

## 5.4 TARGETING ROUTINES

### 5.4.1 GENERAL COMMENTS

The objectives of the targeting routines presented in this section are to provide the LGC capability to determine some of the required input target parameters and set control modes for the various powered flight guidance routines of Section 5.3. The LGC targeting capability represented by the programs described in the following subsections is limited to rendezvous maneuvers and aborts initiated during the powered lunar landing maneuver. These LGC targeting routines are presented in the following subsections.

#### 5.4.2 Rendezvous Targeting

These routines include the following rendezvous maneuver phases:

1. Pre-CSI
2. Pre-CDH
3. Pre-TPI
4. Rendezvous Midcourse Correction

#### 5.4.3 Abort Targeting from Lunar Landing

Initial landing time computations and required descent orbit injection and landing maneuver target parameter computations must be determined by the RTCC. The abort case involving transearth injection SPS backup must also be targeted by the RTCC.

All LGC targeting programs use the Basic Reference Coordinate System defined in Section 5.1.4.1. The basic input parameters required by the targeting routines of this section are the vehicle state vector estimates determined by the navigation programs of Section 5.2. All the Rendezvous Targeting Routines of Section 5.4.2 can be used in either earth or lunar orbit.

## 5. 4. 2 RENDEZVOUS TARGETING

### 5. 4. 2. 1 General

The rendezvous Targeting Programs P-32, P-33, P-34 and P-35 in the LGC are based on the Concentric Flight Plan rendezvous scheme. This scheme illustrated in Fig. 4. 2 -1 employs an active vehicle to ideally make three impulsive maneuvers to establish a rendezvous intercept trajectory with a passive vehicle in an approximately coplanar orbit. Sufficient constraints are imposed to allow only one degree of freedom in choosing the final rendezvous profile. These constraints include the timing of both the first and third maneuvers, the direction of the first maneuver, a unique formulation for the second maneuver and the relative geometry of the two vehicles at the final maneuver point. The second maneuver results in the active vehicle orbit becoming "coelliptic" or "concentric" with the passive vehicle orbit, resulting in the differential altitude between the orbits being approximately constant. The following names have been given to the basic maneuvers of the Concentric Flight Plan rendezvous profile:

- 1) Coelliptic Sequence Initiation (CSI),
- 2) Constant Differential Altitude (CDH),
- 3) Transfer Phase Initiation (TPI).

A rendezvous midcourse correction program is used in addition to the above three primary rendezvous programs to maintain an intercept trajectory from the TPI maneuver until the manual terminal rendezvous is started.

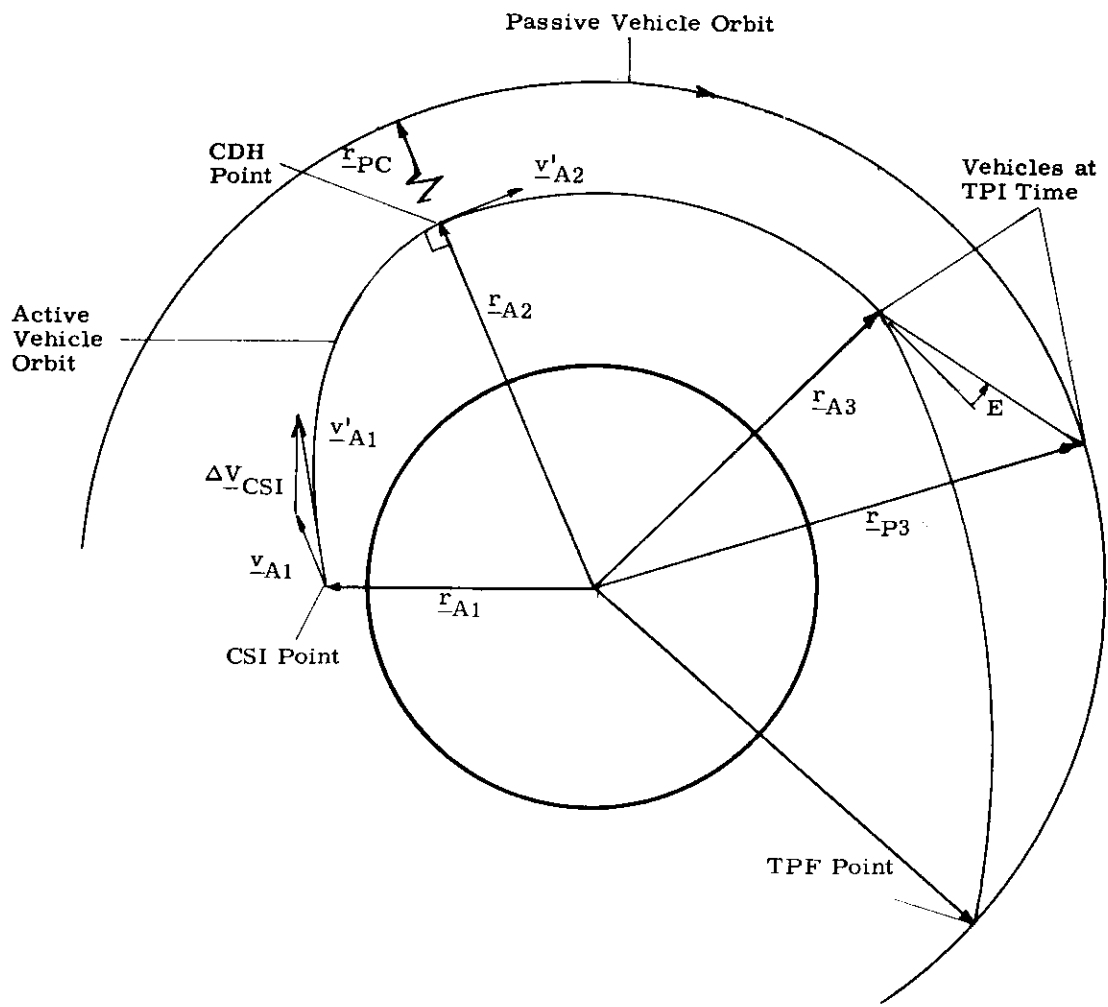


Figure 4.2 -1 Concentric Rendezvous Profile

The LGC Rendezvous Targeting Programs consist of the following programs:

- 1) P-32 Pre-CSI Maneuver Program. This program is called by the astronaut any time before the Co-elliptic Sequence Initiation (CSI) maneuver is performed.
- 2) P-33 Pre-CDH Maneuver Program. This program is called between the CSI and Constant Differential Altitude (CDH) maneuvers.
- 3) P-34 Pre-TPI Maneuver Program. This program is called between the CDH and Transfer Phase Initiation (TPI) maneuvers.
- 4) P-35 Rendezvous Midcourse Maneuver Program. This program is called after the TPI maneuver has been performed.

Each program is designed to select either the LM or CSM as the active vehicle depending on the program number selected by the astronaut. All program operations in Section 5. 4. are independent of which vehicle is chosen as the active vehicle.

A LM direct rendezvous capability is possible using the last two maneuvers in the above list.

All computations performed in the Concentric Rendezvous Program are in the Basic Reference Coordinate System (Section 5. 1. 4. 1) and use the latest state vectors obtained from the Navigation Routines (Sections 5. 2 and 5. 3. 2).

The pre-thrust programs P-32 and P-33 use External  $\Delta V$  guidance (Section 5. 3. 3. 3. 1) and P-34 and P-35 use the Lambert targeting concept. There are 2 options available for obtaining the aim point used in generating the Lambert  $\Delta V$  solution: If the number of offsets (N) is set to zero, the target orbit is advanced through a  $\omega t$  in P34, or the appropriate time of flight in P35, by use of the Kepler (conic) routine. Lambert solution is then generated to this aim point.

The other option (N>0) employs N precision offsets to generate the velocity correction required. (N usually = 2). The target vehicle orbit is advanced the appropriate time of flight on a precision trajectory. The precision offset concept is then used to generate the  $\Delta V$  solution. All four of the above programs use the Cross-Product Steering of Section 5. 3. 3. 4.

5. 4-5

Revised LUMINARY  
 Added GSOP # 567 PCR # 268. 2 Rev. 7 Date 11/69

An alarm code is displayed if the initial conditions along with the astronaut inputs fail to result in a satisfactory solution.

#### 5.4.2.2 Pre-CSI Maneuver

This program, corresponding to Program P-32 (LM active) or P-72 (CSM active) of Section 4, computes the parameters associated with the CSI and CDH maneuvers. The astronaut inputs are:

1. Choice of active vehicle (P-32 LM, P-72 CSM).
2. Time  $t_1$  of the CSI maneuver, (or astronaut may let program compute time of next apoapsis and use it as time of the maneuver.)
3. Number N of the apsidal crossing.  
(If  $N = 1$ , the CDH maneuver occurs when the active vehicle reaches its first apsidal point following the CSI maneuver, etc.)
4. Desired line of sight LOS angle E at the time of the TPI maneuver. (See Fig. 4.2-2 and Fig. 4.2-3).
5. Time  $t_3$  of the TPI maneuver.
6. Selection of flag CDHASW to insure that CDH maneuver will occur at a time, measured from the CSI time, equal to the post CSI maneuver orbital period multiplied by  $N/2$ . If this option is desired, CDHASW is set equal to 1.

The active vehicle state vector  $\underline{r}_A, \underline{v}_A$  and passive vehicle state vector  $\underline{r}_P, \underline{v}_P$  are available in the guidance computer.

The following constraints must be satisfied (see Fig. 4.2-1 and Fig. 4.2-2).

1. The CSI  $\Delta V$  is applied in the horizontal plane of the active vehicle at the CSI time.

5.4-6

Revised LUMINARY  
 Added GSOP # 567 PCR # 695 Rev. 7 Date 11/69

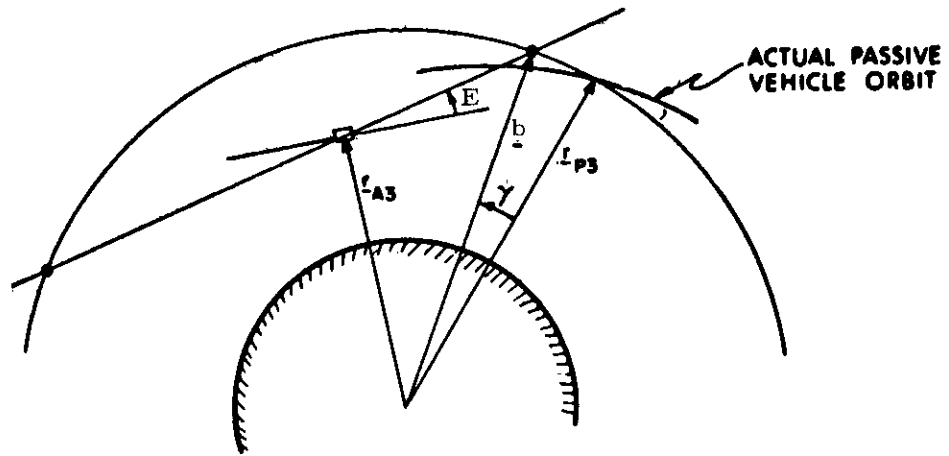


Fig. 4.2 -2a TPI geometry, active vehicle below

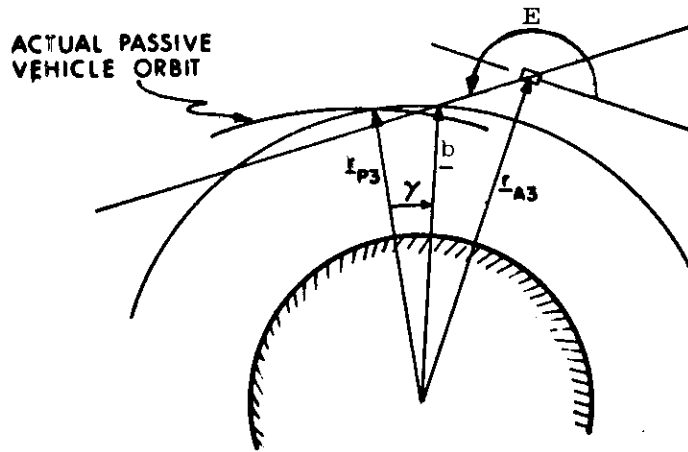
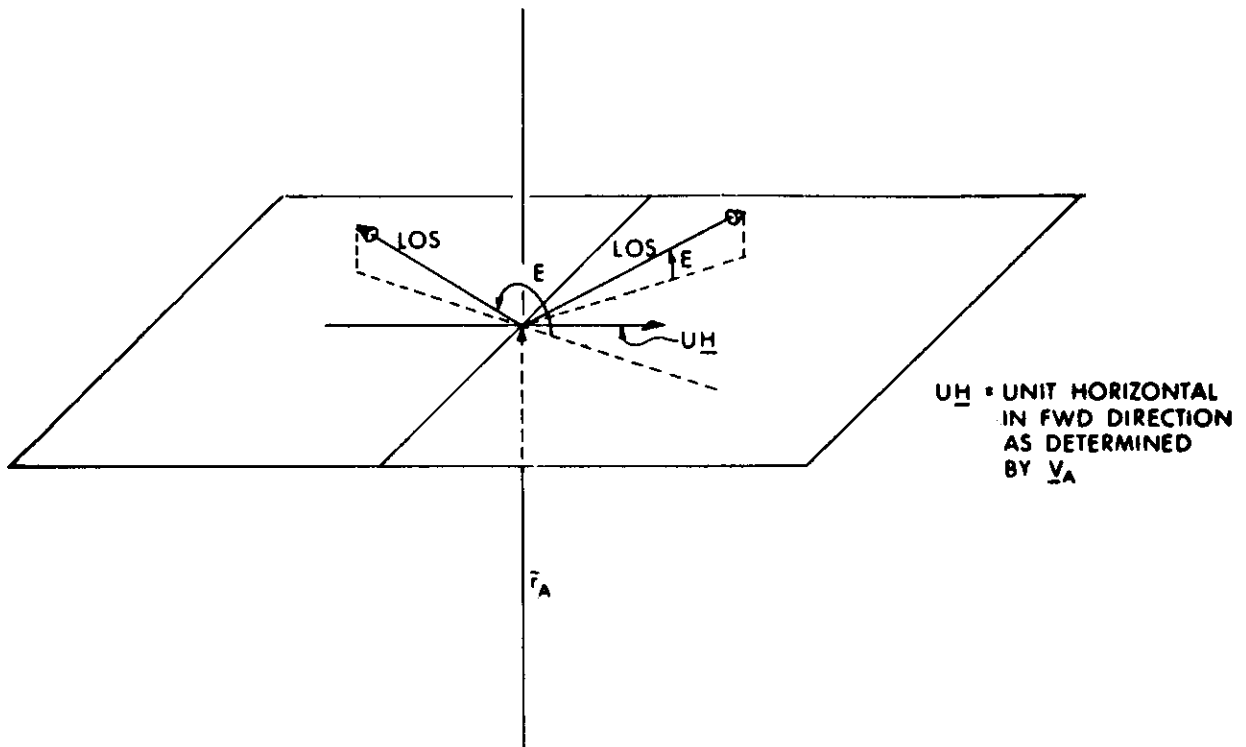


Figure 4.2-2b TPI geometry active vehicle above



- 1) IF THE LOS PROJECTION ON  $\underline{U}_H$  IS POSITIVE:
  - a) WHEN THE LOS IS ABOVE THE HORIZONTAL PLANE  $0 < E < \pi/2$
  - b) WHEN THE LOS IS BELOW THE HORIZONTAL PLANE  $3\pi/2 < E < 2\pi$
  
- 2) IF THE LOS PROJECTION ON  $\underline{U}_H$  IS NEGATIVE
  - a) WHEN THE LOS IS ABOVE THE HORIZONTAL PLANE  $\pi/2 < E < \pi$
  - b) WHEN THE LOS IS BELOW THE HORIZONTAL PLANE  $\pi < E < 3\pi/2$

**Fig. 4.2-3** Definition of Elevation Angle,  $E$



2. The semimajor axis and radial component of velocity of the active vehicle are such that the active and passive vehicles are in coelliptic orbits following the CDH maneuver.
3. The line of sight between the active vehicle and the passive vehicle at the TPI time forms the angle  $E$  with the horizontal plane of the active vehicle.
4. The time intervals between the CSI-CDH and CDH-TPI maneuvers are 10 minutes or greater.
5. After both the CSI and CDH maneuvers, the perigee altitude of the active vehicle orbit is greater than 35,000 ft. for lunar orbits and 85 nm. for earth orbits.

The program solution, as illustrated in the logic flow diagram in Fig. 4.2-4, is based on conic trajectories and contains an iteration loop to select the CSI maneuver magnitude  $v_1$ .

After updating of the active and passive vehicle state vectors to the CSI time, the active vehicle state vector is rotated into the passive vehicle orbital plane resulting in coplanar orbits. The passive vehicle is then updated through the time  $t_3 - t_1$  to the TPI point obtaining  $\underline{r}_{P3}$ ,  $\underline{v}_{P3}$ . The initial guess of  $v_1$ , when added to the velocity of the active vehicle, results in the active vehicle attaining a radius equal to  $r_{P3}$  180 degrees from the CSI point.

If CDHASW equals one, the eccentricity of the active vehicle orbit is less than .0001 or the vertical velocity at the CSI point is less than 7 ft/sec., the time  $t_2$  of the CDH maneuver is set equal to the CSI time plus N times half the period  $t_P$  of the active vehicle. Otherwise, the angle  $\psi$  to the nearest perigee is computed and used in the Time-Theta Subroutine to find the corresponding time of flight  $\Delta t$ . This is then used to calculate  $t_2$  based on the value of N. After verifying that  $t_3$  is greater than  $t_2$ , both vehicles are updated to the CDH point.

The angle  $\phi$  between the active and passive vehicles at the CDH point is calculated and used in the Time-Theta Routine to update the passive vehicle to a point radially above or below the active vehicle, obtaining  $\underline{r}_{PC}$ ,  $\underline{v}_{PC}$ .

The semimajor axis  $a_A$  and the radial component of velocity  $v_{AV}$ , along with a radius, uniquely define the orbit of the active vehicle immediately following the CDH maneuver. The equations for  $a_A$  (based on the semimajor axis  $a_P$  of the passive vehicle) and  $v_{AV}$ , contained in Fig. 4.2-4, therefore specify the "coelliptic" CDH maneuver  $\Delta v_2$ .

