                                       02000051
 3000 DLV20P=DVLOIM                                                     02010051
      TEMP=(VCIR**2+VM20**2-DLV20P**2)/TEMP76                           02020051
      IF(ABS(TEMP).GT.1.)RETURN                                         02030051
      PHI=-ARCOS(TEMP)*RTOD                                             02040051
C                                                                       02050051
 3015 CALL ROTATE(TEMP4,RM20PX,PHI,VM21PX)                              02060051
      DO 3025 I=1,3                                                     02070051
 3025 VM21PX(I)=VM21PX(I)*VCIR                                          02080051
      CALL CROSS(RM20PX,VM21PX,HML,1)                                   02090051
      IF(INJM.EQ.0)GO TO 6200                                           02100051
C                                                                       02110051
 3050 CALL ADDT(DAT20P,TIM20P,0.,PERLOP*TAU/360.,DATE,TIME)             02120051
      IF(ITEST.LE.2)DLV55P=0.                                           02130051
      CALL TAULIM(RM20PX,VM21PX,HML,DLV55P,TAUL,IORBIT)                 02140051
C                                                                       02150051
 6000 TAU=TAUL(1)                                                       02160051
      IF(ITEST.EQ.4)TAU=TAUL(2)                                         02170051
 6200 CALL ROTATE(RM20PX,HML,TAU,RM23PX)                                02180051
      TEMP51=PERLOP*TAU/360.                                            02190051
      CALL ADDT(DAT20P,TIM20P,0.,TEMP51,DAT23P,TIM23P)                  02200051
      DATE=DAT23P                                                       02210051
      TIME=TIM23P                                                       02220051
      CALL QMFIXP(1,RM23PX,RS23PX)                                      02230051
      IF(ITEST.EQ.5)GO TO 7400                                          02240051
C                                                                       02250051
```

```
C          SIMULATE LUNAR STAY TIME AND CSM PLANE CHANGE
C
      TSTAY=AINT(HRSTAY*SECHR/PERLOP+.25)*PERLOP
      CALL ADDT(DAT23P,TIM23P,0.,TSTAY,DATE,TIME)
      CALL QMFIXP(0,RS23PX,TEMP1)
      CALL CROSS(HML,TEMP1,TEMP2,1)
      CALL CROSS(TEMP1,TEMP2,TEMP3,0)
      PCHCMP=ABS(RTOD*ATAN(DOT(TEMP1,HML)/DOT(TEMP3,HML)))
      DLV55P=VCIR*SQRT(2.-2.*COS(PCHCMP*DTOR))
      TEMP51=VALUE(TEMP1)
      TEMP50=RLD/TEMP51
      DO 302 I=1,3
      HMLOPX(I)=TEMP3(I)/TEMP51
      RM23PX(I)=TEMP50*TEMP1(I)
  302 VM23PX(I)=VCIR*TEMP2(I)
      IF(INUM.EQ.0)GO TO 2700
C
       CALL GNDHOM
C
       CALL WGTBUD
C
       SLSLAT=RTOD*ARSIN(RS23PX(3)/VALUE(RS23PX))
       SLSLON=RTOD*ATAN2(RS23PX(2),RS23PX(1))
       WRITE(6,6100)PHI,TAU,DLV20P,DLV55P,DLV26P,WGT,FUEL,SLSLAT,SLSLON
 6100  FORMAT(9X,F8.2,4F7.0,2F8.0,F9.2,F11.2)
C
       IF(ITEST.GT.2)GO TO 5800
C
       IF(ABS(FUEL-FUELMX).LT.TOL.OR.ICOUNT.EQ.10)GO TO 4000
       IF(ICOUNT.EQ.1)GO TO 3400
       IF(ICOUNT.GT.2.OR.(FUELMX-SFUEL)/(FUELMX-FUEL).LT.0.)GO TO 3500
       IF(FUEL.GT.SFUEL.AND.FUEL.GT.FUELMX)GO TO 3450
C
 3400  SFUEL=FUEL
       SPHI=PHI
       PHI=PHI+DELPHI
       ICOUNT=2
 3420  IF(PHI.GT.0.AND.(PHI-DELPHI).LT.-.1)PHI=0.
 3010  DLV20P=SQRT(VCIR**2+VM20**2-TEMP76*COS(PHI*DTOR))
       GO TO 3015
C
 3450  IF(ITEST.EQ.1)GO TO 2420
 3500  TEMP=(FUELMX-FUEL)*(PHI-SPHI)/(FUEL-SFUEL)
       SFUEL=FUEL
       SPHI=PHI
       PHI=PHI+TEMP
       ICOUNT=ICOUNT+1
       GO TO 3010
C
 4000  IF(ITEST.EQ.2)GO TO 4500
       WRITE(6,9001)PHI
 9001  FORMAT('+',T85,'PHIMIN',F12.4)
       CALL SAVE(1)
       DLV55P=10.
       GO TO 5700
C
 4500  IF(PHI-OLDPHI.LT.3.5)GO TO 5710
```

```
      PHI=OLDPHI                                                        028300S1
      DLV55P=10.                                                        028400S1
      GO TO 5700                                                        028500S1
 5710 WRITE(6,9002)PHI                                                  028600S1
 9002 FORMAT('+',T85,'PHIMAX',F12.4)                                    028700S1
      CALL SAVE(2)                                                      028800S1
      GO TO 2420                                                        028900S1
C                                                                       029000S1
 5000 ITEST=2                                                           029100S1
      ICOUNT=1                                                          029200S1
      PHI=PHI-DELPHI/2.                                                 029300S1
      GO TO 3010                                                        029400S1
C                                                                       029500S1
 5700 PHI=PHI+DELPHI                                                    029600S1
      IF(PHI.GT.0.AND.(PHI-DELPHI).LT.-.1)PHI=0.                        029700S1
      OLDPHI=PHI                                                        029800S1
      ITEST=3                                                           029900S1
      ICOUNT=1                                                          030000S1
      SENS26=0.                                                         030100S1
      GO TO 3010                                                        030200S1
C                                                                       030300S1
 5800 IF(ABS(FUEL-FUELMX).LT.TOL.OR.ICOUNT.EQ.10)GO TO 6500             030400S1
      IF(IORBIT.EQ.1.AND.ITEST.EQ.4.AND.ICOUNT.EQ.1.AND.FUEL.LT.FUELMX)030500S1
     .GO TO 7000                                                        030600S1
      IF(DLV55P.EQ.SDLV55.AND.ICOUNT.NE.1)GO TO 6500                    030700S1
      IF(ICOUNT.NE.1)SENS26=(DLV26P-SSDV26)/(DLV55P-SDLV55)             030800S1
      IF(SENS26.LT.-.9)SENS26=-.9                                       030900S1
      SDLV55=DLV55P                                                     031000S1
      SSDV26=DLV26P                                                     031100S1
      SDW=DW(1)                                                         031200S1
      DO 5850 I=2,6                                                     031300S1
 5850 SDW=SDW+DW(I)                                                     031400S1
      A=LOG((-FUELMX+W(1)-SDW-DW(7)*(1.-EXP(-(DV(8)+DV(9))*D20))-       031500S1