The active vehicle state vector is next updated to the TPI time. The unit vector which passes through the active vehicle position and is coincident with the desired TPI line of sight is given by  $\underline{u}_L$ . The position vector of the two points of intersection between the line of sight and a circle (with origin at the center of the attracting body) as shown in Fig. 4.2-2 is defined by:

$$\underline{b} = \underline{r}_{A3} + k \underline{u}_L \quad (4.2.1)$$

where  $k$  normally assumes two values (see Fig. 4.2-4). Equating the magnitude of  $\underline{b}$  to  $r_{P3}$  results in a quadratic expression for  $k$

$$k^2 + 2k \frac{r_{A3}}{r_{L}} \cdot \frac{u_L}{r_{A3}} + r_{A3}^2 - r_{P3}^2 = 0 \quad (4.2.2)$$

If there are no real solutions to the above equation ( $c_2 < 0$  in Fig. 4.2-4), the line of sight does not intersect the circle. If there is a real solution to the above equation, the  $k$  of minimum absolute value is chosen as the desirable TPI solution. The error  $\gamma$  is defined as the central angle between the position vector of the passive vehicle at the TPI point and the position vector  $\underline{b}$  representing where the passive vehicle should be based on the active vehicle location. This error is driven to zero using a Newton-Raphson iteration loop with  $v_1$  as the independent variable. The two initial values of  $v_1$  are 10 ft/sec apart with the succeeding step size  $\Delta v_1$  restricted to 200 ft/sec.

A second computational attempt is made if the first iteration attempt fails.  $v_1$  is incremented in steps of 50 ft/sec. in the direction opposite to that initially taken in the first attempt until the error  $\gamma$  has undergone a sign change. The second attempt then starts with the latest value of  $v_1$  and an adjacent point in the direction of the 50 ft/sec. steps.

Included in the Pre-CSI program are the following seven program checks and the corresponding Alarm Codes AC (see Fig. 4.2-4).

- 1) If on the first iteration there is no solution to the TPI geometry, the iteration cannot be established. The alarm code AC equals A in this case.

2. The iteration counter  $n$  exceeds 15. (AC = C)
3. Two succeeding iterations resulting in  $v_1$  greater than 1000 ft/sec. (AC = B)
4. The altitude of perigee  $h_{P1}$  following the CSI maneuver is less than a minimum acceptable amount. (AC = D)
5. The altitude of perigee  $h_{P2}$  following the CDH maneuver is less than a minimum acceptable amount. (AC = E)
6. The time  $\Delta T_2$  between the CDH and CSI maneuver is less than 10 minutes. (AC = F)
7. The time  $\Delta T_3$  between the TPI and CDH maneuvers is less than 10 minutes. (AC = G)

For the first check above, a check failure results in an immediate program exit. The last four checks are made only if the iteration has succeeded. The alarm codes are set (excluding the first check) only if the second iteration attempt has been made.

The displays for the Pre-CSI maneuver program are:

1. Differential altitude  $\Delta H_{CDH}$  at the CDH point  
 $\Delta H_{CDH}$  is positive when the active vehicle's altitude is less than the passive vehicle altitude when measured at the CDH point.
2. Time  $\Delta T_2$  between the CDH and CSI maneuvers displayed in minutes and seconds with hours deleted.
3. Time  $\Delta T_3$  between the TPI and CDH maneuvers displayed in minutes and seconds with hours deleted.
4.  $\Delta \underline{V}_{CSI}(LV)$  in local vertical coordinates at CSI time.
5.  $\Delta \underline{V}_{CDH}(LV)$  in local vertical coordinates at CDH time
6. The alarm codes

In the Pre-CSI program P-32, the astronaut can change the CSI computed maneuver by over writing the displayed maneuver velocity vector in local vertical coordinates,  $\Delta \underline{V}_{\text{CSI}}$  (LV). In such a case the astronaut input then becomes the defined velocity-to-be-gained in the External  $\Delta V$  Maneuver Guidance (Section 5.3.3.3.1) used to control the CSI maneuver. The over writing procedure is normally used to correct or minimize out-of-plane conditions which are not directly controlled by the CSI or CDH maneuvers computations such that minimum out of plane conditions exist at the TPI maneuver.

In addition, M the number of navigation measurements processed since the last maneuver, MGA the middle gimbal angle at the start of the maneuver assuming plus X acceleration, and TFI the time between the CSI maneuver and the present time are displayed.

The middle gimbal angle is computed with the following equation:

$$\text{MGA} = \sin^{-1} \left[ \underline{Y}_{\text{REF}} \cdot \text{UNIT}(\Delta \underline{V}) \right]$$

If  $-90^\circ \leq \text{MGA} < 0^\circ$ , MGA is redefined such that  $270^\circ \leq \text{MGA} < 360^\circ$ .  $\Delta \underline{V}$  is the maneuver  $\Delta \underline{V}$  in reference coordinates and  $\underline{Y}_{\text{REF}}$  is the second row of the REFSSMMAT matrix defined in Section 5.6.3.4.

#### 5.4.2.3 Pre-CDH Maneuver

This program, corresponding to Program P-33 (LM active) or P-73 (CSM active) of Section 4, computes the parameters associated with the CDH maneuver.

The astronaut inputs to this program are:

1. Choice of active vehicle (P-33 LM, P-73 CSM)
2. Time  $t_2$  of CDH maneuver

The active and passive vehicle's state vectors  $\underline{r}_A$ ,  $\underline{v}_A$ ,  $\underline{r}_P$ ,  $\underline{v}_P$  are available in the guidance computer. The TPI elevation angle  $E$  and the TPI maneuver time  $t_3$  are also available from the previous Pre-CSI program computations.

Figure 4.2-4, page 4 of 7, illustrates the program logic. The state vectors of both the active and passive vehicles are advanced to the CDH time using the Coasting Integration Routine. After rotating the active vehicle state vector into the plane of the passive vehicle, the CDH maneuver is calculated as in the Pre-CSI program. Following a precision update of the state vectors to time  $t_3$ , the Pre-TPI program is used to calculate the time  $t$  at which the specified elevation angle is attained. If the iteration was not successful, an alarm code is displayed. At this point, the astronaut can elect to recycle the program or to proceed with the calculation of the displays, which are:

1.  $\Delta \underline{V}_{CDH}^{(LV)}$
2. Differential altitude  $\Delta H_{CDH}$  at the CDH point.
3. Time  $\Delta T_3$  between the calculated TPI time and the CDH time displayed in minutes and seconds with hours deleted.
4. Time  $\Delta T_{TPI}$  between the calculated TPI time and the specified TPI time. (This number is positive if the new TPI time is later than that previously used.)

If the iteration did not converge, the last two displays reflect an unchanged TPI time. In addition, M, MGA and TFI are displayed as in the Pre-CSI program displays.

The astronaut can modify the CDH maneuver to minimize out-of-plane conditions by overwriting the displayed maneuver velocity vector,  $\Delta \underline{V}_{CDH}^{(LV)}$ , as in the pre-CSI program. The out-of-plane information used in this process is determined by the astronaut with the aid of the Out of Plane Rendezvous Display Routine, R36, of Section 5.6.7.3.

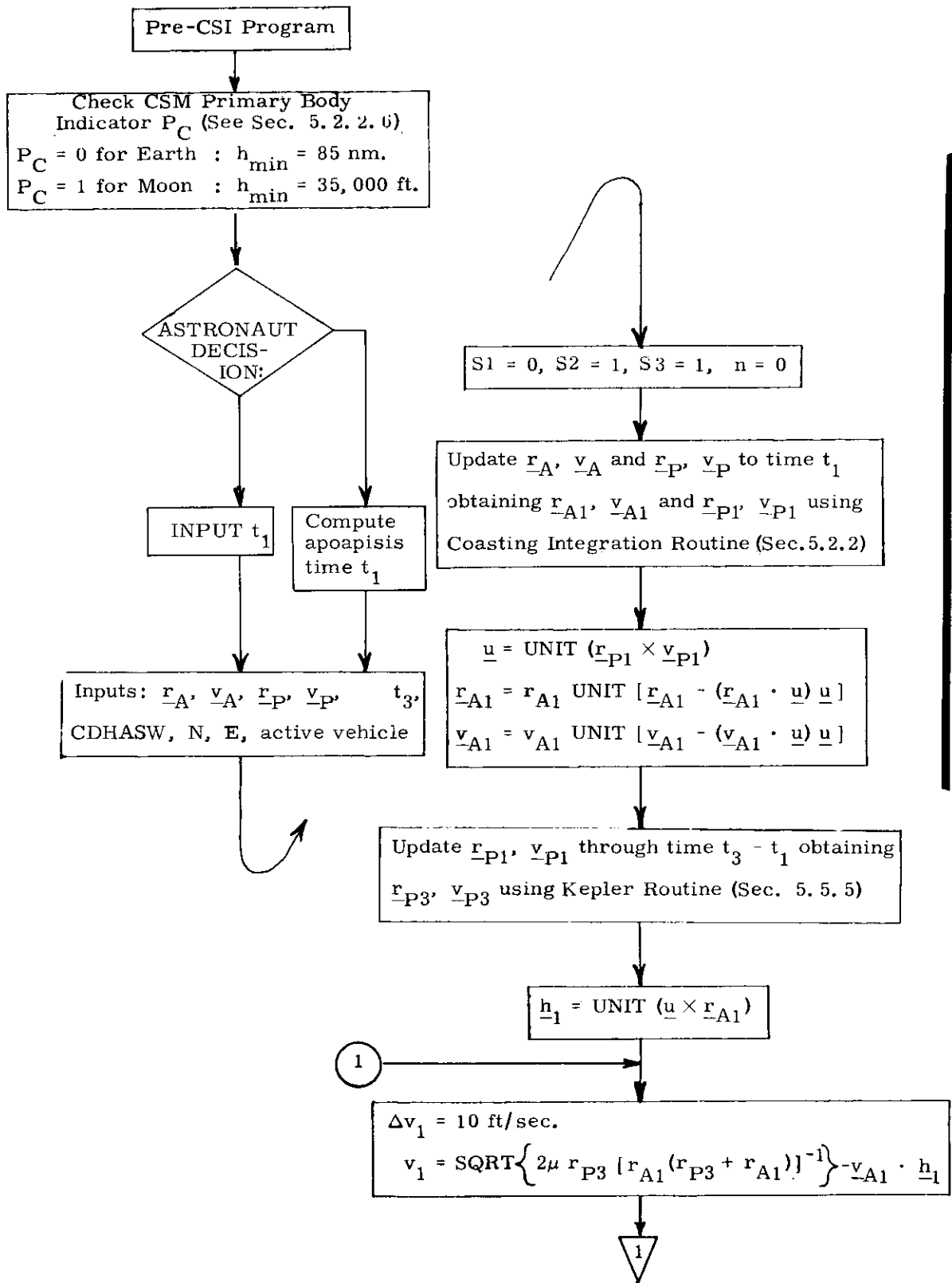


Figure 4.2-4 Pre-CSI-CDH Maneuver Program  
(page 1 of 7)

5.4-15

Revised LUMINARY  
 Added GSOP # 567 PCR # 695 Rev. 7 Date 11/69

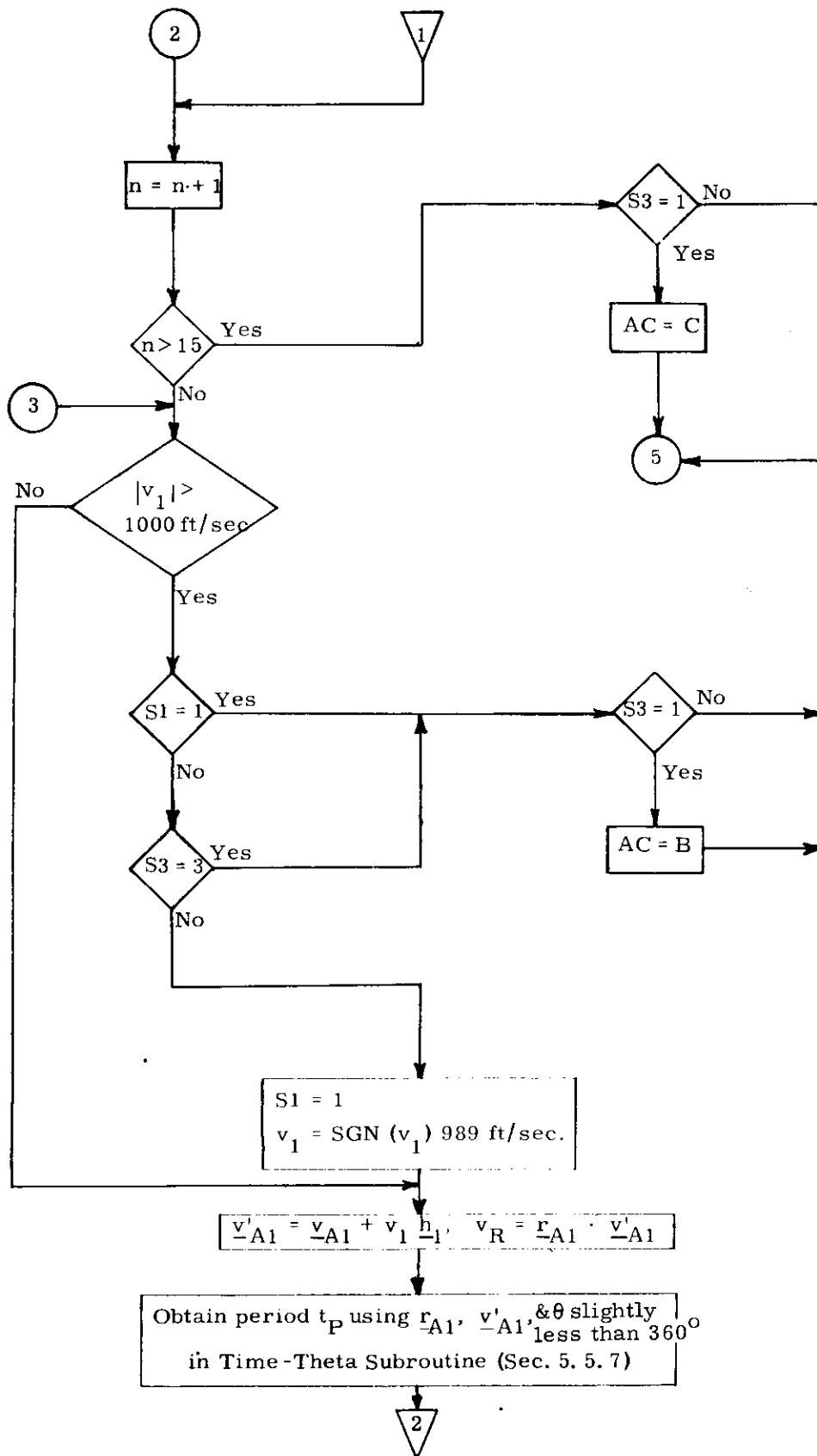


Figure 4.2-4 Pre-CSI-CDH Maneuver Program  
(page 2 of 7)



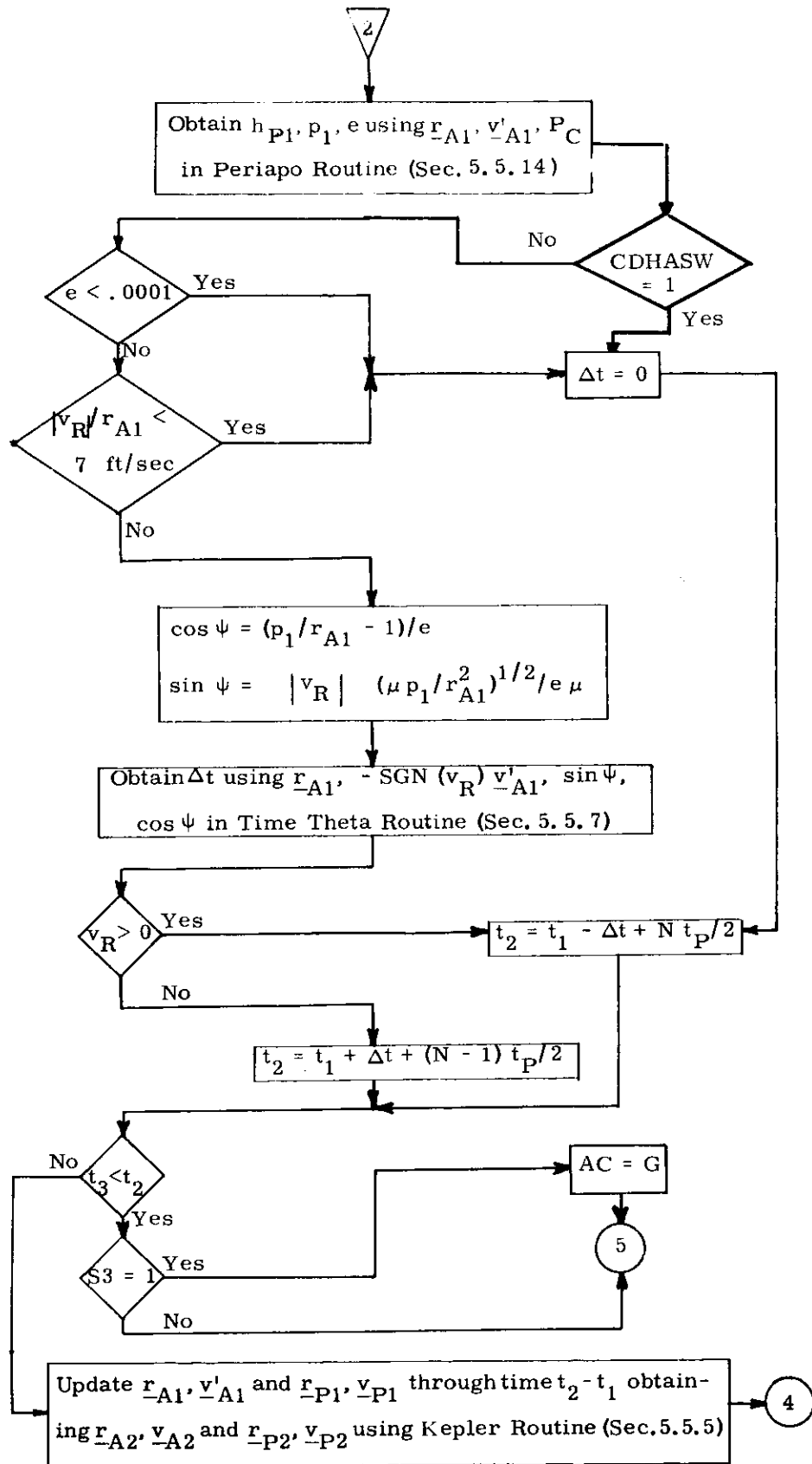


Figure 4.2-4 Pre-CSI-CDH Maneuver Program

(page 3 of 7)

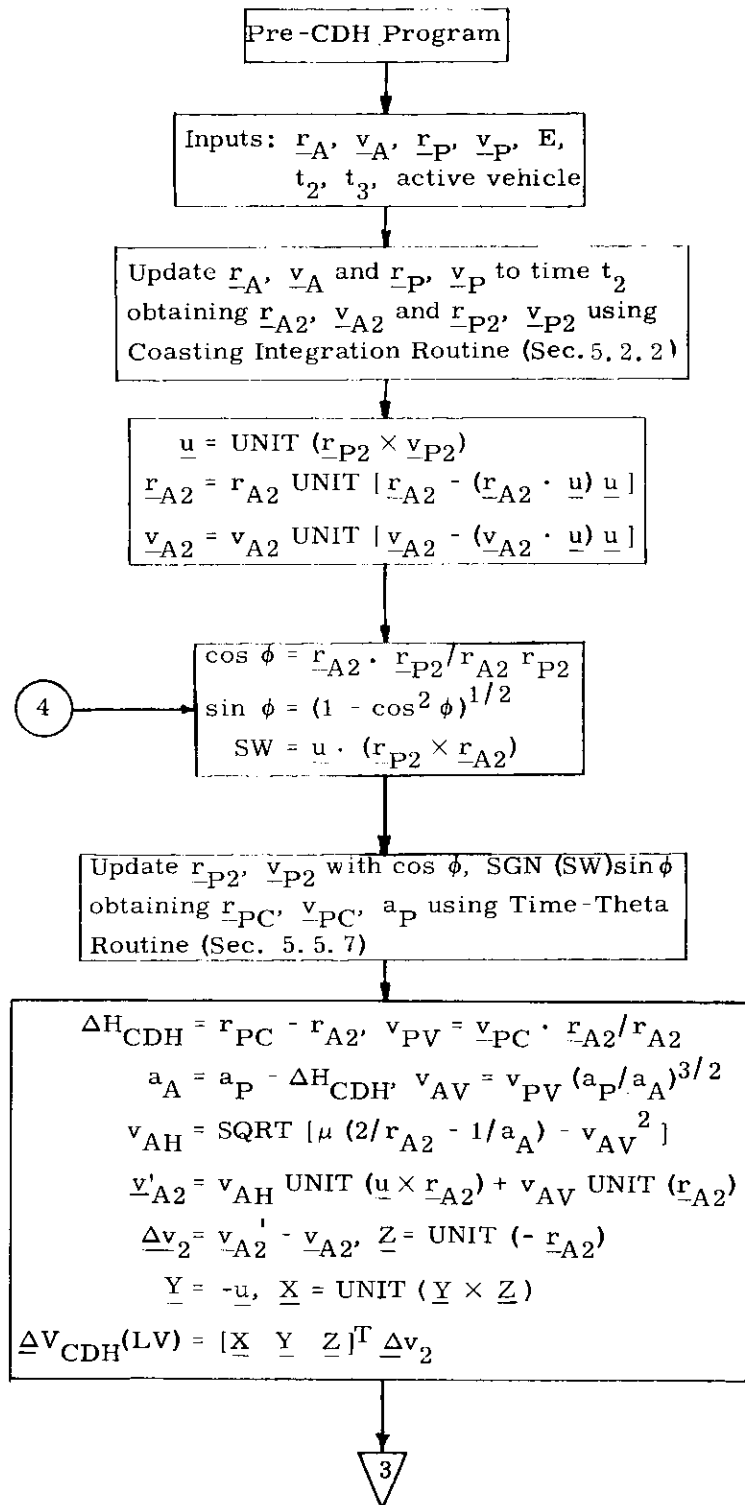


Figure 4.2-4 Pre-CSI-CDH Maneuver Program  
 (page 4 of 7)

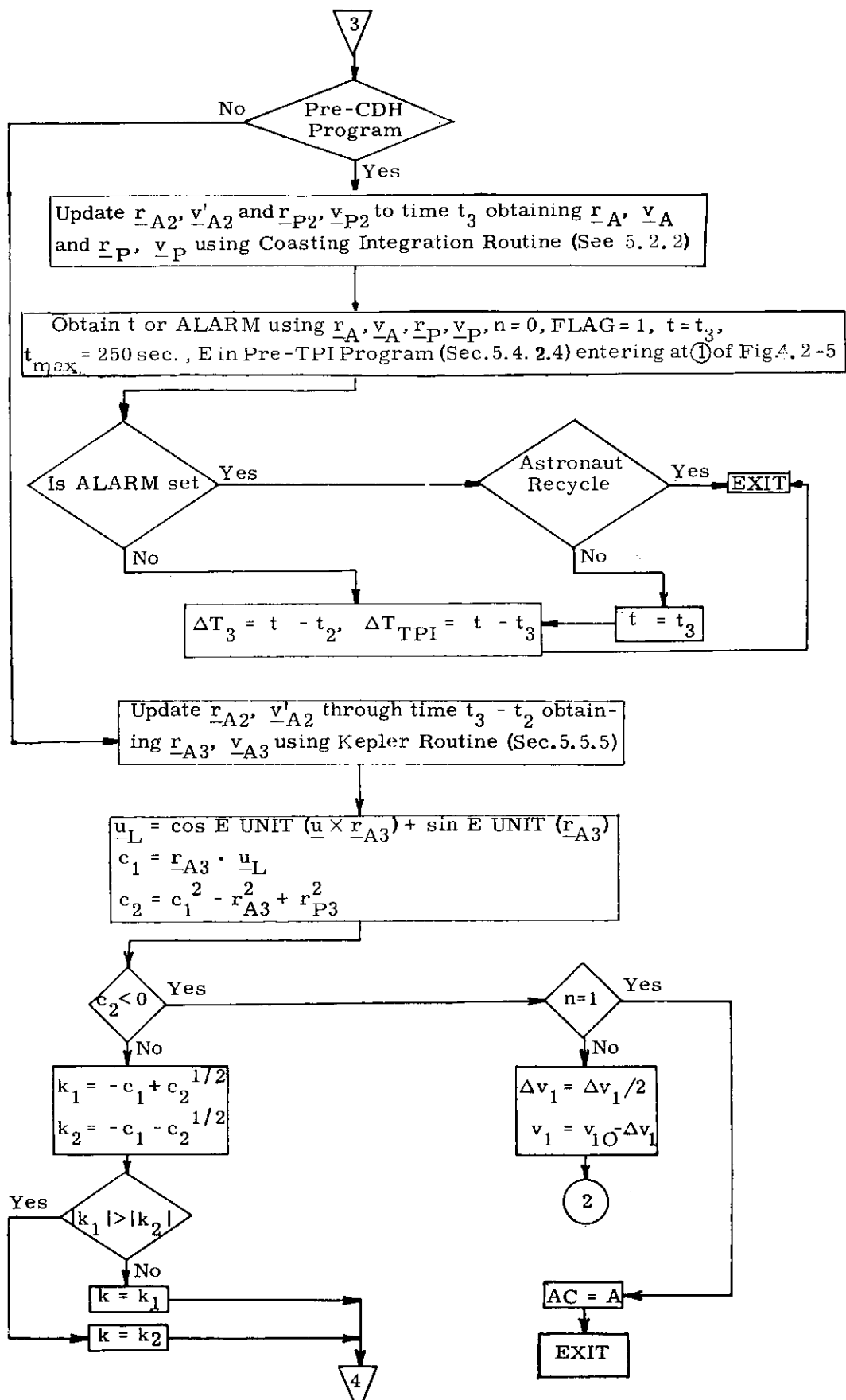


Figure 4.2-4 Pre-CSI-CDH Maneuver Program

(page 5 of 7)

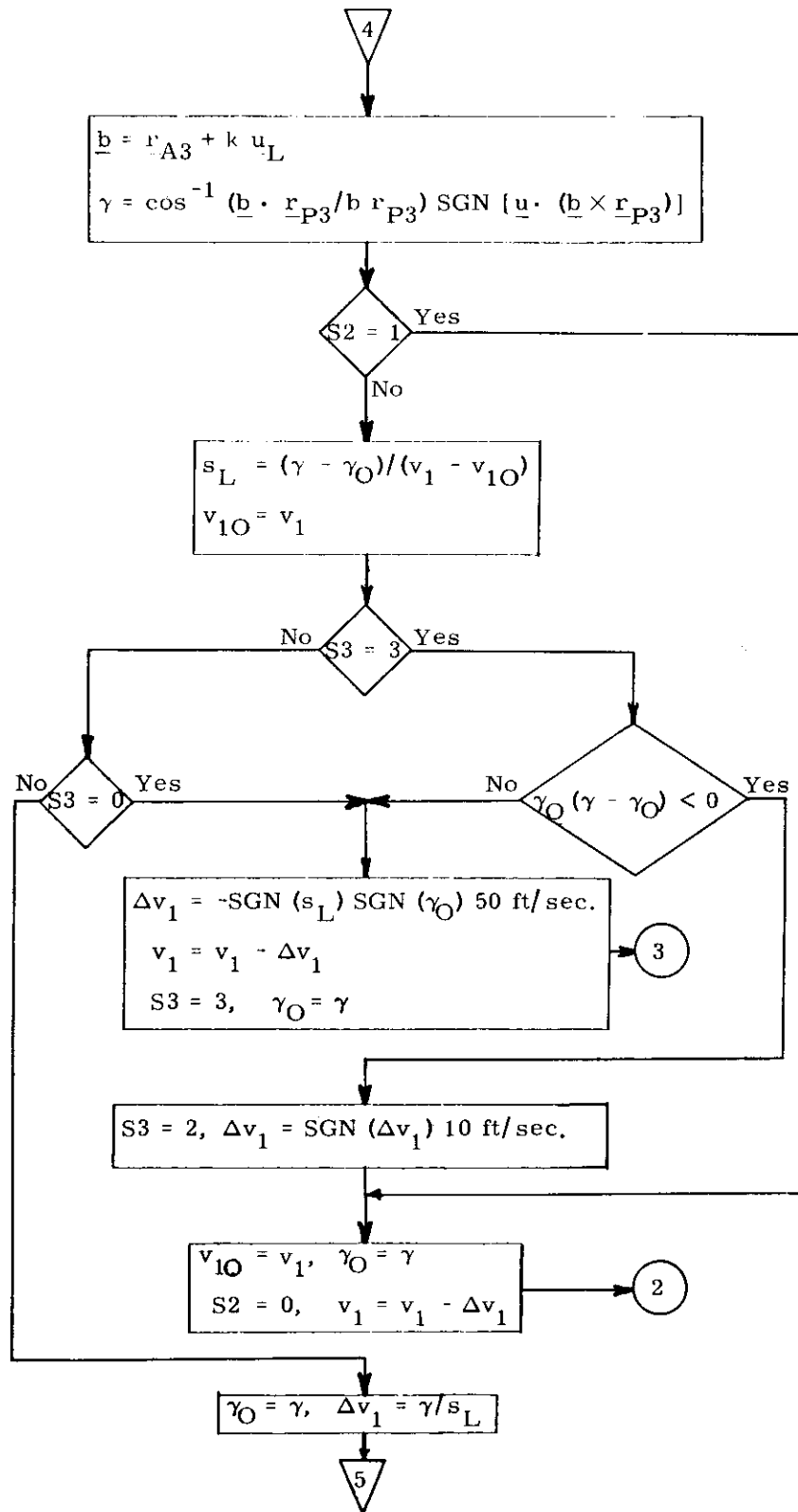


Figure 4.2-4 Pre-CSI-CDH Maneuver Program  
(page 6 of 7)

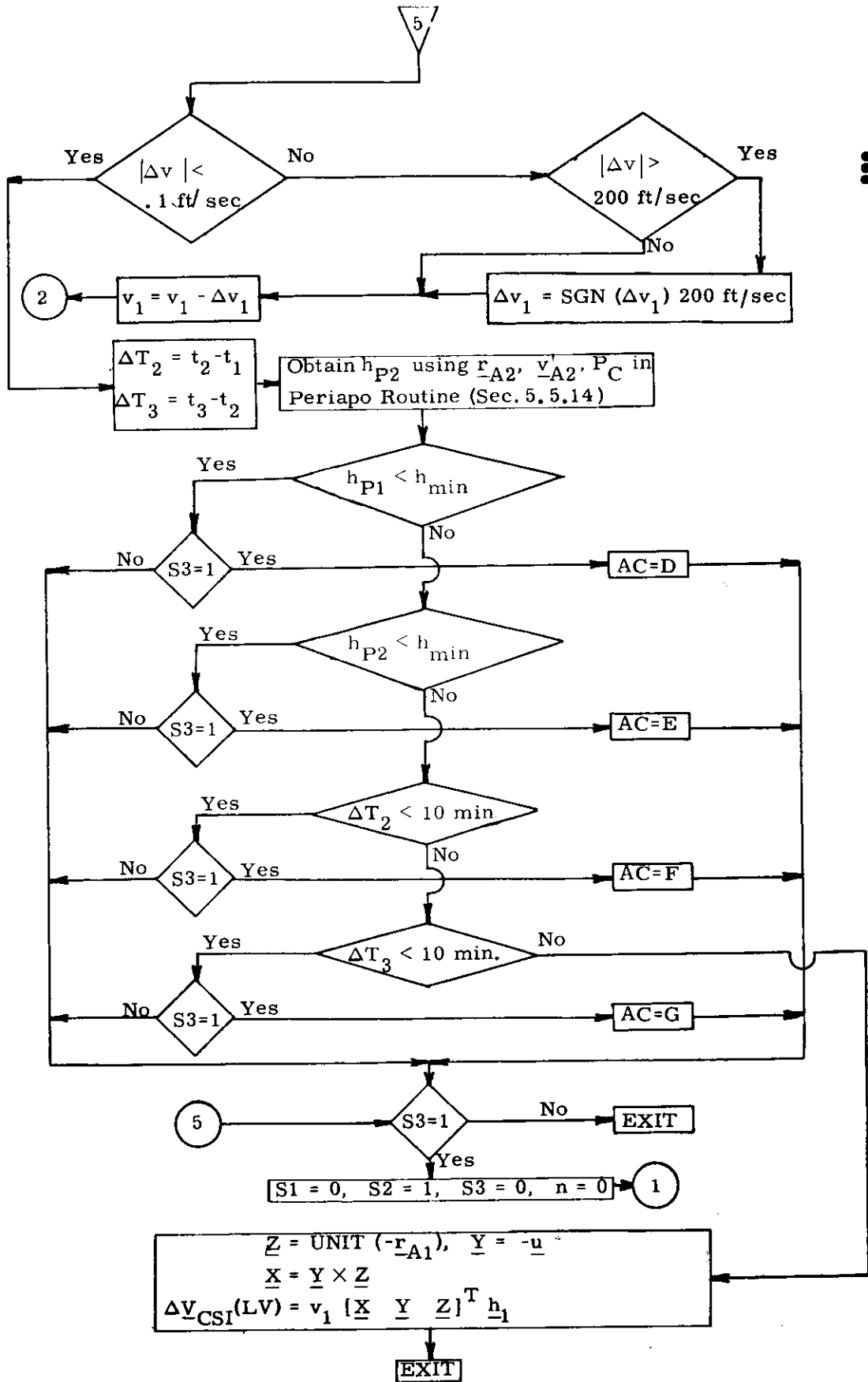


Figure 4.2-4 Pre-CSI-CDH Maneuver Program  
(page 7 of 7)  
5.4-21

#### 5.4.2.4 Pre-TPI Maneuver

This program corresponds to Program P34 (LM active) or P74 (CSM active) of Section 4. Its objective is to establish the terminal phase initiation (TPI) maneuver. The position of the TPI maneuver is determined by specifying either the TPI time or the elevation angle which specifies the relative geometry of the vehicles at the TPI point. The astronaut inputs are:

1. Choice of active vehicle (P34 LM, P74 CSM)
2. Time  $t$  of the TPI maneuver
3. Elevation angle  $E$  (set equal to zero if  $t$  is specified) defined in Fig. 4.2-3
4. Central angle  $\omega t$  of the passive vehicle between the TPF and TPI points
5. Number  $N_1$  of precision offsets made in generating the target vector for the TPI maneuver. If set equal to zero, conic trajectories are used to generate the target vector.

The active  $\underline{r}_{AI}$ ,  $\underline{v}_{AI}$  and passive vehicle  $\underline{r}_{PI}$ ,  $\underline{v}_{PI}$  state vectors are available in the guidance computer. The program starts with a precision update of these vectors to the TPI time.

If the elevation angle is not specified, it is computed after the precision updating of both vehicles to the TPI maneuver time. If  $E$  is specified, an iteration procedure is initiated to find the TPI time at which  $E$  is attained. This procedure uses the input TPI time as the initial guess and is based on conic trajectories (except for the original precision update). The time correction  $\delta t$  is based on (1) the angular distance between  $\underline{r}_P$  and the desired position of the passive vehicle, obtained by assuming the passive vehicle is in a circular orbit, and (2) assuming the vehicles are moving at a constant angular rate. The flow diagram for this procedure is shown in Fig. 4.2-5.