     . DW(8)*(1.-EXP(-DV(9)*D20)))/W(7))                                031600S1
      DLV55P=DLV55P-(A/D20+DV(7)+DV(8)+DV(9))/(1.+SENS26)               031700S1
      IF(DLV55P.LE.0.)GO TO 5000                                        031800S1
      ICOUNT=ICOUNT+1                                                   031900S1
      GO TO 3050                                                        032000S1
C                                                                       032100S1
 6500 IF(ITEST.EQ.4)GO TO 7000                                          032200S1
      ITEST=4                                                           032300S1
      ICOUNT=1                                                          032400S1
      SENS26=0.                                                         032500S1
      TAUMIN=TAU                                                        032600S1
      WRITE(6,9003)TAUMIN                                               032700S1
 9003 FORMAT('+',T85,'TAUMIN',F12.4)                                    032800S1
      MMM=3                                                             032900S1
      IF(IORBIT.EQ.1)MMM=6                                              033000S1
      CALL SAVE(MMM)                                                    033100S1
      IF(FUEL.GE.FUELMX+2.*TOL)GO TO 5700                               033200S1
      GO TO 6000                                                        033300S1
C                                                                       033400S1
 7000 TAUMAX=TAU                                                        033500S1
      IF(TAUMAX-TAUMIN.GT.180.1)TAUMAX=TAUMAX-180.                      033600S1
      WRITE(6,9004)TAUMAX                                               033700S1
 9004 FORMAT('+',T85,'TAUMAX',F12.4)                                    033800S1
      MMM=4                                                             033900S1
```

```
      IF(IORBIT.EQ.1)MMM=7
      CALL SAVE(MMM)
      MMM=5
      IF(IORBIT.EQ.1)MMM=8
      ITEST=5
      DELTAJ=.25
      TAJ=TAJMIN
      GO TO 7500
C
 7400 SLSLAT=RTOD*ARSIN(RS23PX(3)/VALUE(RS23PX))
      SLSLON=RTOD*ATAN2(RS23PX(2),RS23PX(1))
      CALL SAVE(MMM)
 7500 TAJ=TAJ+DELTAJ
      IF(TAJ.GE.TAJMAX)GO TO 5700
      GO TO 6200
C
C
C
C          GRAND HOME
C
C
C
      SUBROUTINE GNDHOM
C
      IF(IFLY.EQ.2)REVLO2=REV2MN
      LOOP=1
3001  CALL ADDT(DAT23P,TIM23P,0.,TSTAY+(REVLO2+1.)*PERLOP,DAT25P,TIM25P)
      TEMP51=DEGCIR*(REVLO2-AINT(REVLO2))
      CALL ROTATE (RM23PX,HMLOPX,TEMP51,RM25PX)
      CALL ROTATE (VM23PX,HMLOPX,TEMP51,VM25PX)
      ILOOP=1
C
C          TRANSEARTH TRAJECTORY
C
      IF(IFLY.NE.0) GO TO 3999
      CALL FLYHOM
      RETURN
3999  TO=(DAT25P-DATELP)*24.+(TIM25P-TIMELP)/3600.
      TGO=(TMMAX+0.1)*24.-TO
      TLMIN=TO+TFLMIN
      TLMAX=TO+TFLMAX
      IF(TGO.GE.TFLMAX) GO TO 4000
      IF(TGO.LT.TFLMIN.AND.LOOP.EQ.1)GO TO 4999
      IF(TGO.LT.TFLMIN.AND.LOOP.GT.1) GO TO 4950
      HRTMXP=TGO-2.
      GO TO 4010
 4000 HRTMXP=TFLMAX-2.
 4010 CALL FLYHOM
      SDLV26=DLV26P
      SHRT=HRTMXP
      TLAND=(DAT29P-DATELP)*24.+(TIM29P-TIMELP)/3600.
      TL(1)=TLAND+24.
      TL(2)=TLAND
      TL(3)=TLAND-24.
      TL(4)=TLAND-48.
      TL(5)=TLAND-72.
      DO 4020 I=1,5
```

```
4020 TLP(I)=0.
     ITLOOP=0
     DO 4030 I=1,5
     IF(TL(I).LT.TLMIN.OR.TL(I).GT.TLMAX.OR.TL(I).GT.TMMAX*24.)
    .GO TO 4030
     ITLOOP=ITLOOP+1
     TLP(ITLOOP)=TL(I)
4030 CONTINUE
     IF(ITLOOP.NE.0) GO TO 4031
     IF(LOOP.GT.1) GO TO 4950
     GO TO 4999
4031 TLPMAX=TLP(1)
     IF(TLPMAX.EQ.TLAND) GO TO 4100
     HRTMXP=TLPMAX-T0+12.
     CALL FLYHOM
     IF(DLV26P.GT.SDLV26)GO TO 4100
     SDLV26=DLV26P
     SHRT=HRTMXP
4100 IF(PD26DT.LE.0.) GO TO 4901
4200 ILOOP=ILOOP+1
     IF(ILOOP.GT.ITLOOP) GO TO 4901
     HRTMXP=TLP(ILOOP)-T0+12.
     CALL FLYHOM
     IF(DLV26P.GT.SDLV26) GO TO 4901
     SDLV26=DLV26P
     SHRT=HRTMXP
     IF(PD26DT.LE.0.)GO TO 4901
     GO TO 4200
4901 IF(LOOP.GT.1.AND.SDV26.LT.SDLV26) GO TO 4902
     SDV26=SDLV26
     SHRMX=SHRT
     SREV2=REVLO2
     IF(IFLY.EQ.1) GO TO 4950
4902 REVLO2=REVLO2+3.
     IF(REVLO2.GT.REV2MX) GO TO 4950
     LOOP=LOOP+1
     GO TO 3001
4950 HRTMXP=SHRMX
     REVLO2=SREV2
     DLV26P=SDV26
     HRTEMX=HRTMXP
     RETURN
4999 WRITE      (6,4998)
4998 FORMAT(/'NO SOLUTION WITHIN FLIGHT TIME CONSTRAINT')
     RETURN
C
C
C
C                                   FLYHOM
C
C
C
      SUBROUTINE FLYHOM
C
     IELON=0
     ELN30P=ELNMIN
3900 K21=1
```

```
 4000 CALL HOME                                                         04540051
      ELT30S=ELT30P                                                     04550051
      SELN30=ELN30P                                                     04560051
      STHM=THETAM                                                       04570051
C                                                                       04580051
C                                                                       04590051
C          CALCULATE LIMITS ON ALF28P DUE TO MAXIMUM INCLINATION        04600051
C                                                                       04610051
      CSLT28=SQRT(RE28PX(1)**2+RE28PX(2)**2)/RE28P                      04620051
      SNLT28=RE28PX(3)/RE28P                                            04630051
      ELT28P=RTOD*ATAN(SNLT28/CSLT28)                                   04640051
      TEMP50=COS(DTOR*RIMAX)/CSLT28                                     04650051
      IF(TEMP50.LT.1.) GO TO 4001                                       04660051
      WRITE(6,9005)                                                     04670051
 9005 FORMAT(1H0,5X,26HRIMAX CAN NOT BE SATISFIED)                      04680051
      CALL CHNXIT                                                       04690051
 4001 ALFMN1=90.*(TYPREN-1.)+RTOD*ATAN(TEMP50/SQRT(1.-TEMP50**2))       04700051
      ALFMX1=TYPREN*180.-ALFMN1                                         04710051
C                                                                       04720051
C          CALCULATE LIMITS ON ALF28P DUE TO LATITUDE CONSTRAINT        04730051
C                                                                       04740051
      ALF1=0.                                                           04750051
      ALF2=180.                                                         04760051