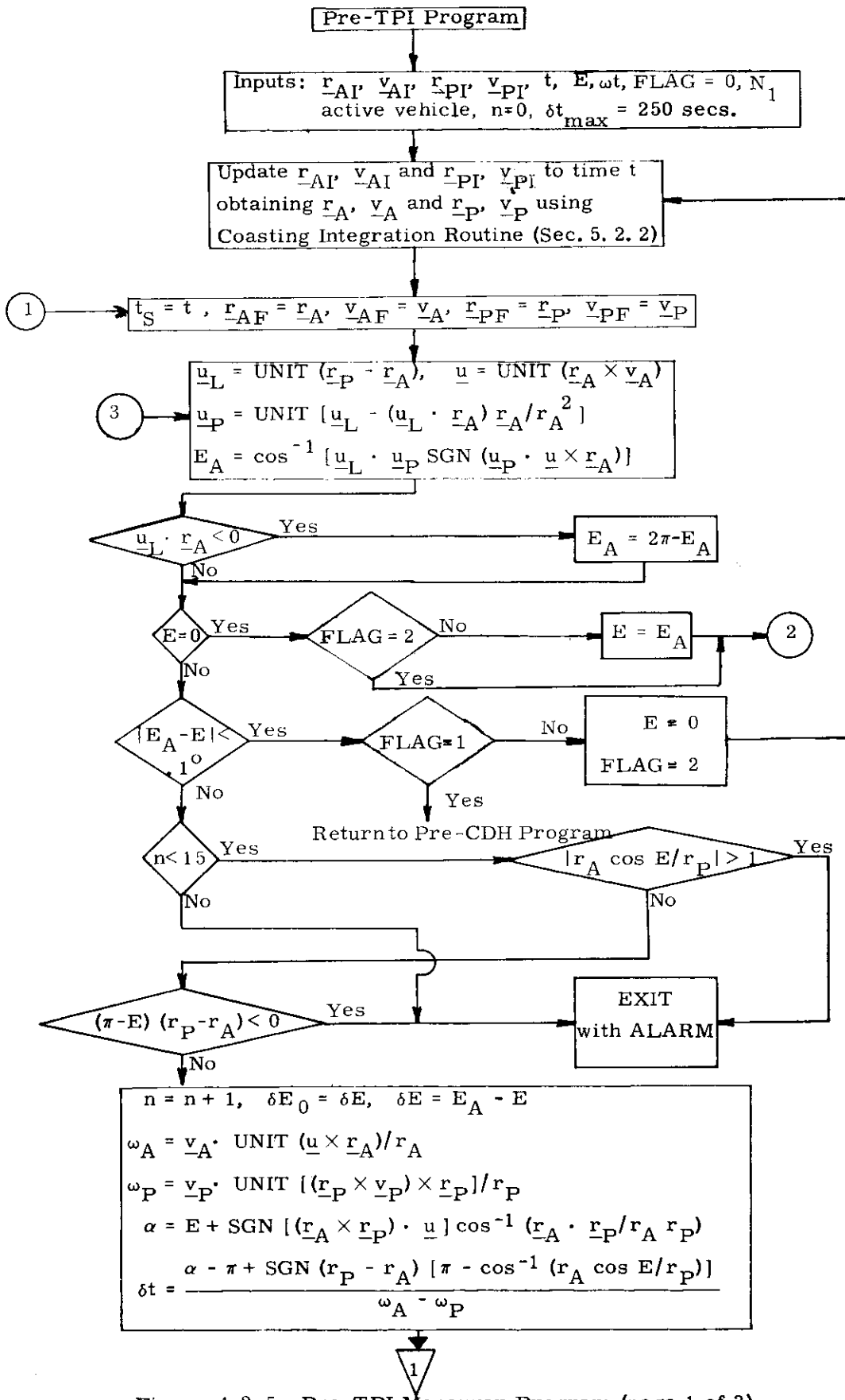


Figure 4.2-5 Pre-TPI Maneuver Program (page 1 of 3)

5.4-23

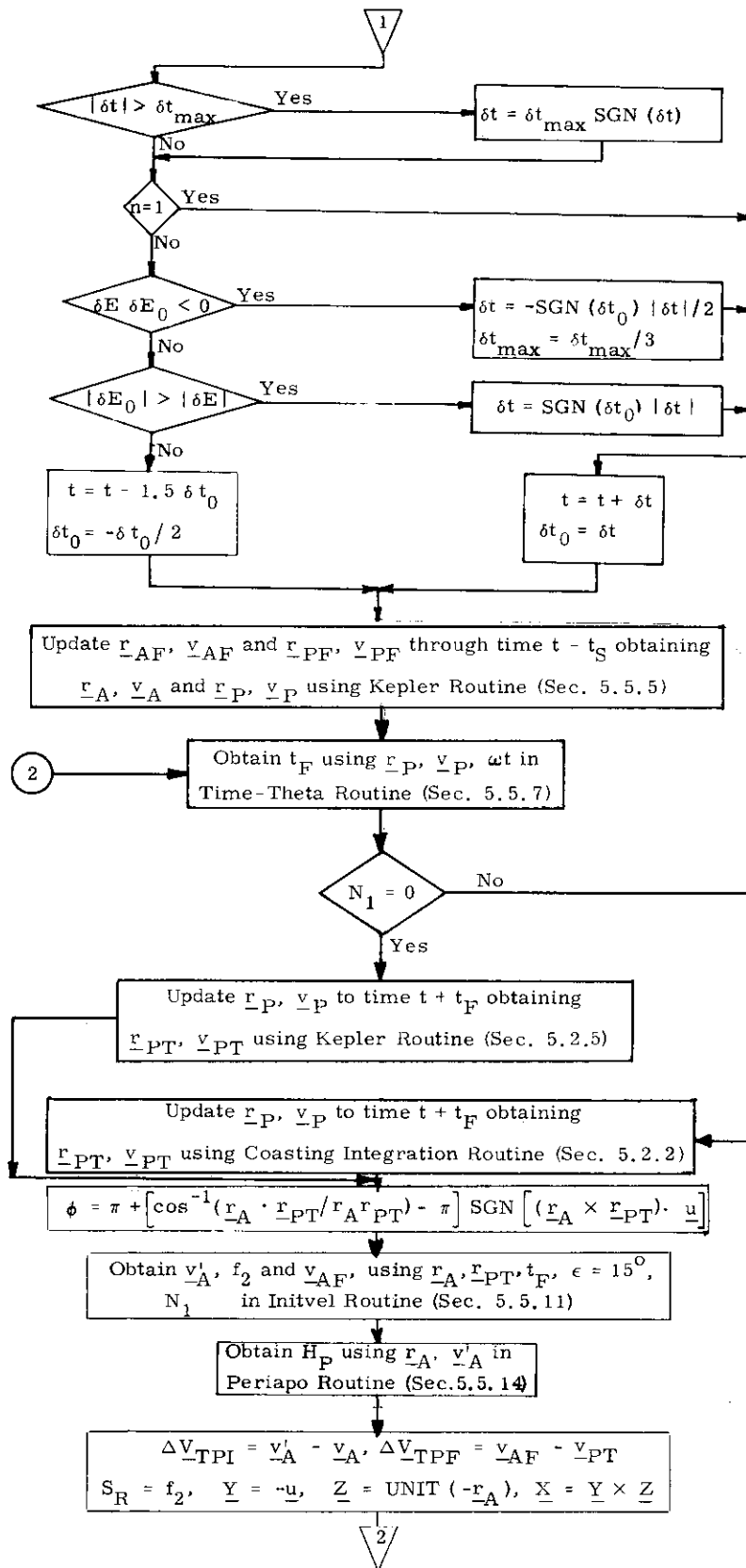


Figure 4.2-5 Pre-TPI Maneuver Program (page 2 of 3)



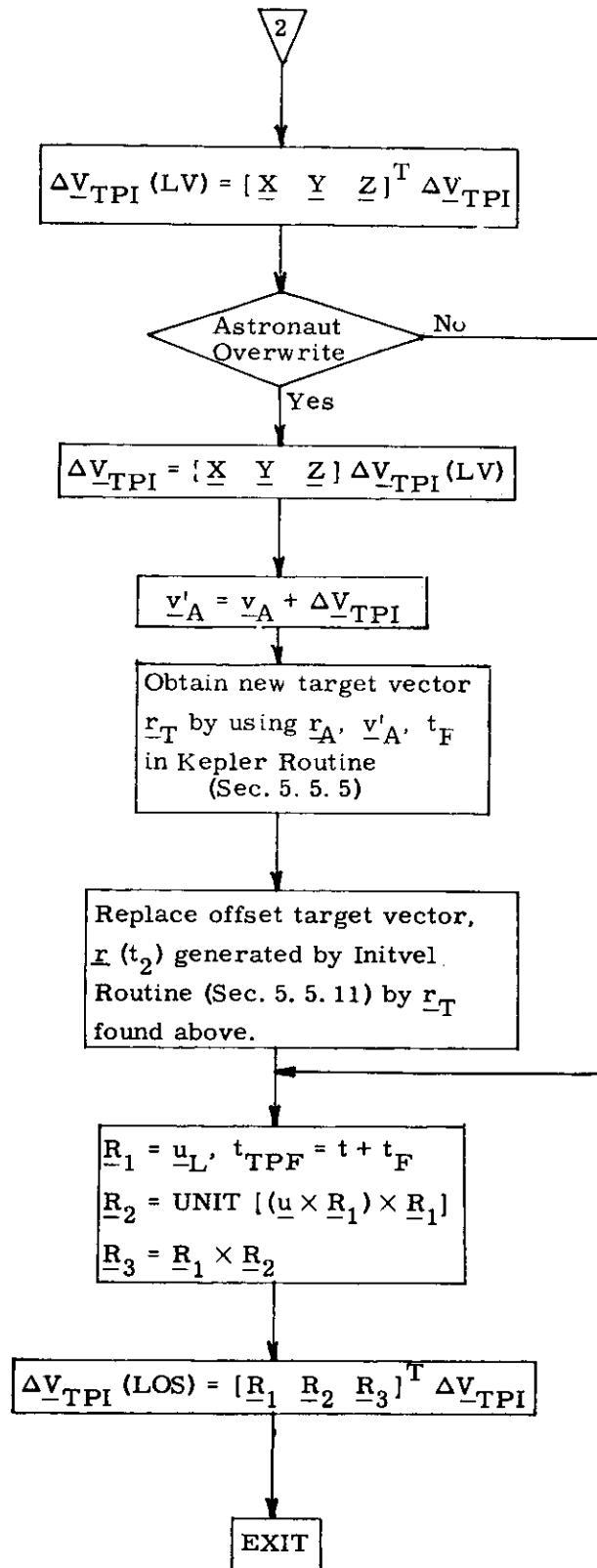


Figure 4.2-5 Pre-TPI Maneuver Program (page 3 of 3)

The iteration is successful when the computed elevation angle  $E_A$  is sufficiently close to  $E$ .

To help insure convergence, the following steps are taken:

- a. The step size  $\delta t$  is restricted to 250 secs.
- b. If the solution has been passed ( $\delta E \delta E < 0$ ) the step size is halved and forced in the opposite direction of the last step.
- c. If the iteration is converging ( $|\delta E_0| - |\delta E| > 0$ ), the sign of  $\delta t$  is maintained.
- d. If the iteration is proceeding in the wrong direction, the step direction is reversed.

The iteration is terminated for any of the following reasons, and a single alarm as described in Section 4 is set:

1. The iteration counter  $n$  has exceeded its maximum value of 15.
2. The line of sight emanating from the active vehicle does not intersect the circular orbit with radius equal to that of the passive vehicle.
3. The elevation angle is inconsistent with the relative altitudes of the two vehicles (e. g., if the elevation angle is less than  $180^\circ$  when the active vehicle is above the passive vehicle).

Upon convergence, the state vectors are precision updated to the TPI time.

The TPI-TPF phase of the program starts with the use of the angle  $\omega t$  in the Time-Theta Subroutine (Section 5.5.7) to determine the corresponding transfer time  $t_F$ . The passive vehicle is next updated through  $t_F$  using either a conic or precision updating routine, depending on the value of  $N_1$ . The Initvel Routine (Section 5.5.11) is then called to compute a transfer trajectory between the TPI and TPF points with the cone angle  $\epsilon$  set equal to  $15^\circ$ . The rotation projection switch  $f_2$  is obtained from the Initvel Routine for use in the powered flight steering program as  $S_R$ .

The central angle  $\Phi$  traversed by the active vehicle from the TPI time to intercept is computed as shown in Fig. 4.2-5 for display if requested by the astronaut. This display is used to avoid  $180^\circ$  transfer angle problems.

The displays for the Pre-TPI Program are:

1.  $\Delta V_{TPI}$
2.  $\Delta V_{TPF}$
3.  $\Delta \underline{V}_{TPI}(\text{LOS})$  : Line of sight coordinates
4.  $\Delta \underline{V}_{TPI}(\text{LV})$  : Local vertical coordinates
5. Perigee altitude ( $H_P$ ) following the TPI maneuver

The display of  $\Delta \underline{V}_{TPI}(\text{LV})$  may be overwritten by the astronaut. If so, a new target vector is generated as shown in Fig. 4.2-5. This new aimpoint is used for powered flight steering.

In addition, M, MGA and TFI are displayed as in the Pre-CSI program displays.

This program, corresponding to Program P35 (LM active) or P75 (CSM active) of Section 4, computes a midcourse correction maneuver. This maneuver insures that the active vehicle will intercept the passive vehicle at the time established in the previous Pre-TPI program. The astronaut may call this program any time after the TPI maneuver, but in general no later than 10 minutes before the intercept time. The flow diagram for this program is shown in Fig. 4.2-6.

There is one astronaut input: choice of the active vehicle (P35 LM, P75 CSM). The intercept time  $t_{\text{TPI}}$  and the number  $N_1$  of precision offsets, both available from the Pre-TPI program, the active  $\underline{r}_A, \underline{v}_A$  and passive  $\underline{r}_P, \underline{v}_P$  vehicle state vectors and a time delay are all available in the guidance computer. The time delay ( $\delta\tau_3$  for P35 and  $\delta\tau_7$  for P75) is the time required to prepare for the thrust maneuver and is stored in either of two erasable locations.

When the program is initiated, the number of navigation measurements since the last maneuver is displayed. In addition, the time since the last maneuver or last maneuver computation is also displayed. Based on this information and additional displays discussed in Section 5.6.7, the astronaut may elect to proceed with the midcourse maneuver at some time. When he does so, the program updates the state vectors to the present time plus the time delay using the Coasting Integration Routine. After updating the passive vehicle to the time  $t_{\text{TPI}}$  to obtain the target vector  $\underline{r}_{\text{PT}}$ , the Initvel Routine is then called, with the cone angle  $\epsilon$  set equal to  $15^\circ$ , to obtain the velocity vector on the transfer ellipse. The rotation projection switch  $f_2$  is obtained from the Initvel Routine for use in the powered flight steering program as  $S_R$ .

The central angle  $\phi$  traversed by the active vehicle from the maneuver time to intercept is computed as shown in Fig. 4.2-6 for display if requested by the astronaut. This display is used to avoid  $180^\circ$  transfer angle problems.

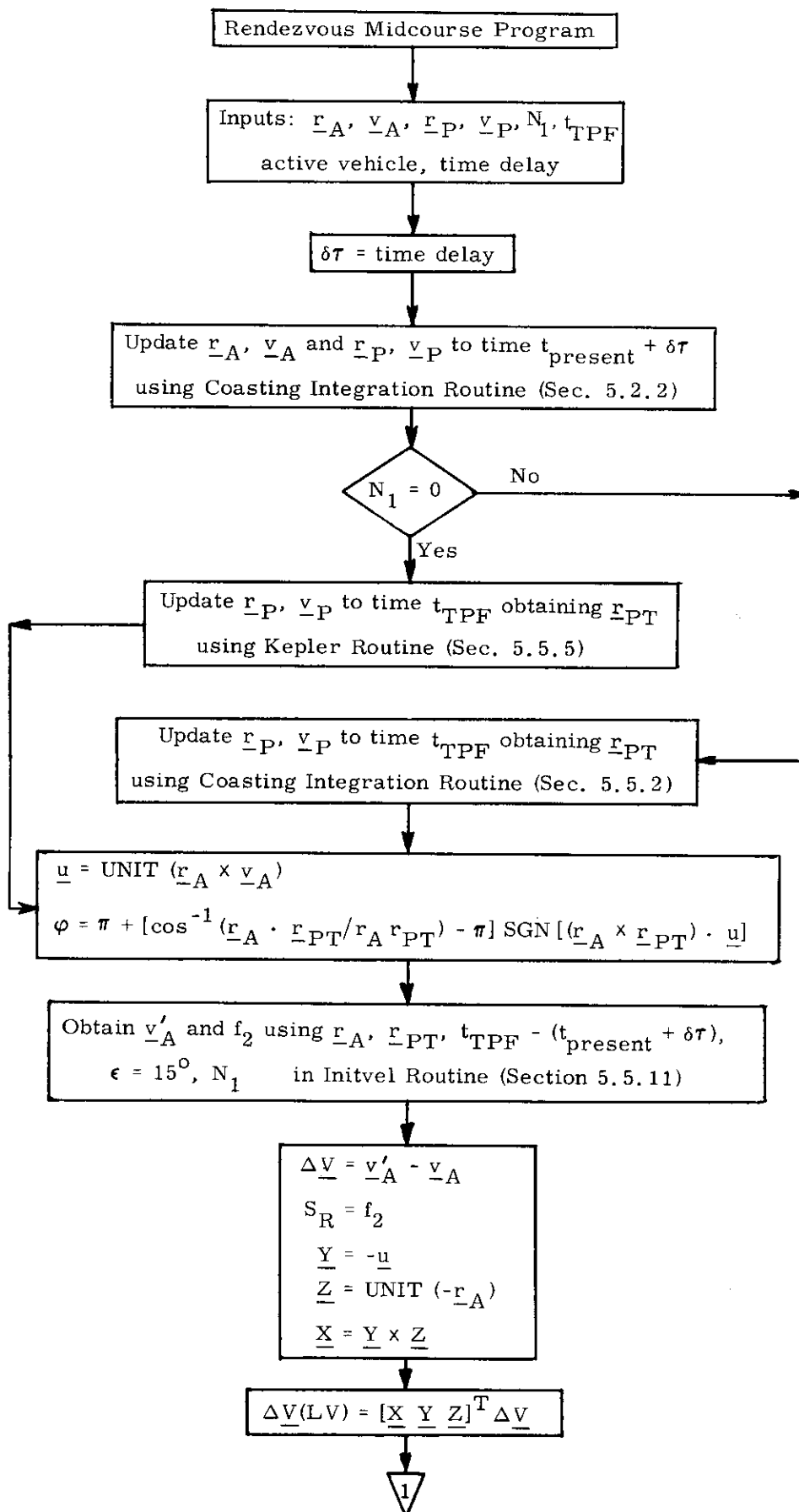


Figure 4.2-6 Rendezvous Midcourse Program (page 1 of 2)