      PHO=ELTMAX                                                        04770051
      I=1                                                               04780051
 4100 CSAL28=(SIN(PHO*DTOR)-SIN(ELT28P*DTOR)*COS(DF30P*DTOR))/(COS(ELT28P04790051
     1P*DTOR)*SIN(DF30P*DTOR))                                          04800051
      IF(ABS(CSAL28).GT.1.) GO TO 4101                                  04810051
      SNAL28=SQRT(1.-CSAL28**2)                                         04820051
      ALF28=RTOD*ATAN2(SNAL28,CSAL28)                                   04830051
      IF(ALF28.LT.90.) ALF1=ALF28                                       04840051
      IF(ALF28.GT.90.) ALF2=ALF28                                       04850051
 4101 I=I+1                                                             04860051
      IF(I.GT.2) GO TO 4102                                             04870051
      PHO=ELTMIN(1)                                                     04880051
      GO TO 4100                                                        04890051
 4102 ALFMN2=ALF1                                                       04900051
      ALFMX2=ALF2                                                       04910051
C                                                                       04920051
C          CALCULATE LIMITS ON ALF28P                                   04930051
C                                                                       04940051
      AL28MN=AMAX1(ALFMN1,ALFMN2)+.01                                   04950051
      AL28MX=AMIN1(ALFMX1,ALFMX2)-.01                                   04960051
      IF(AL28MN.LE.AL28MX)GO TO 4199                                    04970051
      WRITE(6,7001)                                                     04980051
 7001 FORMAT(1H0,5X,'EARTH LANDING AREA IS OUT OF RANGE')               04990051
      CALL CHNXIT                                                       05000051
C                                                                       05010051
C          CALCULATE OPTIMUM ALF28P                                     05020051
C                                                                       05030051
 4199 CALL CROSS(URX,UTX,UNX,0)                                         05040051
      DATE=DAT28P                                                       05050051
      TIME=TIM28P                                                       05060051
      CALL OPTALF(ALF28P,AL28MN,AL28MX,VE28TP,VE28RP,HMLOPX,ALF28T)     05070051
C                                                                       05080051
 5000 IF(K21.EQ.8)RETURN                                                05090051
      IF(ABS(ALF28T-ALF28P).LT.QALF28)GO TO 5510                        05100051
```

```
      ALF28P=ALF28T                                                     05110051
      K21=K21+1                                                         05120051
      GO TO 4000                                                        05130051
 5510 IF(PD26DT.LE.0..OR.IELON.EQ.1) RETURN                             05140051
      IELON=1                                                           05150051
      ELN30P=ELNMAX                                                     05160051
      GO TO 3900                                                        05170051
C                                                                       05180051
C                                                                       05190051
C                                                                       05200051
C                          WGTBUD                                       05210051
C                                                                       05220051
C                                                                       05230051
C                                                                       05240051
      SUBROUTINE WGTBUD                                                 05250051
C                                                                       05260051
      REAL ISP, LMWGT                                                   05270051
      DIMENSION DF(9)                                                   05280051
C                                                                       05290051
      ISP=313.9                                                         05300051
      D20=1./(ISP*GSTD)                                                 05310051
      CMWGT=27290.                                                      05320051
      WGTRNG=98.                                                        05330051
      LMWGT=36379.                                                      05340051
      FUELMX=39410.                                                     05350051
      FUEL=0.                                                           05360051
C                                                                       05370051
C     EVENTS                                                            05380051
C          1 TRANSLUNAR EXPENDABLES                                     05390051
C          2 HYBRID MIDCOURSE BURN                                      05400051
C          3 MIDCOURSE DELTA V                                          05410051
C          4 LOI                                                        05420051
C          5 LM                                                         05430051
C          6 CSM CIRCULARIZATION                                        05440051
C          7 CSM PLANE CHANGE                                           05450051
C          8 LM RESCUE                                                  05460051
C          9 TEI                                                        05470051
C                                                                       05480051
C     WEIGHT DROP AT EVENTS                                             05490051
C                                                                       05500051
      DW(1)=149.                                                        05510051
      DW(2)=118.                                                        05520051
      DW(3)=219.                                                        05530051
      DW(4)=64.                                                         05540051
      DW(5)=610.+LMWGT                                                  05550051