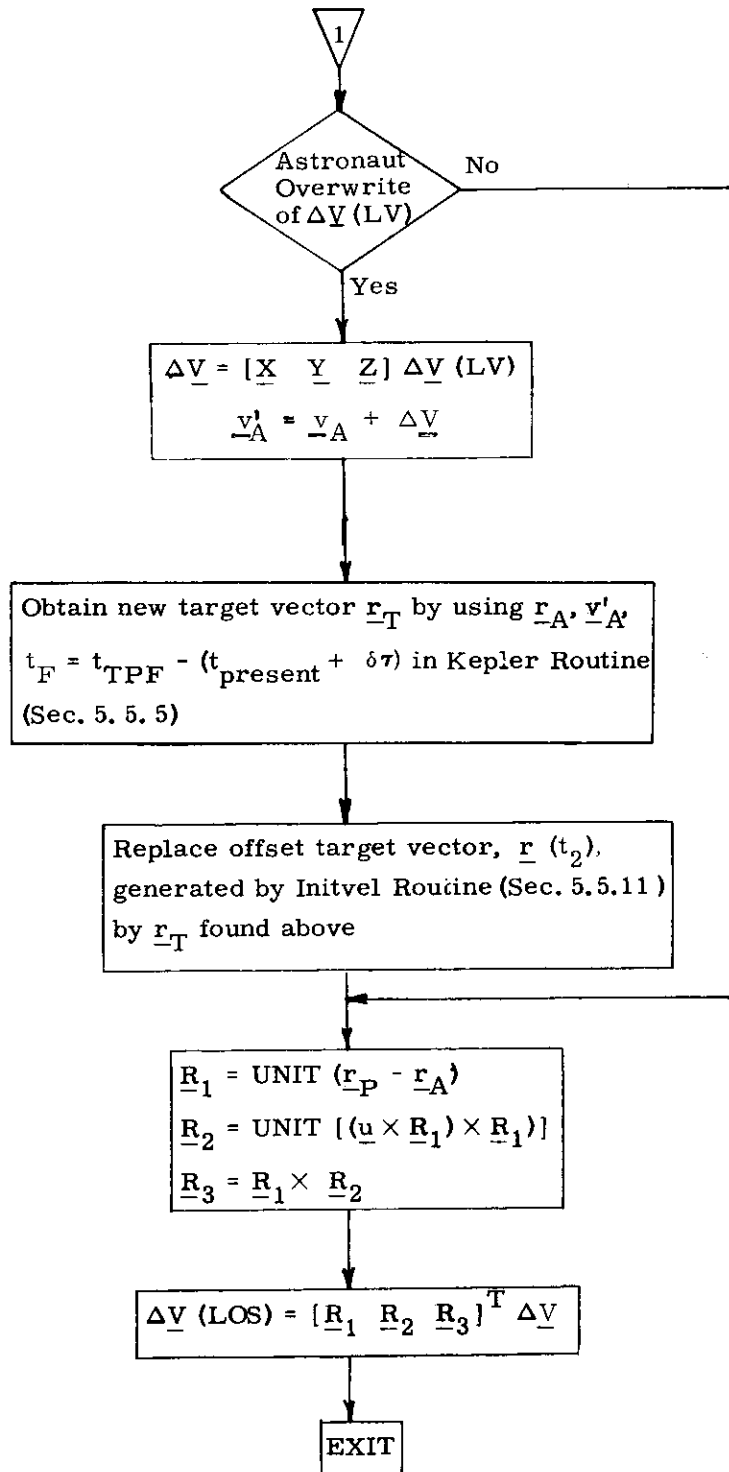


Figure 4.2-6 Rendezvous Midcourse Program  
(page 2 of 2)

After obtaining the maneuver  $\Delta V$ , it is rotated into a local vertical (LV) and a line of sight (LOS) coordinate system obtaining  $\Delta V$  (LV) and  $\Delta V$  (LOS), and both are displayed.

The display of  $\Delta V$  (LV) may be overwritten by the astronaut. If so, a new target vector is generated as shown in Fig. 4.2-6. This new aimpoint is used for powered flight steering.

In addition, M, MGA and TFI are displayed as in the Pre-CSI program displays.

(Pages 5.4-32 through 5.4-38 inclusive deleted by PCR # 844.)



### 5.4.3 ABORT TARGETING

#### 5.4.3.1 Aborts from Powered Landing

##### 5.4.3.1.1 General Objectives

The objective of the DPS and APS Abort Targeting and Control Programs, P70 and P71 respectively, is to control the abort maneuver such that a suitable injection trajectory is achieved from which a rendezvous profile can be accomplished.

The abort targeting is not fixed, as in the ascent. Instead, the desired down-range speed,  $\dot{Z}_D$ , is computed as a function of the LM-CSM phase angle and the cutoff altitude. If the phase angle of abort initiation is too large ( $\theta > \theta_C$ ), the LM requires an extra revolution to make the rendezvous. In the P70/P71 initialization, this test is made and the appropriate solution to the phasing problem chosen.

For the cutoff conditions,  $\theta$  is approximated by the current value, and the perigee radius ( $R_P$ ) used is an approximate integration of the radial thrust component equation. For an initial value, the nominal cutoff radius ( $R_D$ ) is used.

The initialization operations are shown in Fig. 4.3-1. The iterative guidance computations are shown in Fig. 3.5-3 along the path followed for P7071FLG=1. This flag is set at the beginning of P70 and P71 to cause the guidance equations to operate in the abort target mode.

The cross range translation to be achieved by the abort maneuver is automatically limited by the P70/P71 target initialization to a maximum of 1/2 deg (approximately 8 nm).

The DPS Abort Program, P70, is called by the LGC upon receipt of an abort discrete bit, while P71 is called automatically when an abort stage discrete bit is sensed by the LGC. Either program may be called by the astronaut using a V37 DSKY entry.

When P70 or P71 is called to supersede any of the landing maneuver programs (P63 through P67), their initialization and operation are very similar, both using the Powered Ascent Guidance Equations to control the vehicle to the desired injection conditions. The input parameters required for the P70 and P71 abort programs when used in this mode, which are supplied by the landing programs are:

1.  $t_{IG}$  (Descent) - time of DPS landing maneuver ignition
2.  $t, \underline{r}(t), \underline{v}(t)$  - current LM state vector
3.  $\underline{r}_c, \underline{v}_c$  - any recent CSM state vector
4.  $m$  - vehicle mass (used only by P70)

All other input parameters are prestored in the LGC.

The primary outputs of P70 and P71 are the same as those for the Powered Ascent Guidance Program, P12, of Section 5.3.5, namely: LM Digital Autopilot (DAP) attitude commands, engine-off signals, and display parameters for the Vertical Rise and Ascent Guidance Phases (Section 5.3.5.3).

#### 5.4.3.1.2 DPS Abort Program, P70

The functional logic for P70 is shown in Fig. 4.3-1, together with the Ascent Guidance Computations, Sec. 5.3.5.9. When a DPS powered abort is initiated, the thrust filter must be initialized. Because of the decrease of vehicle mass during the descent burn preceding the abort, the vehicle dynamics must be related to the current mass of the vehicle. In addition, the engine performance itself varies because of erosion which takes place during burning. The main effect of the erosion is to increase the mass flow rate,  $\dot{m}$ , leaving the exhaust velocity,  $V_e$ , relatively unchanged. The initialization of P70 assumes a constant  $\dot{m}$  which introduces an error into the initialization of the engine performance and the thrust magnitude filter. This error will decrease and eventually vanish. The engine performance initialization takes place as follows:

$$a_T = \frac{\dot{m} V_e}{m} = \frac{V_e}{\tau} \quad (4.3.1)$$

where

$$\tau = \frac{m}{\dot{m}}$$

The thrust filter is initialized with the reciprocals of three dummy PIPA readings,  $\Delta V_1$ ,  $\Delta V_2$ ,  $\Delta V_3$ . The same number is used for all three

$$\frac{1}{\Delta V_1} = \frac{1}{\Delta V_2} = \frac{1}{\Delta V_3} = \frac{1}{\Delta V_{IG}}$$

where

$$\Delta V_{IG} = a_T \Delta t,$$

$a_T$  is the initialization acceleration computed above, and  $\Delta t$  is the computation cycle time. The final engine performance parameter to be initialized is

$\Delta t_{\text{Tail-off}}$  - a negative constant time increment used to correct  $t_{go}$  for the P70 DPS tail-off. It is initialized from  $\Delta t_{\text{Tail-off}}$  (P70) in fixed memory.

The next step is to set a mission sequence flag, FLP70, to indicate that P70 has been called, so that any subsequent call to P71 will take this fact into account.

The initialization of the target conditions is done next. First, the time-to-go is estimated by

$$t_{go} = TFI$$

where

$$TFI = t - t_{IG} \text{ (Descent)}$$

The desired cutoff conditions are transferred from fixed to erasable storage, and  $Q$ , the normal to the CSM orbit plane, is computed.

5.4.3.1.3 APS Abort Program, P71

The APS Abort Program, P71, can be used in two modes, either as a primary abort program, or as a follow-up to P70. In either mode, the thrust filter initialization is the same, and differs from the P70 initialization in that all parameters are prestored in the LGC. The initialization of  $t_{go}$  is different in the two modes. In the primary mode it is initialized in the same way as in P70. In the follow-up mode, a value of  $t_{go}$  is available from the operation of P70, and this is doubled to account for the lower acceleration of a full ascent stage compared with an empty descent stage. The initialization of the target is bypassed in the follow-up mode, since it has already been done by P70.

When P71 is called, the flag ABORTFL is set to 0 to prohibit any further calls to the abort programs P70 and P71.

ABTTGFLG is used for downlink information to indicate which set of E-load parameters are being used (when set  $J_2, K_2$ ; when reset  $J_1, K_1$ ). This flag is reset by V37 before entering P70 or P71, and set by the selection of  $J_2, K_2$  within P70 or P71.

5.4-43

Revised LUMINARY  
 Added GSOP # 567 PCR # 893 Rev. 7 Date 11 / 69

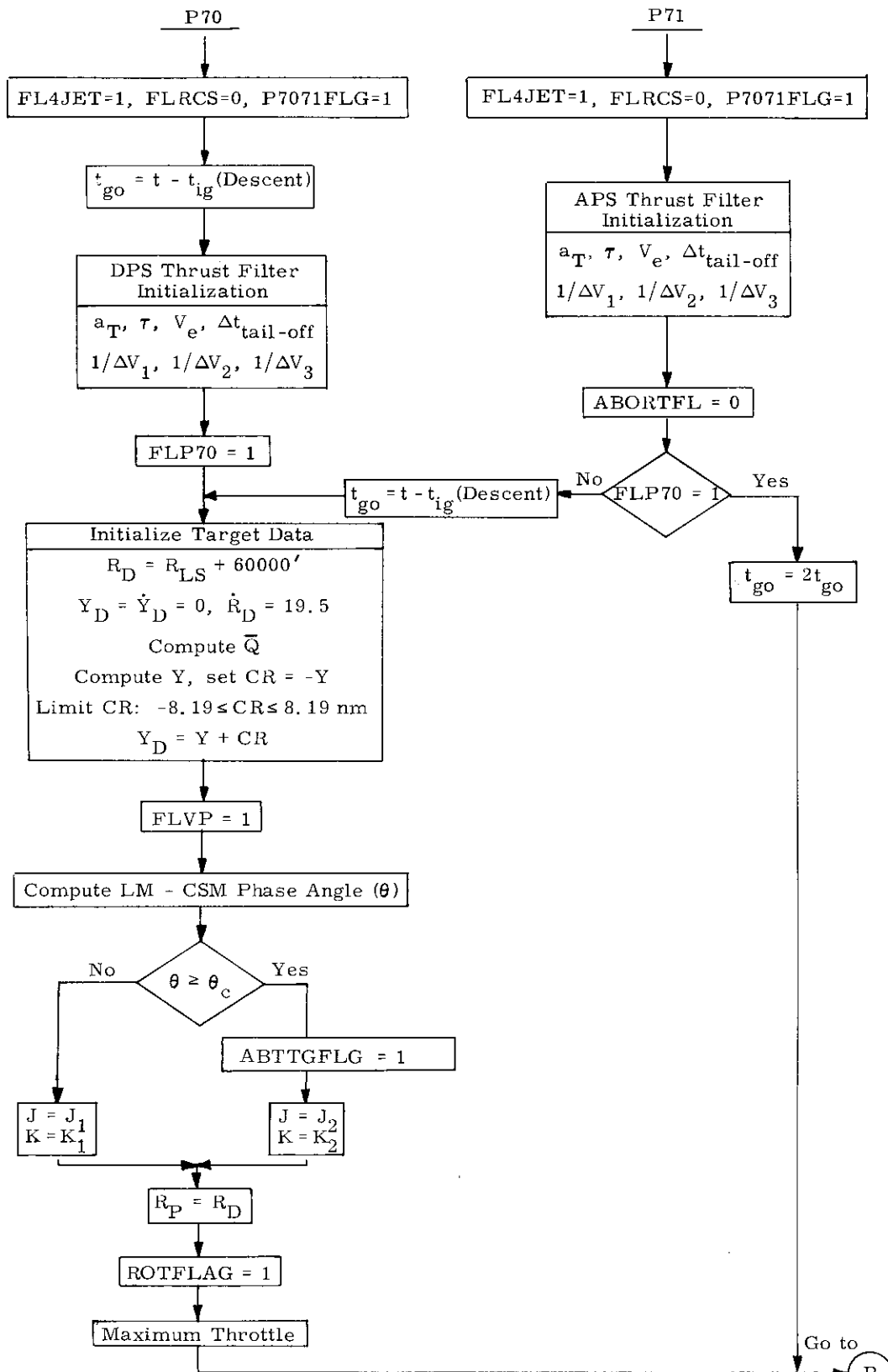


Figure 4.3-1 Abort Targeting and Maneuver Control from Powered Lunar Landing

Go to  
of  
Ascent  
Guidance  
Computations  
Fig. 3.5-3

5.4-44

#### 5.4.3.1.4 Abort Rotation Control

When an abort is initiated, it is necessary to put some constraint on the direction of rotation from initial to desired attitude. The FINDCDUW program would, if left to itself, select the shortest rotation arc and, under some circumstances, this could cause the vehicle x-axis to rotate down through the local vertical, causing problems with certain antennas.

To prevent this, the desired thrust vector,  $\underline{u}_{FDP}$ , is modified before delivery to FINDCDUW. A flag, ROTFLAG, is set to 1 during P70/P71 initialization to invoke this mode of operation, and it can be followed on page 4 of Fig. 3.5-3.

The desired thrust vector is above the local horizontal; hence, a downward-pointing  $\underline{X}_B$  which is within 90 deg of  $\underline{u}_{FDP}$  must be posigrade of the local vertical, and  $\underline{X}_B$  will not rotate through  $-\underline{R}$  during the maneuver. If  $\underline{X}_B$  is above the horizontal, the shortest route is through  $+\underline{R}$ , which is satisfactory. If  $\underline{X}_B$  is not within 90 deg of  $\underline{u}_{FDP}$ , the vehicle is commanded to vertical, which means it will rotate away from  $-\underline{R}$ , no matter where it is. Finally, when  $\underline{X}_B$  is within 30 deg of  $+\underline{R}$ , it is safe to allow the computed  $\underline{u}_{FDP}$  to be commanded, since the shortest route is now through  $+\underline{R}$ .





## 5.5 BASIC SUBROUTINES

### 5.5.1 GENERAL COMMENTS

The basic solar system and conic trajectory sub-routines which are used by the various guidance and navigation routines are described in this section.

#### 5.5.1.1 Solar System Subroutines

The subroutines used to determine the translation and rotation of the relevant solar system bodies (earth, moon and sun) are designed specifically for a fourteen day lunar landing mission. The method of computing the moon and the sun lines - of - sight relative to the earth is given in Section 5.5.4. The transformations between the Basic Reference Coordinate System and the Earth - and Moon-fixed Coordinate Systems are described in Section 5.5.2. The procedure for transforming between vectors in the Basic Reference Coordinate System and latitude, longitude, altitude coordinates is given in Section 5.5.3. Although these subroutines are normally used in the lunar landing mission, they are valid for use in any mission of not more than fourteen days duration in earth-moon space.

### 5.5.1.2 Conic Trajectory Subroutines

This is a description of a group of conic trajectory subroutines which are frequently used by higher level routines and programs in both the Command Module and the Lunar Module computers.

Those subroutines, whose block diagrams are presented in Sections 5.5.5 to 5.5.10, provide solutions to the following conic problems. (See nomenclature which follows)

- (1) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $t_D$ ; solve for  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Kepler Subroutine)
- (2) Given  $\underline{r}(t_1)$ ,  $\underline{r}(t_2)$ ,  $t_{D21}$ ,  $s_G$ ; solve for  $\underline{v}(t_1)$   
(Lambert Subroutine)
- (3) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $\theta$ ; solve for  $t_{21}$ ,  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Time-Theta Subroutine)
- (4) Given  $\underline{r}(t_1)$ ,  $\underline{v}(t_1)$ ,  $r(t_2)$ ,  $s_r$ ; solve for  $t_{21}$ ,  $\underline{r}(t_2)$ ,  $\underline{v}(t_2)$   
(Time-Radius Subroutine)
- (5) Given  $\underline{r}(t)$ ,  $\underline{v}(t)$ ; solve for  $r_P$ ,  $r_A$ ,  $e$   
(Apsides Subroutine)

In addition, the following useful subroutines are provided.

- (6) Conic Parameters Subroutine (See Fig. 5.10-1).
- (7) Geometric Parameters Subroutine (See Fig. 5.10-2).
- (8) Iterator Subroutine (See Fig. 5.10-3).

The solutions to the above set of conic problems have stringent accuracy requirements. Programming the fixed-point Apollo computer introduces two constraints which determine accuracy limitations: the 28 bit double precision word length, and the range of variables which is several orders of magnitude for the Apollo Mission.

In order to maintain numerical accuracy when these subroutines are programmed into the Apollo Computer, floating point programming techniques must be exercised. The effect is for even a simple equation to require a large number of computer instructions. The alternative to this is to separate the problem into phases, each with a different variable range. This, however, requires an even larger number of instructions. These considerations provide the incentive for efficiently organizing the conic equations as shown on the block diagrams.

In addition to the requirement for accuracy the solution to the Kepler and Lambert Problems must be accomplished in a minimum of computation time in order that the guidance system operate satisfactorily in real time. This additional constraint dictates that a minimum of computer instructions be performed when solving the problem.

#### Method of Solution

To minimize the total number of computer instructions, the problems are solved in the "universal" form; i. e. only equations which are equally valid for the ellipse, parabola and hyperbola are used. Also these subroutines can be used with either the earth or the moon as the attracting body.

Kepler's equation, in the universal form, is utilized to relate transfer time to the conic parameters. All other necessary equations are also universal. The Kepler and Lambert problems are solved with a single iteration loop utilizing a simple first-order slope iterator. In the case of the Kepler problem a third order approximation is available to produce the initial guess for the independent variable (See Eq. (2.2.4) of Section 5.2.2.2).