      DW(6)=169.                                                        05560051
      DW(7)=330.                                                        05570051
      DW(8)=299.                                                        05580051
      DW(9)=0.                                                          05590051
C                                                                       05600051
C     DELTA V AT EVENT                                                  05610051
C                                                                       05620051
      DV(1)=0.                                                          05630051
      DV(2)=0.                                                          05640051
      DV(3)=32.9                                                        05650051
      DV(4)=155.+DLV2DP                                                 05660051
      DV(5)=0.                                                          05670051
```

```
      DV(6)=75.                                                         05680051
      DV(7)=DLV55P                                                      05690051
      DV(8)=600.                                                        05700051
      DV(9)=90.+DLV26P                                                  05710051
C                                                                       05720051
      W(1)=CMWGT+WGTRNG+LMWGT+FUELMX                                    05730051
      DO 140 I=2,10                                                     05740051
      W(I)=W(I-1)/EXP(DV(I-1)*D20)-DW(I-1)                              05750051
      DF(I-1)=W(I-1)-W(I)-DW(I-1)                                       05760051
 140  FUEL=FUEL+DF(I-1)                                                 05770051
      WGT=W(10)                                                         05780051
      RETURN                                                            05790051
C                                                                       05800051
C                                                                       05810051
C                                                                       05820051
C                                SAVE                                   05830051
C                                                                       05840051
C                                                                       05850051
C                                                                       05860051
      SUBROUTINE SAVE(MM)                                               05870051
C                                                                       05880051
      IF(FUEL.LT.FUELMX+2.*TOL)GO TO 50                                 05890051
      IF(ITEST.NE.5)WRITE(6,25)                                         05900051
 25   FORMAT('+',T105,'******NO*****NO******')                          05910051
      RETURN                                                            05920051
C                                                                       05930051
 50   IFLAG=0                                                           05940051
      IPLOT=IPLOT+1                                                     05950051
      IF(ABS(SLSLON).LE.90.)GO TO 100                                   05960051
      SLSLAT=-SLSLAT                                                    05970051
      SLSLON=SLSLON-SIGN(180.,SLSLON)                                   05980051
      IFLAG=1                                                           05990051
C                                                                       06000051
 100  IF(ABS(SLSLON).GT.60.2)RETURN                                     06010051
      ILON=2.*(SLSLON+60.)+1.5                                          06020051
      IF(SLSLAT.GE.SLAT(ILON,1))GO TO 200                               06030051
      SLAT(ILON,1)=SLSLAT                                               06040051
      M=1                                                               06050051
      GO TO 300                                                         06060051
 200  IF(SLSLAT.LE.SLAT(ILON,2))RETURN                                  06070051
      SLAT(ILON,2)=SLSLAT                                               06080051
      M=4                                                               06090051
C                                                                       06100051
 300  NDAT=(DATELP-2441000.)                                            06110051
      IF(NDAT.LT.0)NDAT=NDAT+1000                                       06120051
      NDAT=NDAT*100+TIMELP/864.+.5                                      06130051
      NALP=0                                                            06140051
      IF(ABS(ALPHA-ALFA(1)).LT..1)NALP=1                                06150051
      IF(ABS(ALPHA-ALFA(2)).LT..1)NALP=2                                06160051
      ITVAL(M,ILON)=NALP*10000000+NDAT                                  06170051
      NDAT=(DATBP -2441000.)                                            06180051
      IF(NDAT.LT.0)NDAT=NDAT+1000                                       06190051
      NDAT=NDAT*100+TIMBP /864.+.5                                      06200051
      NPAR=ALPHA+.5                                                     06210051
      ITVAL(M+1,ILON)=NPAR*100000+NDAT                                  06220051
      NDAT=(DAT23P-2441000.)                                            06230051
      IF(NDAT.LT.0)NDAT=NDAT+1000                                       06240051
```

```
      IF(TAJ.GT.REVLD1*360.+180.)IFLAG=-IFLAG                           06250051
      NDAT=NDAT*100+(TIM23P+IFLAG*PERLOP/2.)/864.+.5                    06260051
      NPAR=AMOD(-ERG7P,1.E7)/1.E5                                       06270051
      ITVAL(M+2,ILON)=MM*10000000+NPAR*100000+NDAT                      06280051
      IF(M.EQ.1)GO TO 200                                               06290051
      RETURN                                                            06300051
C                                                                       06310051
      END                                                               06320051
```

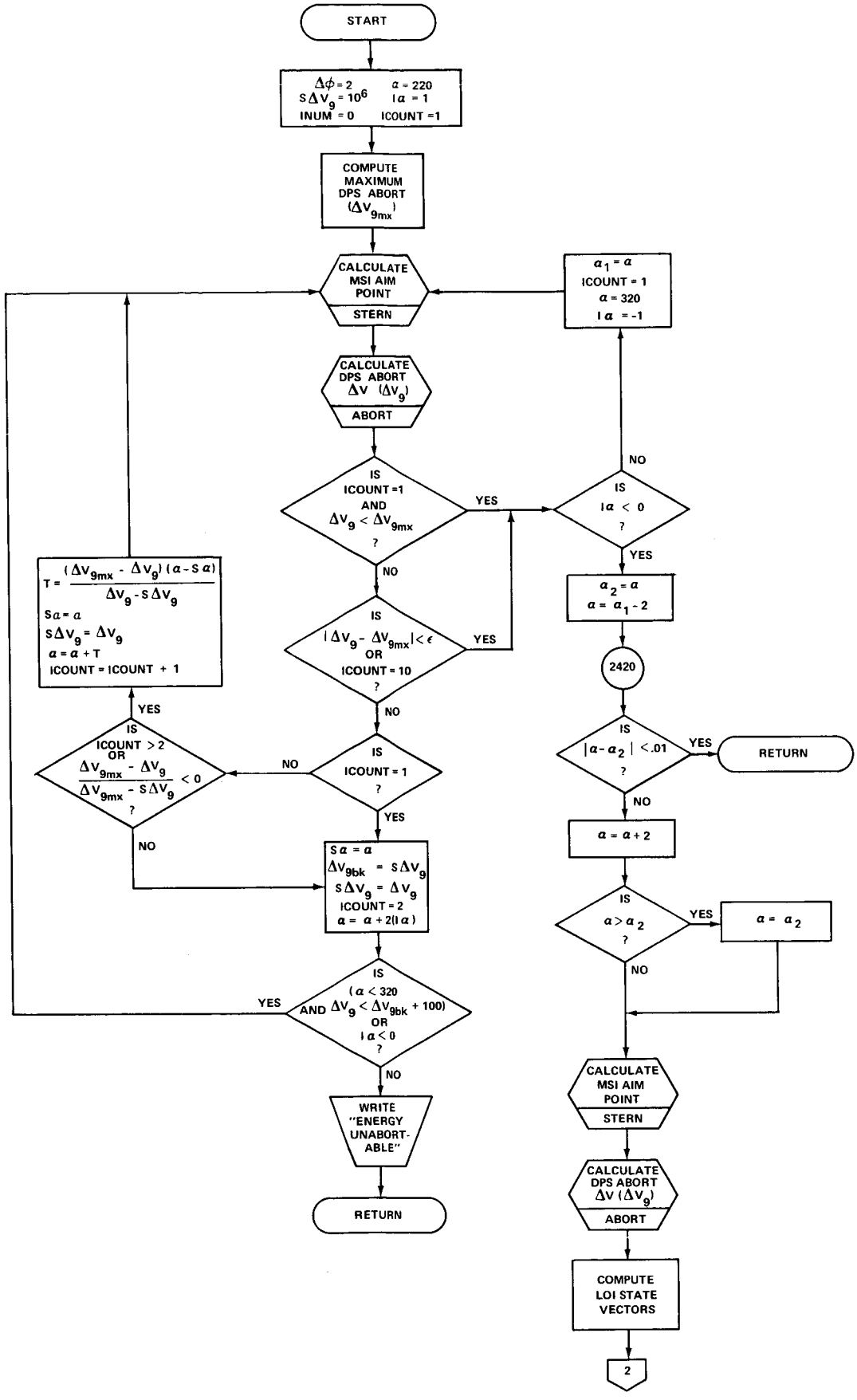


FIGURE D-1: ANLSIS FLOWCHART

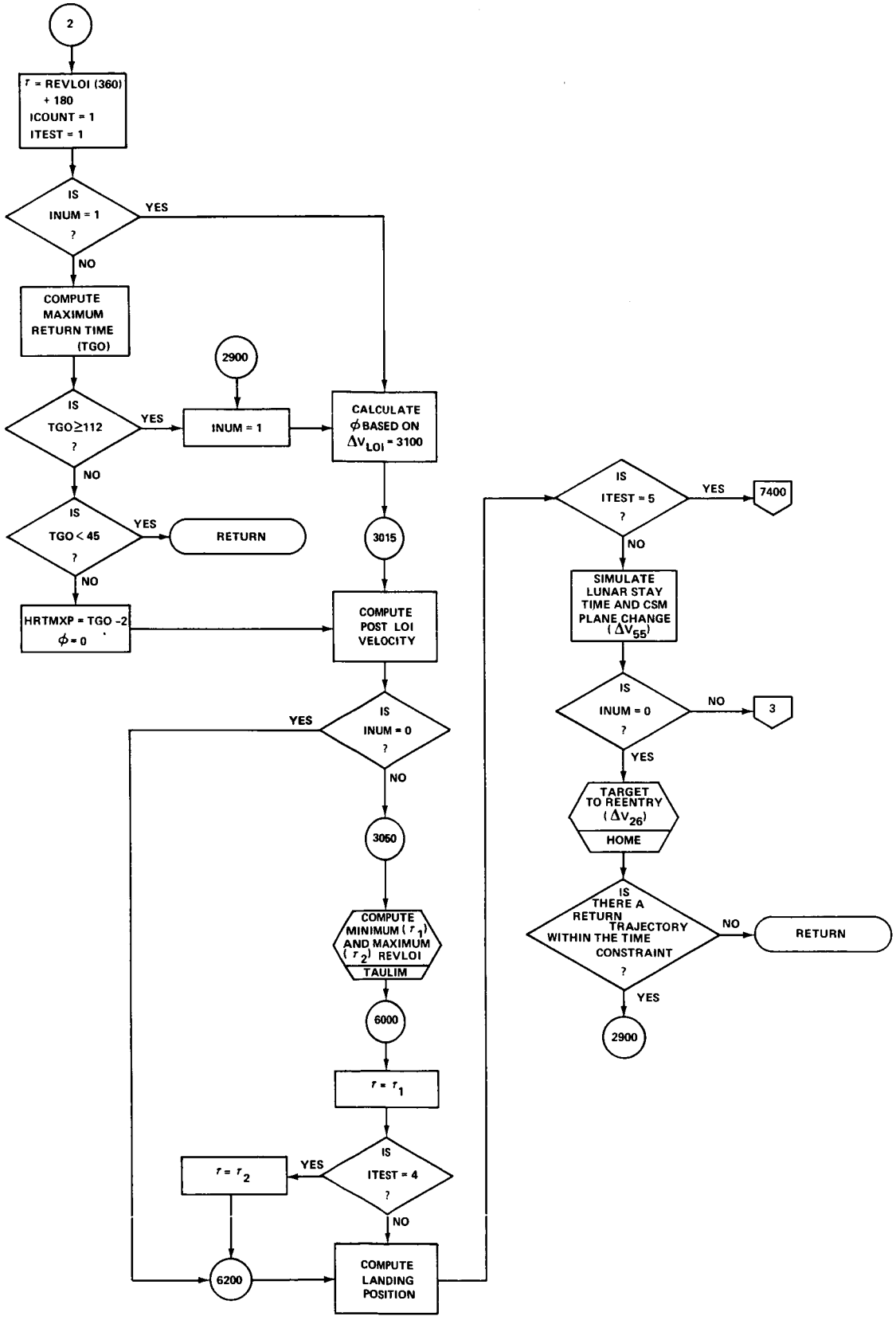


FIGURE D-2: ANLSIS FLOWCHART (CONT.)