Sections 5.5.5 thru 5.5.10 provide block diagrams of the detailed computational procedures for solving the various problems. The equations are presented in block diagram form with nomenclature below.

#### Range of Variables

As indicated previously, the programming of the conic subroutines requires a careful balance between accuracy, computational speed and number of instructions. This balance, in the Apollo Guidance Computer, leaves very little margin in any of these areas.

Since the values of problem variables are determined by the solution of the problem being solved and since the problem may originate from the ground system, it is essential that the variable range limitations be defined. The conic routines are incapable of handling problems when the solution lies outside of the range.

The following is a list of the maximum allowable numeric values of the variables. Note that, in addition to fundamental quantities such as position and velocity, there are limitations on intermediate variables and combinations of variables.

Scaling for Conic Subroutines (Sections 5.5.5 to 5.5.10)

<u>Parameter</u>	<u>Maximum Value</u> *	
	<u>Earth</u> <u>Primary Body</u>	<u>Moon</u> <u>Primary Body</u>
r	$2^{29}$	$2^{27}$
v	$2^7$	$2^5$
t	$2^{28}$	$2^{28}$
$\alpha^{**}$	$2^{-22}$	$2^{-20}$
$\alpha_N^{**}$	$2^6$	$2^6$
$p_N$	$2^4$	$2^4$
$\cot \gamma$	$2^5$	$2^5$
$\cot \frac{\theta}{2}$	$2^5$	$2^5$
x	$2^{17}$	$2^{16}$
$\xi = \alpha x 2^{***}$	- 50 $+ 4\pi^2$	- 50 $+ 4\pi^2$
$c_1 = \frac{r \cdot v}{\sqrt{\mu}}$	$2^{17}$	$2^{16}$
$c_2 = r v^2 / \mu - 1$	$2^6$	$2^6$
$\lambda = r(t_1) / r(t_2)$	$2^7$	$2^7$
$\cos \theta - \lambda$	$2^7$	$2^7$

\* All dimensional values are in units of meters and centiseconds.

\*\* The maximum absolute value occurs for negative values of this parameter.

\*\*\* Both the maximum and minimum values are listed since neither may be exceeded.

<u>Parameter</u>	<u>Maximum Value*</u>	
	<u>Earth</u>	<u>Moon</u>
$e$	$2^3$	$2^3$
$x^2$	$2^{34}$	$2^{32}$
$x^2 c(\xi)$	$2^{33}$	$2^{31}$
$x^3 s(\xi)/\sqrt{\mu}$	$2^{28}$	$2^{28}$
$c_1 x^2 c(\xi)$	$2^{49}$	$2^{46}$
$c_2 x^2 s(\xi)$	$2^{35}$	$2^{33}$
$x [c_2 x^2 s(\xi) + r(t_1)]$	$2^{49}$	$2^{46}$
$\xi s(\xi)$	$2^7$	$2^7$
$x^2 c(\xi)/r$	$2^8$	$2^8$
$\sqrt{\mu} x (\xi s(\xi) - 1)/r(t_2)$	$2^{15}$	$2^{13}$
$c(\xi)$	$2^4$	$2^4$
$s(\xi)$	$2^1$	$2^1$

\* All dimensional values are in units of meters and centiseconds.

Nomenclature for Conic Subroutines (Sections 5. 5. 5 to 5. 5. 10)

$\underline{r}(t_1)$	initial position vector
$\underline{v}(t_1)$	initial velocity vector
$\underline{r}(t_2)$	terminal position vector
$\underline{v}(t_2)$	terminal velocity vector
$\underline{u}_N$	unit normal in the direction of the angular momentum vector
$\alpha$	reciprocal of semi-major axis (negative for hyperbolas)
$r_P$	radius of pericenter
$r_A$	radius of apocenter
$e$	eccentricity
$\alpha_N$	ratio of magnitude of initial position vector to semi-major axis
$p_N$	ratio of semi-latus rectum to initial position vector magnitude
$\gamma$	inertial flight path angle as measured from vertical
$\theta$	true anomaly difference between $\underline{r}(t_1)$ and $\underline{r}(t_2)$
$f$	true anomaly of $r(t_2)$

$x$	a universal conic parameter equal to the ratio of eccentric anomaly difference to $\sqrt{+\alpha}$ for the ellipse, or the ratio of the hyperbolic analog of eccentric anomaly difference to $\sqrt{-\alpha}$ for the hyperbola
$x'$	value of $x$ from the previous Kepler solution
$t_{21}$	computed transfer time from Kepler's equation ( $t_2 - t_1$ )
$t'_{21}$	transfer time corresponding to the previous solution of Kepler's equation
$t_D$	desired transfer time through which the conic update of the state vector is to be made
$t_{D21}$	desired transfer time to traverse from $\underline{r}(t_1)$ to $\underline{r}(t_2)$
$t_{ERR}$	error in transfer time
$\epsilon_t$	fraction of desired transfer time to which $t_{ERR}$ must converge
$\Delta x$	increment in $x$ which will produce a smaller $t_{ERR}$
$\epsilon_x$	value of $\Delta x$ which will produce no significant change in $t_{21}$
$\Delta \cot \gamma$	increment in $\cot \gamma$ which will decrease the magnitude of $t_{ERR}$
	value of $\Delta \cot \gamma$ which will produce no significant change in $t_{21}$



$\mu$	product of universal gravitational constant and mass of the primary attracting body
$x_{MAX}$	maximum value of x
$x_{MIN}$	minimum value of x
$\cot_{MAX}$	maximum value of $\cot \gamma$
$\cot_{MIN}$	minimum value of $\cot \gamma$
$\ell_{MAX}$	upper bound of general independent variable
$\ell_{MIN}$	lower bound of general independent variable
$x_{MAX 1}$	absolute upper bound on x with respect to the moon
$x_{MAX 0}$	absolute upper bound on x with respect to the earth
k	a fraction of the full range of the independent variable which determines the increment of the independent variable on the first pass through the iterator
y	general dependent variable
y'	previous value of y
y <sub>ERR</sub>	error in y
z	general independent variable

$\Delta z$	increment in z which will produce a smaller $y_{ERR}$
$s_G$	a sign which is plus or minus according to whether the true anomaly difference between $\underline{r}(t_1)$ and $\underline{r}(t_2)$ is to be less than or greater than 180 degrees
$s_r$	a sign which is plus or minus according to whether the desired radial velocity at $\underline{r}(t_2)$ is plus or minus
$\underline{\eta}_1$	general vector # 1
$\underline{\eta}_2$	general vector # 2
$\phi$	angle between $\underline{\eta}_1$ and $\underline{\eta}_2$
$f_1$	a switch set to 0 or 1 according to whether a guess of $\cot \gamma$ is available or not
$f_2$	a switch set to 0 or 1 according to whether Lambert should determine $\underline{u}_N$ from $\underline{r}(t_1)$ and $\underline{r}(t_2)$ or $\underline{u}_N$ is an input
$f_3$	a tag set to 0 or 1 according to whether the iterator should use the "Regula Falsi" or bias method
$f_4$	a flag set to 0 or 1 according to whether the iterator is to act as a first order or a second order iterator
$f_5$	a flag set to 0 or 1 according to whether Lambert converges to a solution or not

- $f_6$  a switch set to 0 or 1 according to whether or not the new state vector is to be an additional output requirement of the Time-Theta or Time-Radius problems.
- $f_7$  a flag set to 1 if the inputs require that the conic trajectory must close through infinity
- $f_8$  a flag set to 1 if the Time-Radius problem was solved for pericenter or apocenter instead of  $r(t_2)$
- $f_9$  a flag set to 1 if the input to the Time-Radius Subroutine produced an  $e$  less than  $2^{-18}$
- $t_p$  period of the orbit
- $k_1$  the minimal acceptance fraction of  $t_{D21}$  to which  $t_{ERR}$  must converge
- $n_1$  a flag set to 0 or 1 according to whether or not the velocity vector at the terminal position is to be an additional output requirement of the Lambert Routine

5.5-11

Revised LUMINARY  
 Added GSOP # 567 PCN # 659, 2\* Rev. 7 Date 11/69

### 5. 5. 2 PLANETARY INERTIAL ORIENTATION SUBROUTINE

This subroutine is used to transform vectors between the Basic Reference Coordinate System and a Planetary (Earth-fixed or Moon-fixed) Coordinate System at a specified time. These three coordinate systems are defined in Section 5. 1. 4.

Let  $\underline{r}$  be a vector in the Basic Reference Coordinate System,  $\underline{r}_P$  the same vector expressed in the Planetary Coordinate System, and  $t$  the specified ground elapsed time (GET). Then,

$$\underline{r}_P = M(t) (\underline{r} - \underline{\ell} \times \underline{r}) \quad (5. 2. 1)$$

and

$$\underline{r} = M^T(t) (\underline{r}_P + \underline{\ell}_P \times \underline{r}_P) \quad (5. 2. 2)$$

where  $M(t)$  is a time dependent orthogonal transformation matrix,  $\underline{\ell}$  is a small rotation vector in the Basic Reference Coordinate System, and  $\underline{\ell}_P$  is the same vector  $\underline{\ell}$  expressed in the Planetary Coordinate System. The vector  $\underline{\ell}$  is considered constant in one coordinate system for the duration of the mission. The method of computing  $M(t)$  and  $\underline{\ell}$  depends on whether the relevant planet is the earth or the moon.

#### Case I - Earth

For the earth, the matrix  $M(t)$  describes a rotation about the polar axis of the earth (the Z-axis of the Earth-fixed Coordinate

System), and the vector  $\underline{\ell}$  accounts for the precession and nutation of the polar axis (the deviation of the true pole from the mean pole).

Let  $A_X$  and  $A_Y$  be the small angles about the X- and Y-axes of the Basic Reference Coordinate System, respectively, that describe the precession and nutation of the earth's polar axis. The values of these two angles at the midpoint of the mission are included in the pre-launch erasable data load and are considered constant throughout the flight. Then,

$$\underline{\ell} = \begin{pmatrix} A_X \\ A_Y \\ 0 \end{pmatrix}$$

$$A_Z = A_{Z0} + \omega_E (t + t_0) \quad (5.2.3)$$

$$M(t) = \begin{pmatrix} \cos A_Z & \sin A_Z & 0 \\ -\sin A_Z & \cos A_Z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\underline{\ell}_P = M(t) \underline{\ell}$$

where  $A_{Z0}$  is the angle between the X-axis of the Basic Reference Coordinate System and the X-axis of the Earth-fixed Coordinate System (the intersection of the Greenwich meridian and the equatorial plane of the earth) at July 1.0, 1969 universal time (i.e., midnight at Greenwich just prior to July 1, 1969),  $t_0$  is the elapsed time between July 1.0, 1969 universal time and the time that the computer clock was zeroed, and  $\omega_E$  is the angular velocity of the earth.

## Case II - Moon

For the moon, the matrix  $M(t)$  accounts for the difference in orientation of the Basic Reference and Moon-fixed Coordinate Systems in exact accordance with Cassini's laws, and the rotation vector  $\underline{\ell}$  corrects for deviations from the above orientation because of physical libration.

Define the following three angles which are functions of time:

- B = the obliquity, the angle between the mean earth equatorial plane and the plane of the ecliptic.
- $\Omega_1$  = the longitude of the node of the moon's orbit measured from the X-axis of the Basic Reference Coordinate System.
- F = the angle from the mean ascending node of the moon's orbit to the mean moon.

Let  $I$  be the constant angle between the mean lunar equatorial plane and the plane of the ecliptic ( $5521.5''$ ). Then, the sequence of rotations which brings the Basic Reference Coordinate System into coincidence with the Moon-fixed Coordinate System (neglecting libration) is as follows:

<u>Rotation</u>	<u>Axis of Rotation</u>	<u>Angle of Rotation</u>
1	X	B
2	Z	$\Omega_I$
3	X	-I
4	Z	$\pi + F$

The transformation matrices for these rotations are, respectively,

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos B & \sin B \\ 0 & -\sin B & \cos B \end{pmatrix}$$

$$M_2 = \begin{pmatrix} \cos \Omega_I & \sin \Omega_I & 0 \\ -\sin \Omega_I & \cos \Omega_I & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos I & -\sin I \\ 0 & \sin I & \cos I \end{pmatrix}$$

$$M_4 = \begin{pmatrix} -\cos F & -\sin F & 0 \\ \sin F & -\cos F & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(5.2.4)

The matrix  $M(t)$  is then given by

$$M(t) = M_4 M_3 M_2 M_1 \quad (5.2.5)$$

The following approximate method is used to determine the transformation between the Basic Reference and Moon-fixed Coordinate Systems.

The angles  $B$ ,  $\Omega_I$  and  $F$  are computed as linear functions of time. Let  $\underline{\ell}_M$  be the value of the vector libration  $\underline{\ell}_P$  (expressed in the Moon-fixed Coordinate System) at the nominal lunar landing time. While the LM is on the surface of the moon, a new value of  $\underline{\ell}_M$  referenced to the nominal launch time should be uplinked. The vector  $\underline{\ell}_M$  is included in the pre-launch erasable data load and is considered constant throughout the flight. Then,

$$\begin{aligned} \underline{\ell}_P &= \underline{\ell}_M \\ t_M &= t + t_0 \\ B &= B_0 + \dot{B} t_M \\ \Omega_I &= \Omega_{I0} + \dot{\Omega}_I t_M \\ F &= F_0 + \dot{F} t_M \end{aligned}$$

$$\underline{a} = \begin{pmatrix} \cos \Omega_I \\ \cos B \sin \Omega_I \\ \sin B \sin \Omega_I \end{pmatrix} \quad (5.2.6)$$

$$\underline{b} = \begin{pmatrix} -\sin \Omega_I \\ \cos B \cos \Omega_I \\ \sin B \cos \Omega_I \end{pmatrix}$$

5.5-16



$$\underline{c} = \begin{pmatrix} 0 \\ -\sin B \\ \cos B \end{pmatrix}$$

$$\underline{d} = \underline{b} C_I - \underline{c} S_I$$

$$\underline{m}_2 = \underline{b} S_I + \underline{c} C_I$$

$$\underline{m}_0 = -\underline{a} \cos F - \underline{d} \sin F \quad (5.2.6)$$

(cont.)

$$\underline{m}_1 = \underline{a} \sin F - \underline{d} \cos F$$

$$M(t) = \begin{pmatrix} \underline{m}_0^T \\ \underline{m}_1^T \\ \underline{m}_2^T \end{pmatrix}$$

$$\underline{\ell} = M^T(t) \underline{\ell}_P$$

where  $B_0$ ,  $\Omega_{I0}$ , and  $F_0$  are the values of the angles  $B$ ,  $\Omega_I$  and  $F$ , respectively, at July 1.0, 1969 universal time;  $\dot{B}$ ,  $\dot{\Omega}_I$  and  $\dot{F}$  are the rates of change of these angles; and  $C_I$  and  $S_I$  are the cosine and sine, respectively, of the angle  $I$ .

### 5. 5. 3 LATITUDE-LONGITUDE SUBROUTINE

For display and data load purposes, the latitude, longitude, and altitude of a point near the surface of the earth or the moon are more meaningful and more convenient to use than the components of a position vector. This subroutine is used to transform position vectors between the Basic Reference Coordinate System and Geographic or Selenographic latitude, longitude, altitude at a specified time.

In the case of the moon, the altitude is computed above either the landing site radius,  $r_{LS}$ , or the mean lunar radius,  $r_M$ . For the earth, the altitude is defined with respect to either the launch pad radius,  $r_{LP}$  or the radius of the Fisher ellipsoid,  $r_F$ , which is computed from

$$r_F^2 = \frac{b^2}{1 - \left(1 - \frac{b^2}{a^2}\right) (1 - \text{SINL}^2)} \quad (5.3.1)$$

where  $a$  and  $b$  are the semi-major and semi-minor axes of the Fischer ellipsoid, respectively, and  $\text{SINL}$  is the sine of the geocentric latitude.