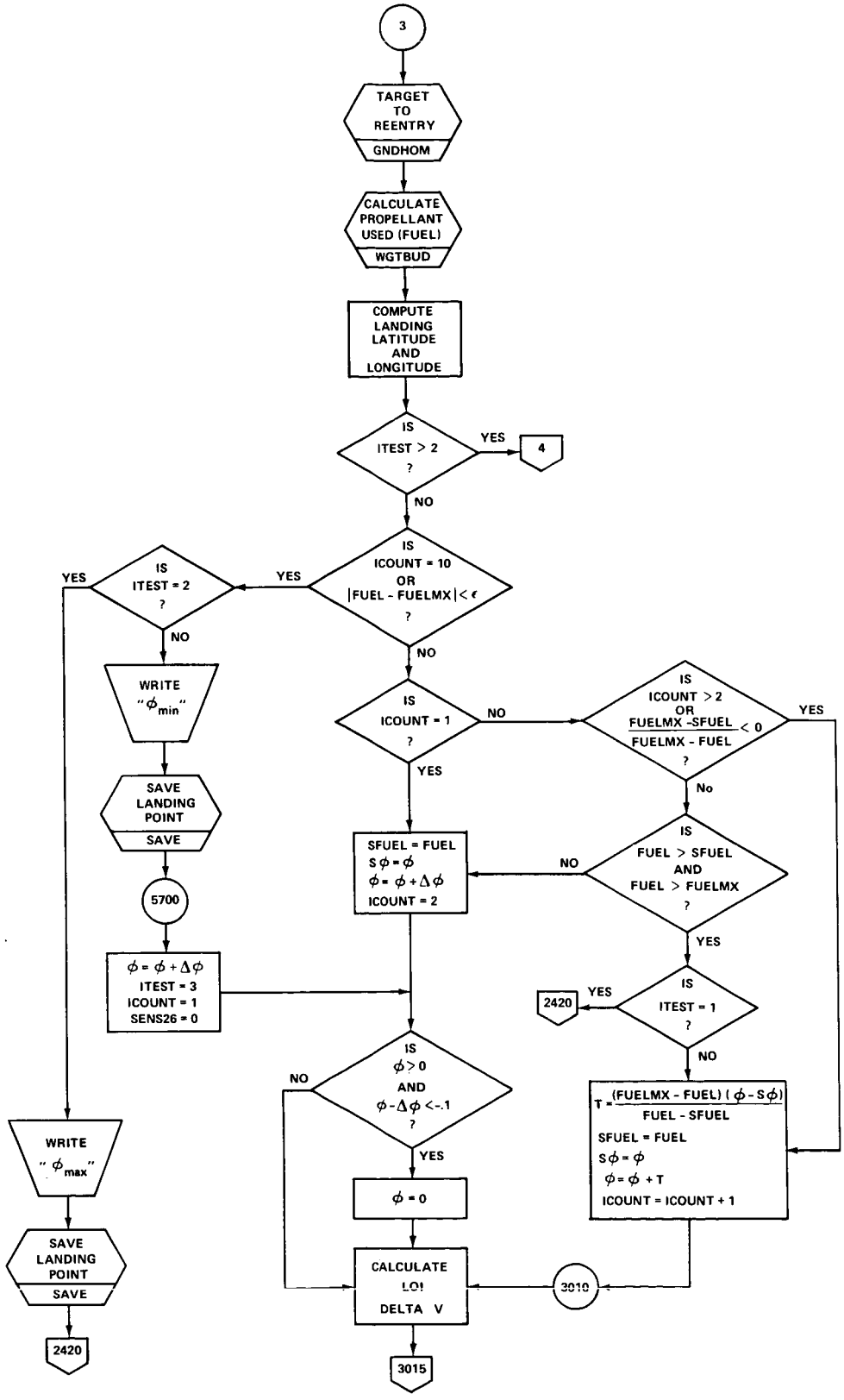


FIGURE D-3: ANLSIS FLOWCHART (CONT.)

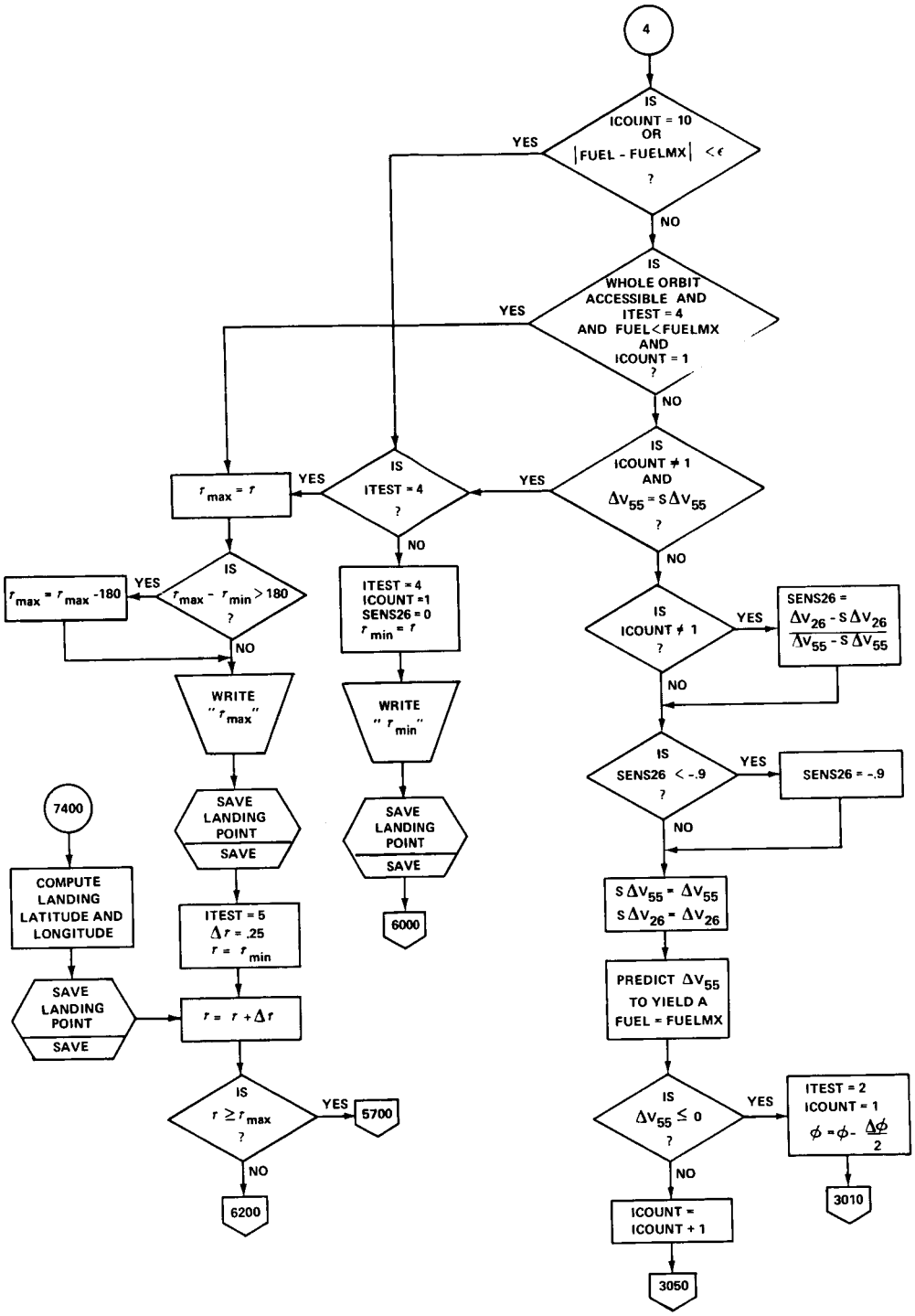


FIGURE D-4: ANLSIS FLOWCHART (CONT.)

APPENDIX E

PLOT

Calling Sequence: CALL PLOT

Purpose: To produce a printer plot of the entire pheriphery matrix, to smooth the pheriphery matrix and then plot the smoothed version on the SC4020 plotter, and to compute the mean arrival time using the data matrix.

Input:*

SLON(241)	- Longitude of landing
SLAT(241,2)	- Minimum and maximum latitude of landing
ITVAL(6,241)	- Data matrix corresponding to boundary matrix

*For further detail see outputs of subroutine ANLSIS in Appendix D.

```
C
      SUBROUTINE PLOT
C
      COMMON/VALUES/SLON(241),SLAT(241,2),ITVAL(6,241)
      REAL INCHES
C
      WRITE(6,2)
 2    FORMAT('1')
      DO 35 L=6,7
 35   WRITE(L,1)(SLON(I),(SLAT(I,K),K=1,2),(ITVAL(J,I),J=1,6),I=1,241)
 1    FORMAT(F6.1,2F7.2,6I10)
C
      D=0.
      DATE=0.
      DO 40 I=1,241
      DO 40 J=3,6,3
      IBLK=MOD(ITVAL(J,I),100000)
      IF(IBLK.NE.0)D=D+1.
 40   DATE=IBLK+DATE
      IF(D.GT.0)DATE=DATE/D
      TIME=AMOD(DATE,100.)*864.
      DATE=INT(DATE/100.)+2441000.5
      HRS=TIME/3600.
      DEG=HRS*12.19/24.
      SCALE=.2
      INCHES=DEG*SCALE
      CALL DANDT(DATE,TIME,IDAY,IMON,IYR,IHR,IMIN,SEC)
      WRITE(6,3)DATE,TIME,IMON,IDAY,IYR,HRS,DEG,INCHES
      WRITE(7,4)DATE,TIME,IMON,IDAY,IYR,HRS,DEG,INCHES
 3    FORMAT('0MEAN ARRIVAL TIME',F11.1,F9.1,3X,A3,I3,I5,F7.2,5X,'DEG',
     . F8.3,5X,'INCHES',F7.3)
 4    FORMAT(F11.1,F9.1,3X,A3,I3,I5,3F8.3)
C
      DO 10 I=1,241
      DO 10 J=1,2
      CALL COLECT('SLSLAT',SLAT(I,J))
 10   CALL COLECT('SLSLON',SLON(I))
      CALL TITLE
      CALL PLCHAR('D')
      CALL PLTICS('SLSLAT',-75.,75.)
      CALL PLTICS('SLSLON',-90.,90.)
      CALL PTX('SLSLON','SLSLAT')
      CALL COLOUT('SLSLAT')
      CALL COLOUT('SLSLON')
C
      CALL PLCHAR(0)
      IPLOTT=0
      IPLOT=0
      DO 65 JJ=1,2
      IF(JJ.EQ.2)GO TO 12
      ISTART=1
      IEND=121
      GO TO 15
 12   ISTART=121
```

```
      IEND=241                                                          005500A1
 15   CALL PLTSIM(1)                                                    005600A1
C                                                                       005700A1
      DO 60 J=1,2                                                       005800A1
      DO 50 I=ISTART,IEND                                               005900A1
C                                                                       006000A1
      IF(ABS(SLAT(I,J)).GT.90.5)GO TO 20                                006100A1
C                                                                       006200A1
      IF(I.EQ.IEND.OR.IPLOT.EQ.0)GO TO 19                               006300A1
      IF(ABS(SLAT(I+1,J)).GT.90.5)GO TO 19                              006400A1
      M1=IBASE+1                                                        006500A1
      M2=I+1                                                            006600A1
      DO 17 M=M1,M2                                                     006700A1
      SM=((SLON(M)-SLON(IBASE))*(SLAT(M2,J)-SLAT(IBASE,J))-(SLAT(M,J)-  006800A1
     . SLAT(IBASE,J))*(SLON(M2)-SLON(IBASE)))/SQRT((SLON(IBASE)-SLON(M2)006900A1
     . )**2+(SLAT(IBASE,J)-SLAT(M2,J))**2)                              007000A1
      IF(J.EQ.1)SM=-SM                                                  007100A1
 17   IF(SM.LT.-.2.OR.SM.GT.2.)GO TO 19                                 007200A1
      GO TO 50                                                          007300A1
 19   IBASE=I                                                           007400A1
C                                                                       007500A1