The computational procedures are illustrated in Figs. 5.3-1, 5.3-2, and 5.3-3. The calling program must specify either a vector  $\underline{r}$  or latitude (Lat), longitude (Long), and altitude (Alt). In addition, the program must set the time  $t$  and the two indicators  $P$  and  $F$  where

$$P = \begin{cases} 0 & \text{for earth} \\ 1 & \text{for moon} \end{cases}$$

$$F = \begin{cases} 1 & \text{for Fischer ellipsoid or mean lunar radius} \\ 0 & \text{for launch pad or landing site radius} \end{cases}$$

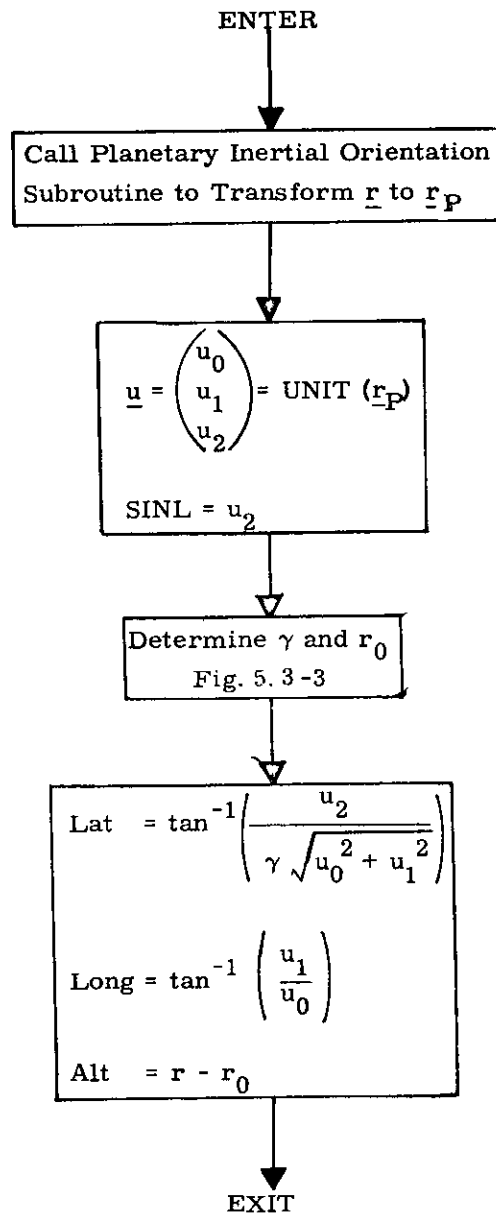


Fig. 5.3-1 Vector to Latitude, Longitude, Altitude Computation Logic Diagram

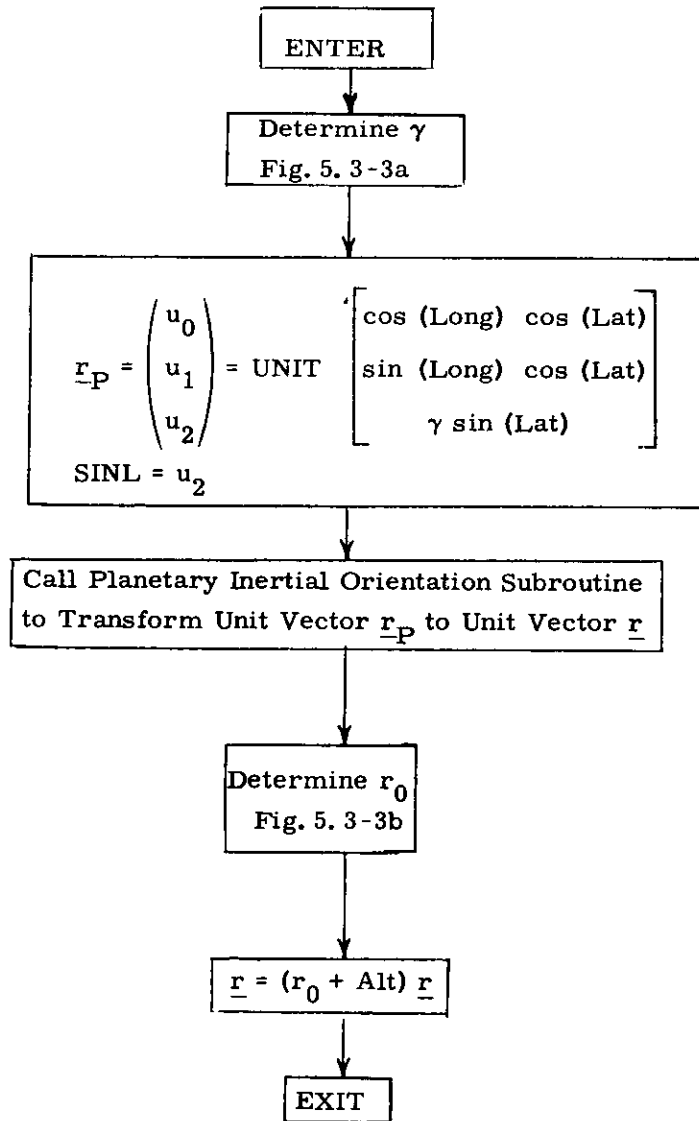


Fig. 5.3-2 Latitude, Longitude, Altitude to Vector Computation Logic Diagram

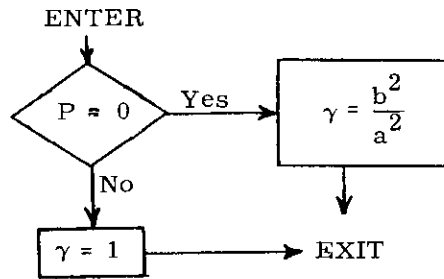


Figure 5.3-3a Determination of  $\gamma$

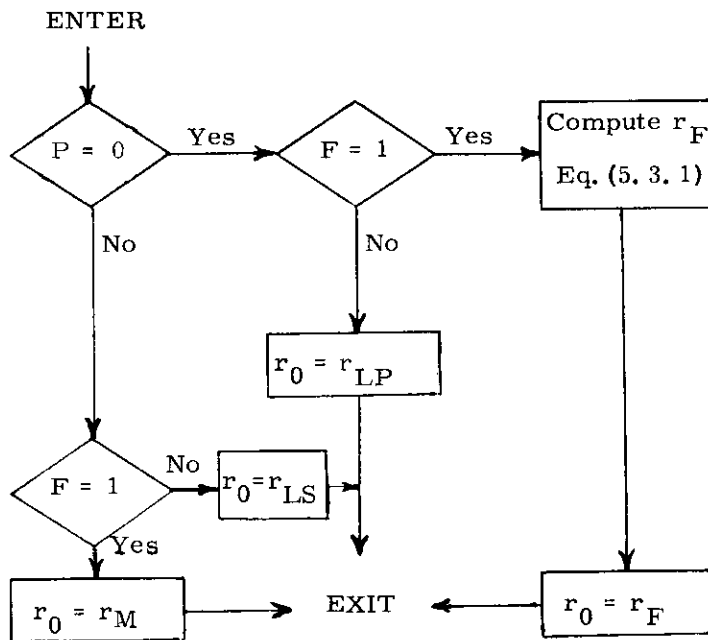


Figure 5.3-3b Determination of  $r_0$

#### 5.5.4 LUNAR AND SOLAR EPHEMERIDES

This subroutine is called by LOSSEM (Section 5.5.13) in order to obtain the line-of-sight unit vectors of the moon and sun, with respect to the LM, which may be used for IMU alignment.

The positions of the moon and sun are obtained from simple approximation schemes which are stored in the computer's fixed memory.

If the obliquity of the ecliptic and the inclination of the lunar orbit are considered constant, the unit position vector of the moon with respect to the earth in the Basic Reference Coordinate System defined in Section (5.1.4), is given by

$$\underline{u}_{EM} = \begin{bmatrix} \cos (LOM) \\ K_1 \sin (LOM) - K_2 \sin (LOM - LON) \\ K_3 \sin (LOM) + K_4 \sin (LOM - LON) \end{bmatrix}$$

where the longitude of the mean moon (LOM) is approximated using its value and rate at the appropriate July 1.0, 1969, and using empirically determined periodic terms:

$$LOM = LOM_O + LOM_R t - \left[ A \sin (OMEGA_A t + PHASE_A) \right. \\ \left. + B \sin (OMEGA_B t + PHASE_B) \right]$$

and where the longitude of the moon's node (LON) is similarly approximated:

$$LON = LON_O + LON_R t$$

and where  $t$  is measured from July 1, 0, 1969.

The approximate unit position vector of the sun with respect to the earth is given by:

$$\underline{u}_{ES} = \begin{bmatrix} \cos (LOS) \\ K_1 \sin (LOS) \\ K_3 \sin (LOS) \end{bmatrix}$$

where the longitude of the mean sun is approximated by:

$$LOS = LOS_O + LOS_R t - C \sin (OMEGA_C t + PHASE_C)$$

and where  $K_1$  and  $K_3$  are the same as for the moon.



### 5. 5. 5 KEPLER SUBROUTINE

The Kepler Subroutine solves for the two body position and velocity vectors at the terminal position given the initial position and velocity vectors and a transfer time to the terminal position.

This section contains information to aid the reader in understanding the less obvious aspects of the Kepler Subroutine block diagram depicted in Figs. 5. 5-1 thru 5. 5-3. The subroutines referred to in these figures are presented in Section 5. 5. 10 and the nomenclature is found in Section 5. 5. 1. 2.

Prior to entering the Kepler Subroutine, an initial estimate of  $x$  can be generated via Eq. (2. 2. 4) of Section 5. 2. 2. 2 with  $\frac{\Delta t}{2} = t_D - t_{21}$  and  $r = t_D$ . However,  $x'$  and  $t_{21}$  are non-zero only if the subroutine is being used repetitively.

Although, theoretically, there is no upper bound on  $x$ , the practical bound is set to  $x_{MAX0}$  or  $x_{MAX1}$  to eliminate non-feasible trajectories and increase the accuracy to which  $x$  can be computed. In addition,  $\alpha x^2$  has a practical range of  $-50 < \alpha x^2 < (2\pi)^2$  which determines an independent upper bound on  $x$ . The  $x_{MAX}$  used, then, corresponds to the smaller of the two values.

The transfer time convergence criterion is approximately the same as the granularity of the time input. Since, for some of the problems to be solved, the sensitivity of time to  $x$  is so large that the granularity in  $x$ ,  $\epsilon_x$ , produces a change in time which exceeds the granularity in time, it is necessary to introduce  $\epsilon_x$  as a redundant convergence criterion.

The Kepler Subroutine, provided the parameter range constraints are satisfied, will always produce a solution.

A negative value of  $t_D$  will cause the subroutine to update the state vector backward in time (i. e. backdate the state vector). The subroutine may be called to update or backdate for any amount of time; there are no restrictions on whether the time  $t_D$  is less than a period.

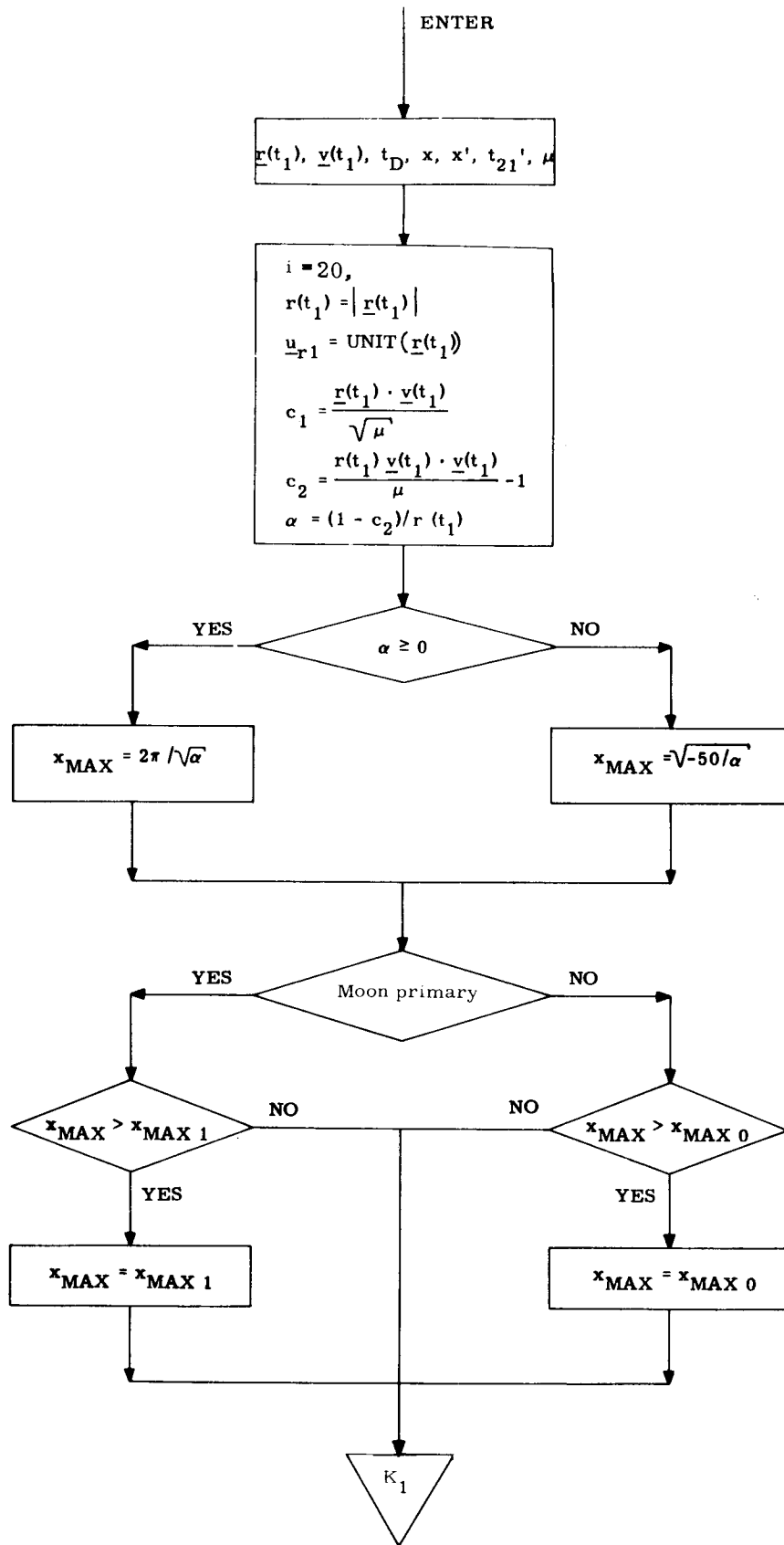


Figure 5.5-1 Kepler Subroutine

.....

.....

.....

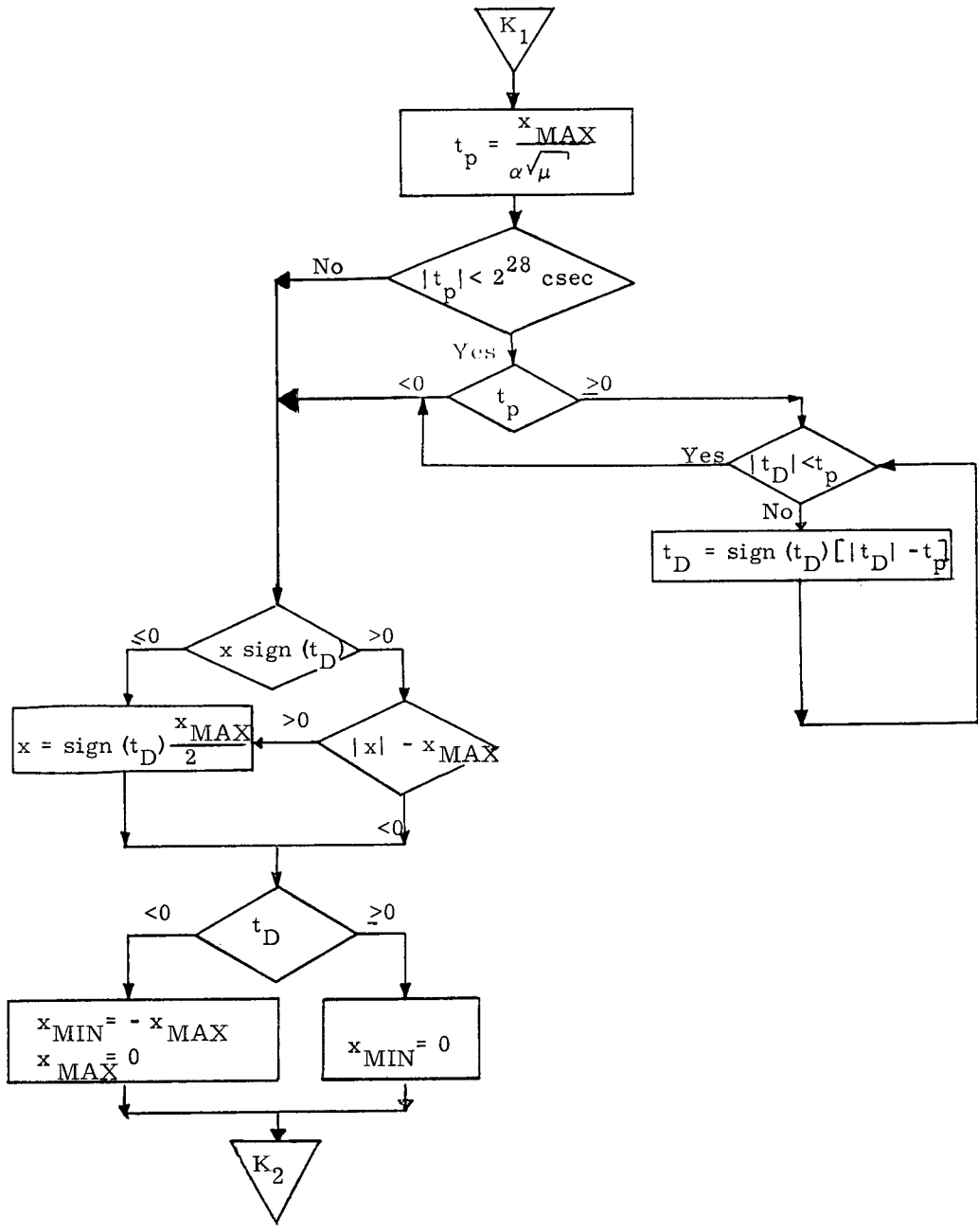


Fig. 5.5-2 Kepler Subroutine

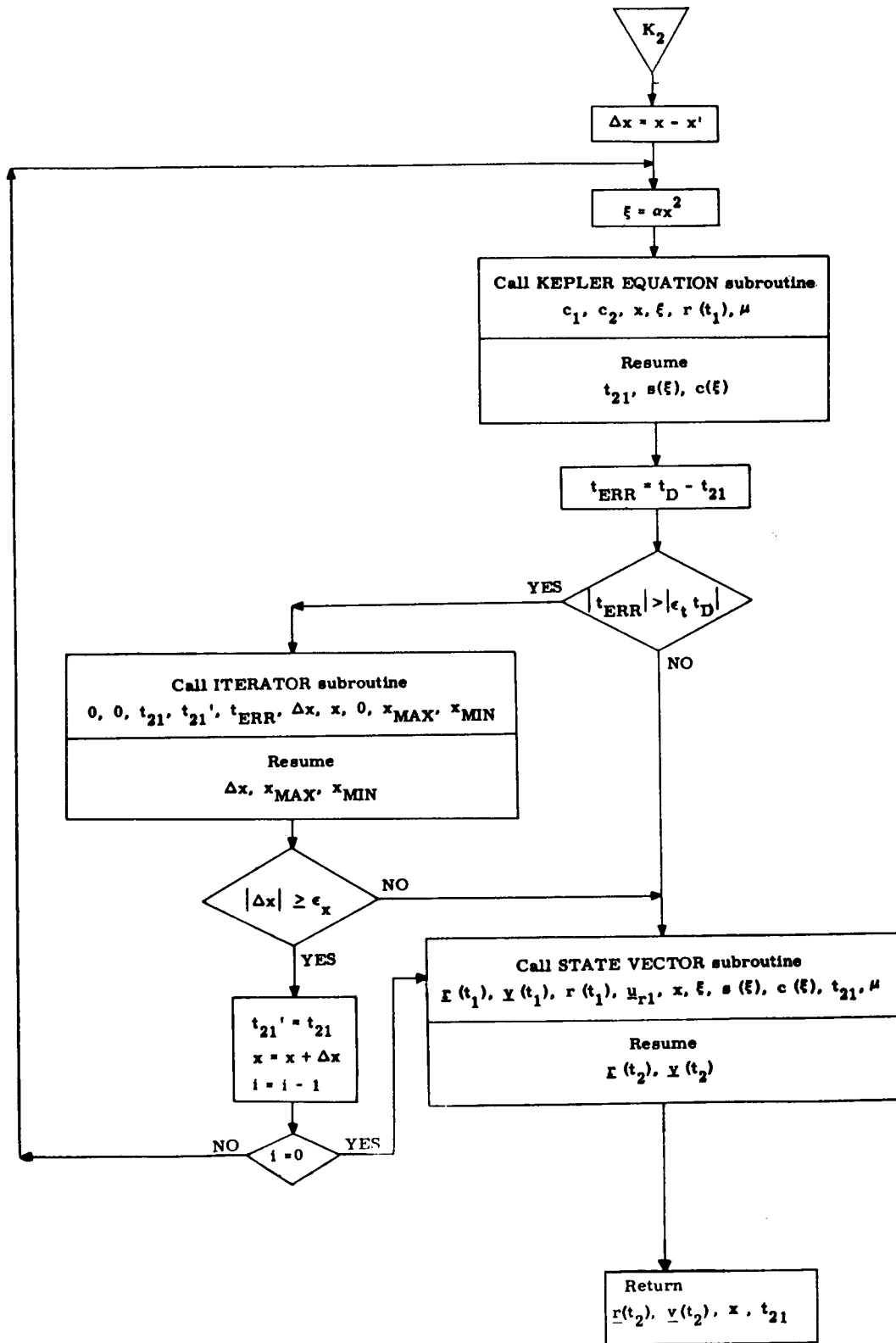


Fig. 5.5-3 Kepler Subroutine

### 5.5.6 LAMBERT SUBROUTINE

The Lambert Subroutine solves for the two body initial velocity vector given the initial and terminal position vectors and a transfer time between the two.