      CALL COLECT('SLSLAT',SLAT(I,J))                                   007600A1
      CALL COLECT('SLSLON',SLON(I))                                     007700A1
      IPLOT=IPLOT+1                                                     007800A1
      GO TO 50                                                          007900A1
C                                                                       008000A1
 20   IF(IPLOT.GT.0)CALL PLOT2                                          008100A1
 50   CONTINUE                                                          008200A1
 60   IF(IPLOT.GT.0)CALL PLOT2                                          008300A1
 65   IF(IPLOTT.EQ.0)CALL PLOTBL                                        008400A1
      RETURN                                                            008500A1
C                                                                       008600A1
      SUBROUTINE PLOT2                                                  008700A1
      IF(IPLOT.EQ.1)GO TO 100                                           008800A1
      CALL PLTICS('SLSLAT',-40.,50.)                                    008900A1
      CALL PLTICS('SLSLON',-90.,0.)                                     009000A1
      IF(JJ.EQ.2)CALL PLTICS('SLSLON',0.,90.)                           009100A1
      CALL PLTDEN('SLSLAT',-90.)                                        009200A1
      CALL PLTDEN('SLSLON',-90.)                                        009300A1
      CALL QTX('SLSLON','SLSLAT')                                       009400A1
      IPLOTT=1                                                          009500A1
 100  CALL CJLOUT('SLSLAT')                                             009600A1
      CALL CJLOUT('SLSLON')                                             009700A1
      IPLOT=0                                                           009800A1
      RETURN                                                            009900A1
C                                                                       010000A1
      SUBROUTINE PLOTBL                                                 010100A1
      CALL COLECT('SLSLAT',0.)                                          010200A1
      CALL COLECT('SLSLON',-90.)                                        010300A1
      CALL COLECT('SLSLAT',0.)                                          010400A1
      CALL COLECT('SLSLON',-80.)                                        010500A1
      IPLOT=2                                                           010600A1
      CALL PLOT2                                                        010700A1
      RETURN                                                            010800A1
C                                                                       010900A1
C                                                                       011000A1
C                                                                       011100A1
```

```
      SUBROUTINE TITLE                                                  011200A1
      INTEGER T,ZERO,  TIT,F0,F1,F2,F3                                  011300A1
      DIMENSION T(8),TIT(8)                                             011400A1
C                                                                       011500A1
      DATA T/'        MEAN ARRIVAL TIME 2440000.5    00000.0     '/,   011600A1
     . ZERO/'000000'/                                                   011700A1
C                                                                       011800A1
      DO 1 I=1,8                                                        011900A1
    1 TIT(I)=0                                                          012000A1
C                                                                       012100A1
      N1=DATE-2440000                                                   012200A1
      N2=TIME*10                                                        012300A1
C                                                                       012400A1
      F0=N1/1000+ZERO                                                   012500A1
      F1=MOD(N1,1000)/100+ZERO                                          012600A1
      F2=MOD(N1,100)/10+ZERO                                            012700A1
      F3=MOD(N1,10)+ZERO                                                012800A1
      F5=N2/100000+ZERO                                                 012900A1
      F6=MOD(N2,100000)/10000+ZERO                                      013000A1
      F7=MOD(N2,10000)/1000+ZERO                                        013100A1
      F8=MOD(N2,1000)/100+ZERO                                          013200A1
      F9=MOD(N2,100)/10+ZERO                                            013300A1
      F10=MOD(N2,10)+ZERO                                               013400A1
C                                                                       013500A1
      FLD(24,6,TIT(5))=F0                                               013600A1
      FLD(30,6,TIT(5))=F1                                               013700A1
      FLD( 0,6,TIT(6))=F2                                               013800A1
      FLD( 6,6,TIT(6))=F3                                               013900A1
      FLD( 6,6,TIT(7))=F5                                               014000A1
      FLD(12,6,TIT(7))=F6                                               014100A1
      FLD(18,6,TIT(7))=F7                                               014200A1
      FLD(24,6,TIT(7))=F8                                               014300A1
      FLD(30,6,TIT(7))=F9                                               014400A1
      FLD( 6,6,TIT(8))=F10                                              014500A1
C                                                                       014600A1
      DO 2 I=1,8                                                        014700A1
    2 TIT(I)=TIT(I)+T(I)                                                014800A1
C                                                                       014900A1
      CALL PLTITL(TIT)                                                  015000A1
      RETURN                                                            015100A1
C                                                                       015200A1
      END                                                               015300A1
```

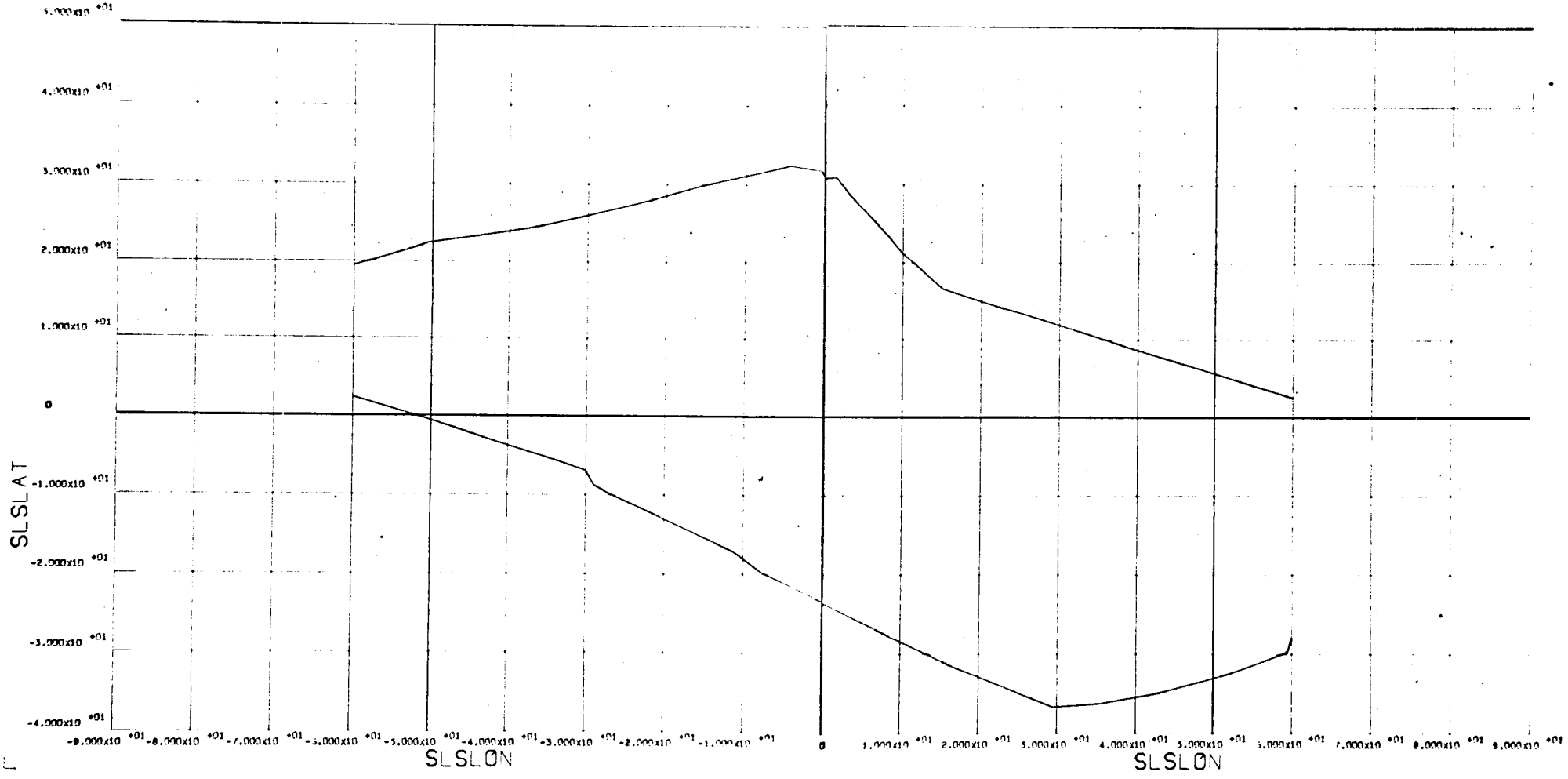


FIGURE 1: SAMPLE SC 4020 PLOT

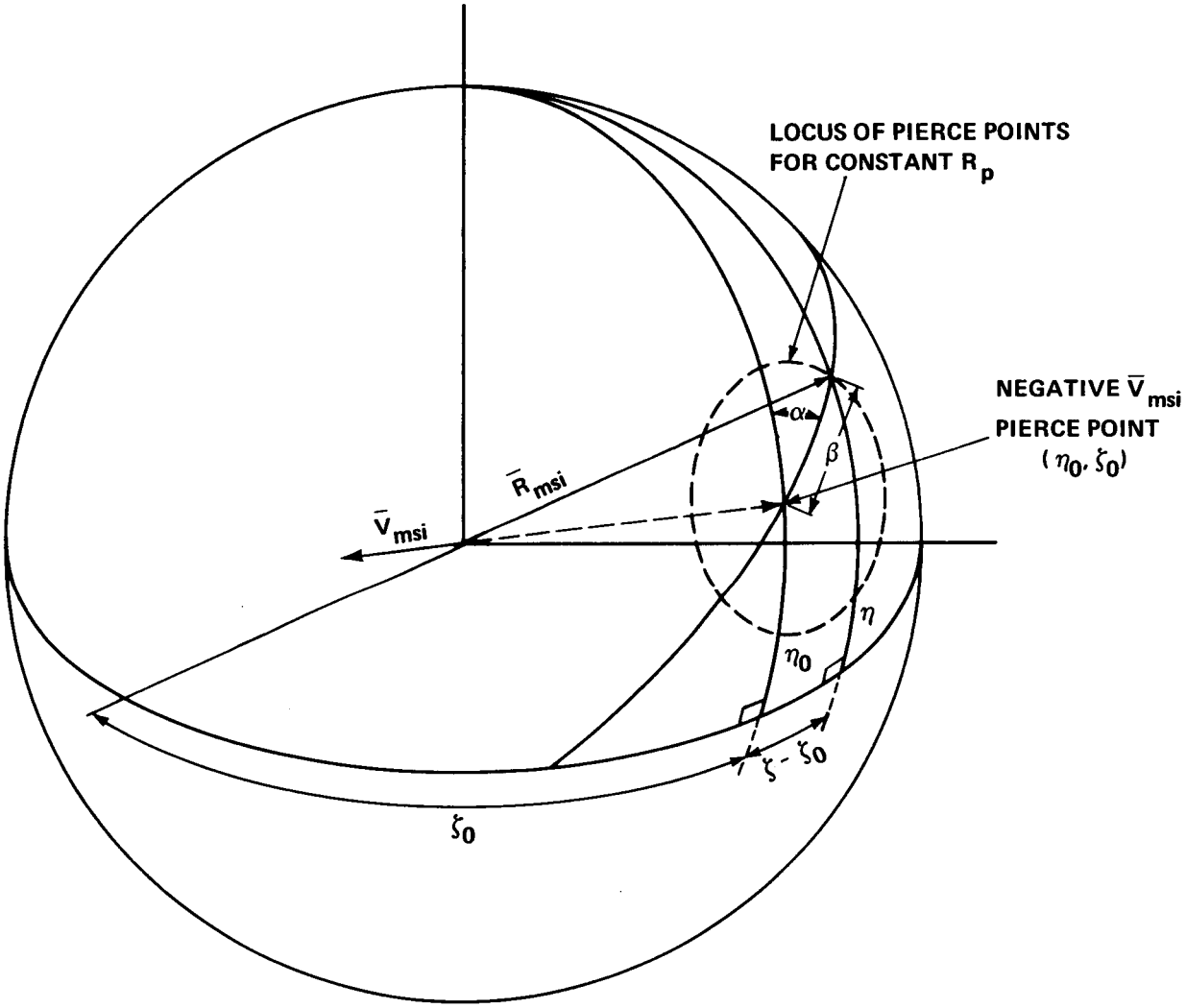


FIGURE 2: MSI ENTRANCE GEOMETRY

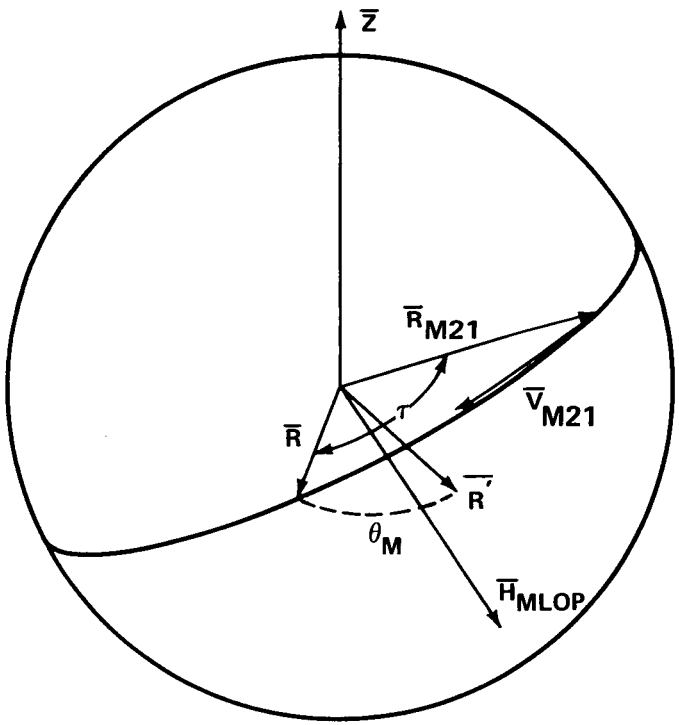


FIGURE 3: LUNAR ORBIT GEOMETRY

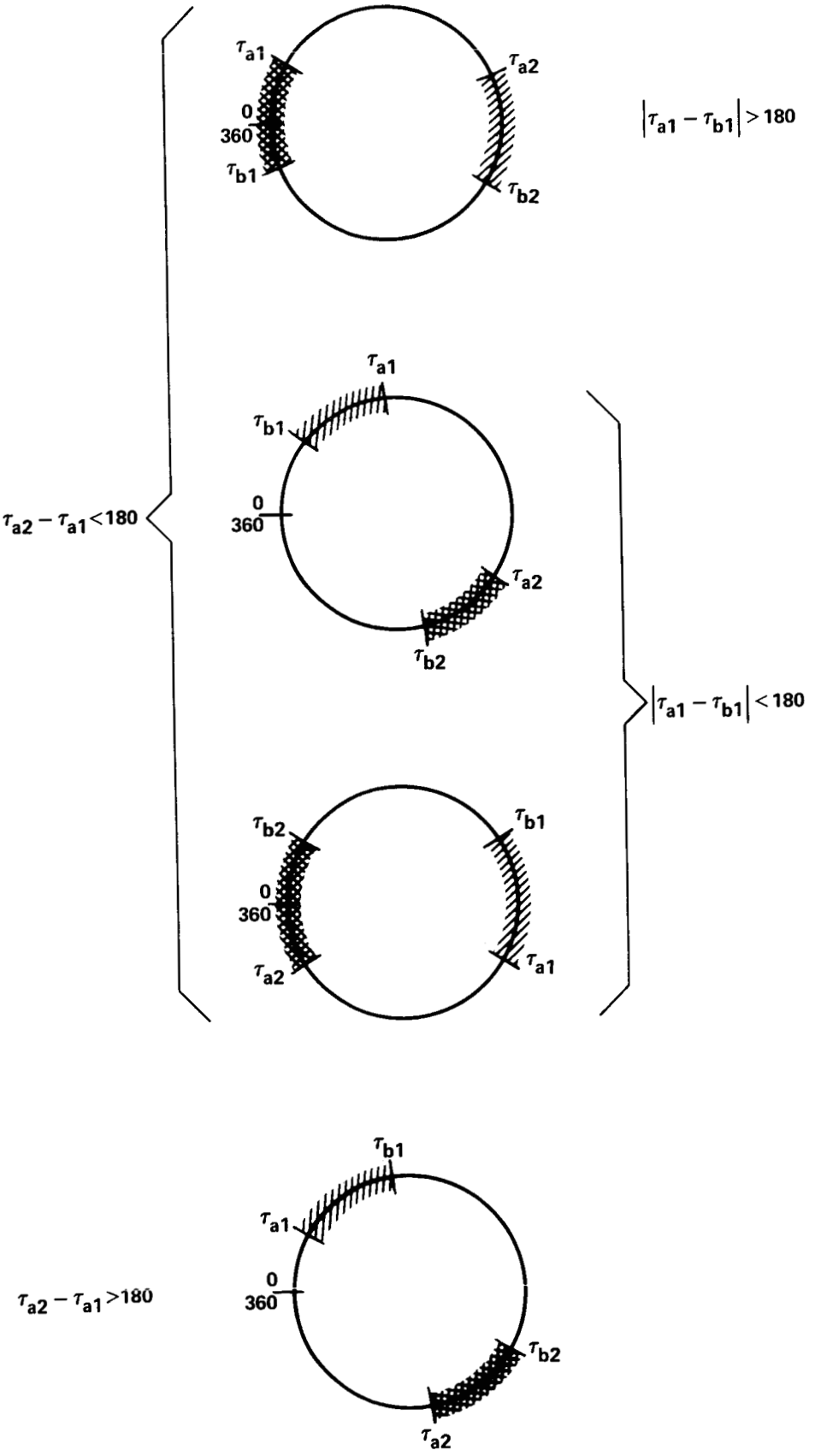


FIGURE 4: POSSIBLE τ LIMITS

BELLCOMM, INC.

Subject: Time Specific Apollo Lunar Surface Accessibility for Relaxed Free Return Missions-Computer Program Description Case 310

From: S. F. Caldwell

Distribution List

Bellcomm, Inc.

D. R. Anselmo
A. P. Boysen, Jr.
J. O. Cappellari, Jr.
D. R. Hagner
W. G. Heffron
N. W. Hinners
J. L. Marshall, Jr.
K. E. Martersteck
J. Z. Menard
P. E. Reynolds
J. W. Timko
R. L. Wagner
All Members Department 2013
Central File
Department 1024 File
Library

Abstract Only to

Bellcomm, Inc.

J. P. Downs
I. M. Ross
M. P. Wilson