This section contains information to aid the reader in understanding the less obvious aspects of the Lambert Subroutine block diagrams depicted in Figs. 5.6-1 and 5.6-2. The subroutines referred to in these figures are presented in Section 5.5.10. Nomenclature is found in Section 5.5.1.2.

If the Lambert Subroutine is used repetitively and rapid computation is required, the previous value of the independent variable,  $\cot \gamma$ , can be used as a starting point for the new iteration. Flag  $f_1$  provides this option.

The Lambert Subroutine computes the normal to the trajectory,  $\underline{u}_N$ , using the two input position vectors. If these vectors are nearly colinear, it is desirable to specify the normal as an input rather than rely on the ill-defined normal based on the two input position vectors. Flag  $f_2$  provides this option. The presence of the inputs in parentheses, therefore, is contingent upon the setting of these flags.

The theoretical bounds on the independent variable,  $\cot \gamma$ , correspond to the infinite energy hyperbolic path and the parabolic path which closes through infinity. These bounds are dynamically reset by the iterator to provide a more efficient iteration scheme. In addition, if during the course of the iteration,  $\cot \gamma$  causes a parameter of the problem to exceed its maximum as determined by its allowable range, the appropriate bound is reset and the iterator continues trying to find an acceptable solution. (This logic does not appear in Figs. 5.6-1 and 2

as it is pertinent only to fixed-point programming). If no acceptable solution is reached, the transfer time input was too small to produce a practical trajectory between the input position vectors. When this happens,  $\Delta \cot \gamma$  approaches its granularity limit  $\epsilon_c$  before time converges to within a fraction  $\epsilon_t$  of the desired time. However, this same granularity condition exists when the sensitivity problem occurs as described in the Kepler Subroutine, Section 5.5.5. In this case an acceptable solution does exist. This dual situation is resolved via a third convergence criterion. If the error in transfer time is greater than the usual fraction  $\epsilon_t$  of the desired transfer time, but still less than a slightly larger fraction  $k_1$  of the desired transfer time and  $\Delta \cot \gamma$  is less than  $\epsilon_c$ , then the solution is deemed acceptable and the required velocity is computed. If, on the other hand, the error in the transfer time is greater than the fraction  $k_1$  of the desired transfer time and  $\Delta \cot \gamma$  is less than  $\epsilon_c$ , then the flag  $f_5$  is set to indicate that the required velocity will have degraded accuracy.

The maximum number of iterations allowable in the Lambert Subroutine has been made an input to the routine, in order to give the user direct control over the maximum computation time to be allowed. If no initial guess of  $\cot \gamma_0$  is supplied, this number should be set to 20. If an initial guess of  $\cot \gamma_0$  is supplied, which is sufficiently close to the true  $\cot \gamma_0$ , then this number may be set as low as 5.

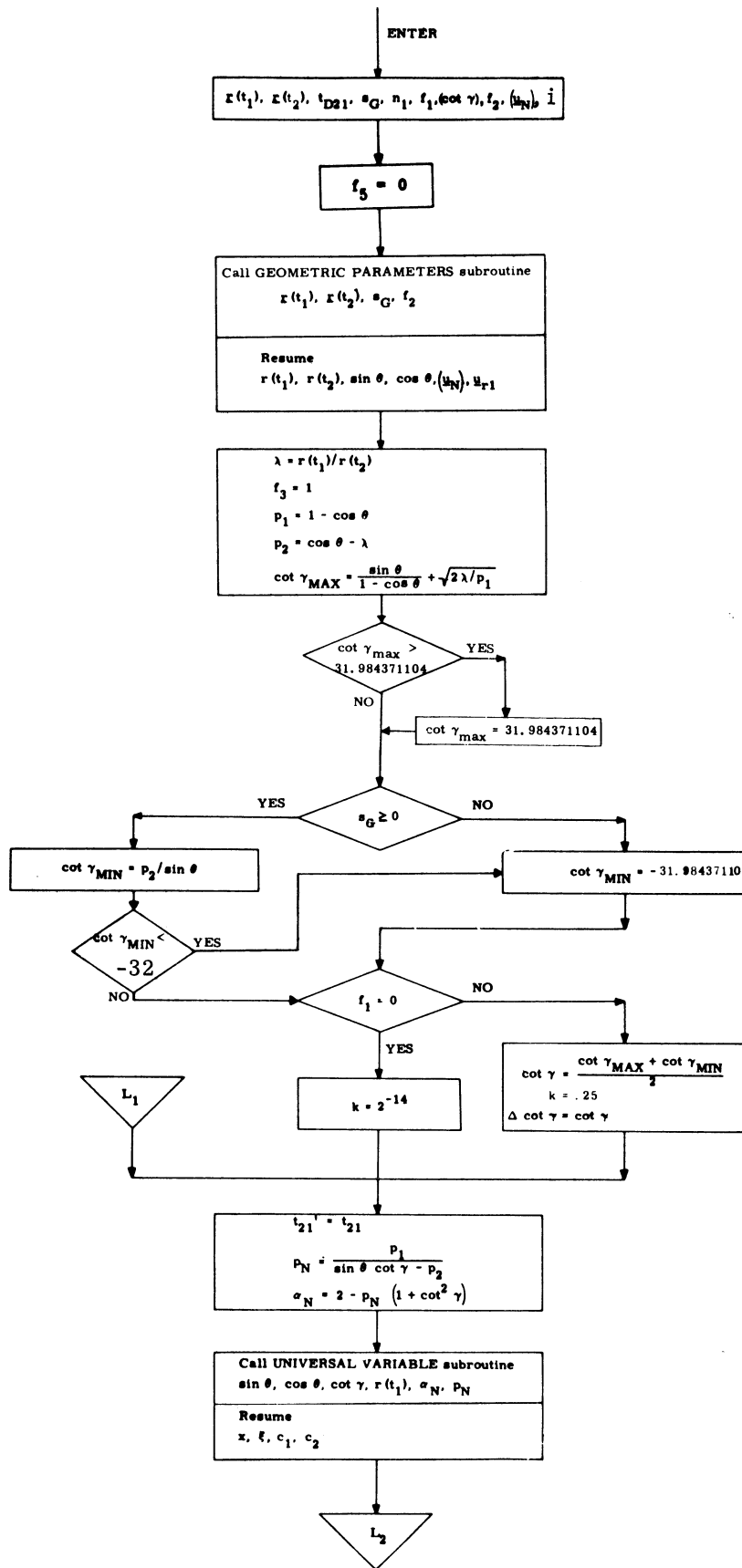


Figure 5. 6-1 Lambert Subroutine



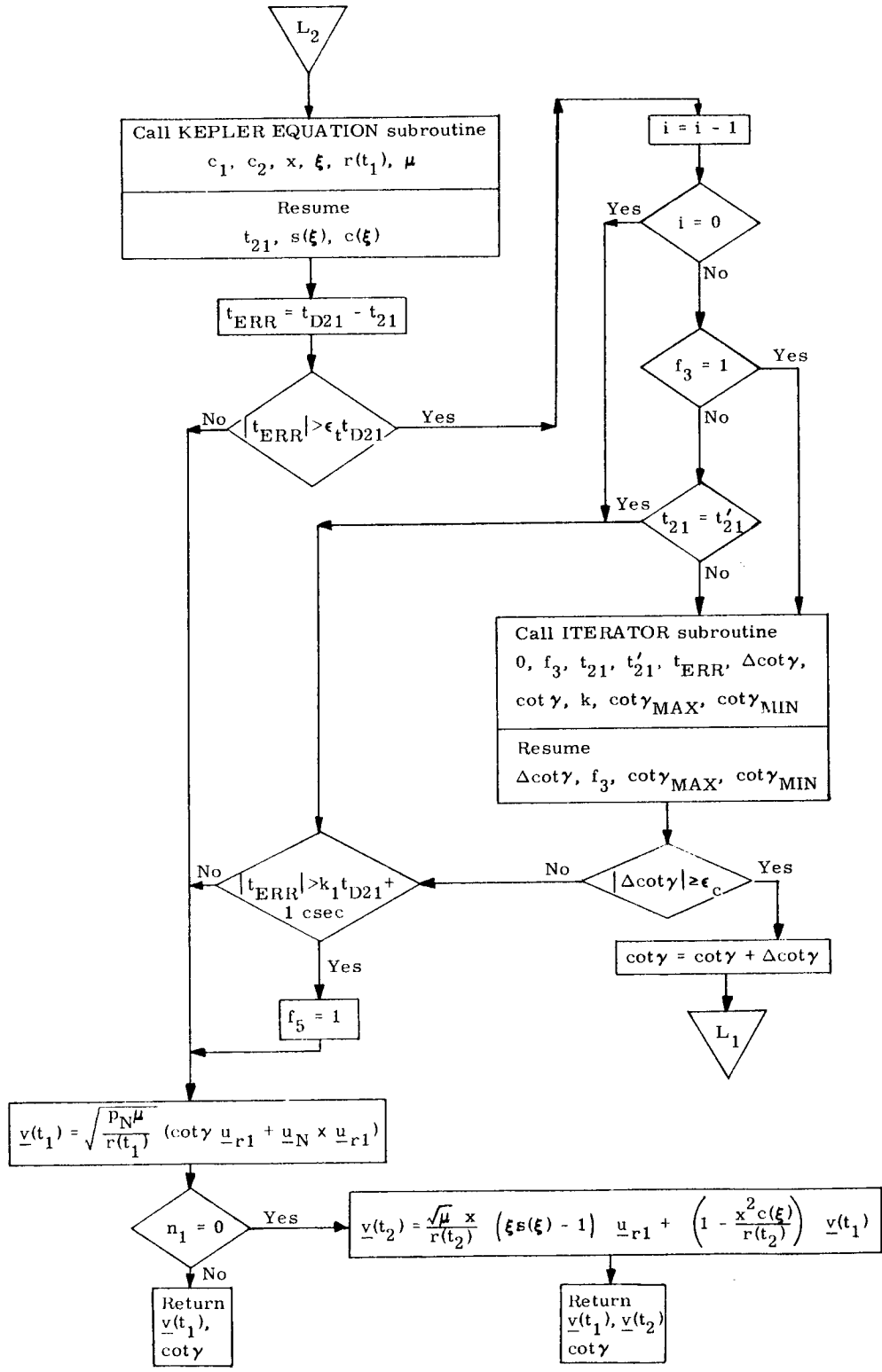


Figure 5.6-2 Lambert Subroutine

### 5.5.7 TIME-THETA SUBROUTINE

The Time-Theta Subroutine solves for the two body transfer time given the initial position and velocity vectors and the true anomaly difference (transfer angle) to the terminal position.

This section contains information to aid the reader in understanding the less obvious aspects of the Time-Theta Subroutine block diagram depicted in Fig. 5.7-1. The subroutines referred to in this figure are presented in Section 5.5.10. Nomenclature is found in Section 5.5.1.2.

The flag  $f_6$  must be zero if the user desires computation of the terminal state vector in addition to the transfer time.

If the conic trajectory is a parabola or hyperbola and the desired transfer angle,  $\theta$ , lies beyond the asymptote of the conic,  $f_7$  will be set indicating that no solution is possible.\*

In addition to the parameter range constraints imposed on Kepler's equation, the additional restriction on Time-Theta that the trajectory must not be near rectilinear is indicated by the range of  $\cot \gamma$ .\*

The Time-Theta problem is not well defined for near rectilinear trajectories, i. e. the transfer angle  $\theta$  is no longer a meaningful problem parameter. This will not cause difficulties provided the input variables are within the specified ranges.

---

\*If the Time-Theta Routine is called with inputs for which no solution is possible (for either or both of these two reasons), the routine will abort with an alarm code of 20607.

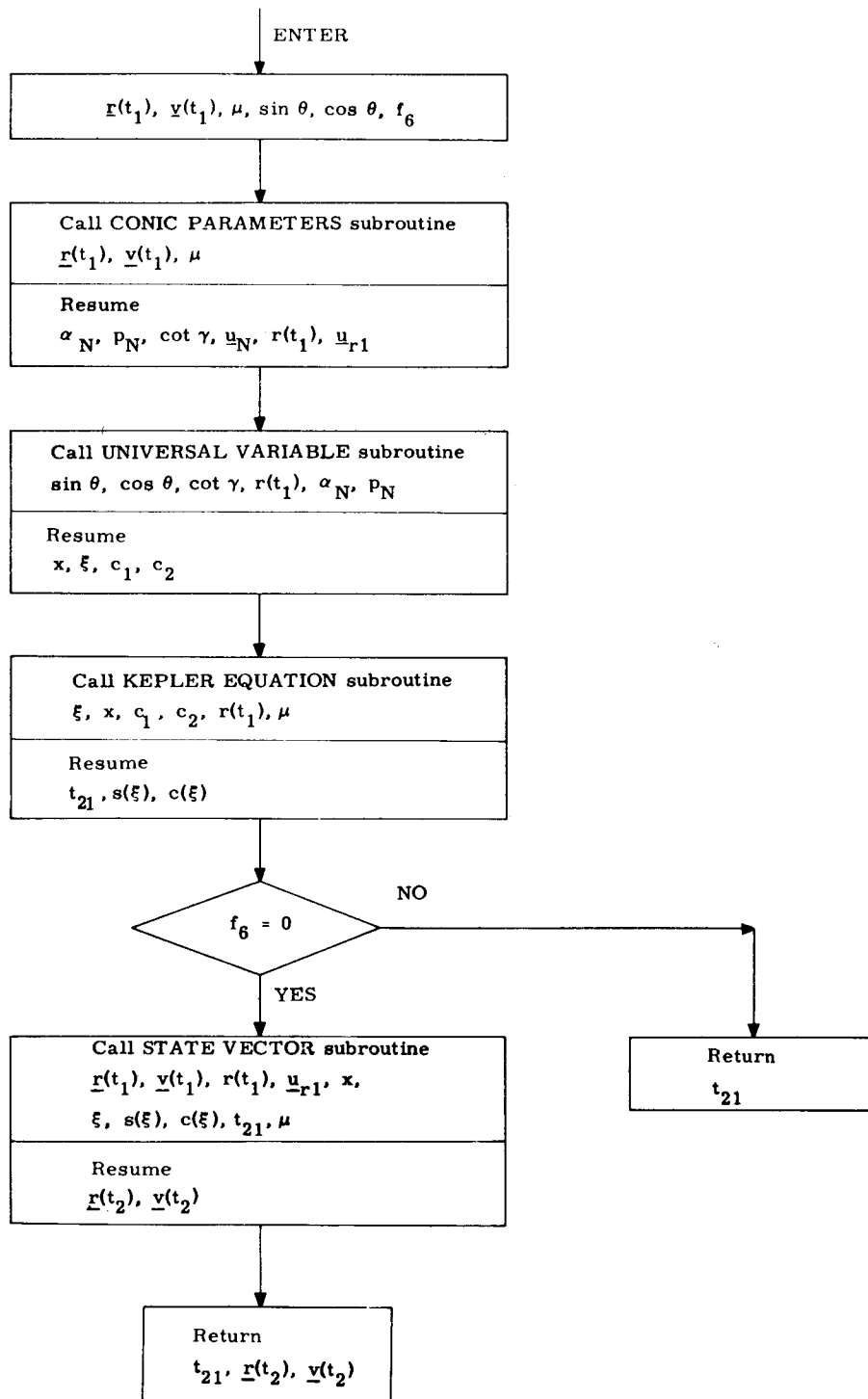


Figure 5.7-1 Time-Theta Subroutine

### 5.5.8 TIME-RADIUS SUBROUTINE

The Time Radius Subroutine solves for the two body transfer time to a specified radius given the initial position and velocity vectors and the radius magnitude.

This section contains information to aid the reader in understanding the less obvious aspects of the Time-Radius Subroutine block diagrams depicted in Figs. 5.8-1 and 5.8-2. The subroutines referred to in this figure are presented in Section 5.5.10. Nomenclature is found in Section 5.5.1.2

Paragraphs 3, 4, and 5 of Section 5.5.7 apply to the Time-Radius Subroutine as well.\*

Since an inherent singularity is present for the circular orbit case, near-circular orbits result in a loss of accuracy in computing both the transfer time,  $t_{21}$ , and the final state vector. This is caused by the increasing sensitivity of  $t_{21}$  to  $r(t_2)$  as the circular orbit is approached. In the extreme case that eccentricity is less than  $2^{-18}$ , the problem is undefined and the subroutine will exit without a solution, setting flag  $f_9$  to indicate this.\*

If  $r(t_2)$  is less than the radius of pericenter or greater than the radius of apocenter, then  $r(t_2)$  will be ignored and the pericenter or apocenter solution, respectively, will be computed. A flag,  $f_8$ , will be set to indicate this.

---

\*If the Time-Radius Routine is called with inputs for which no solution is possible (for any one or more of the reasons given in Paragraphs 4 and 5 of Section 5.5.7 and Paragraph 4 above), the routine will abort with an alarm code of 20607<sub>8</sub>.



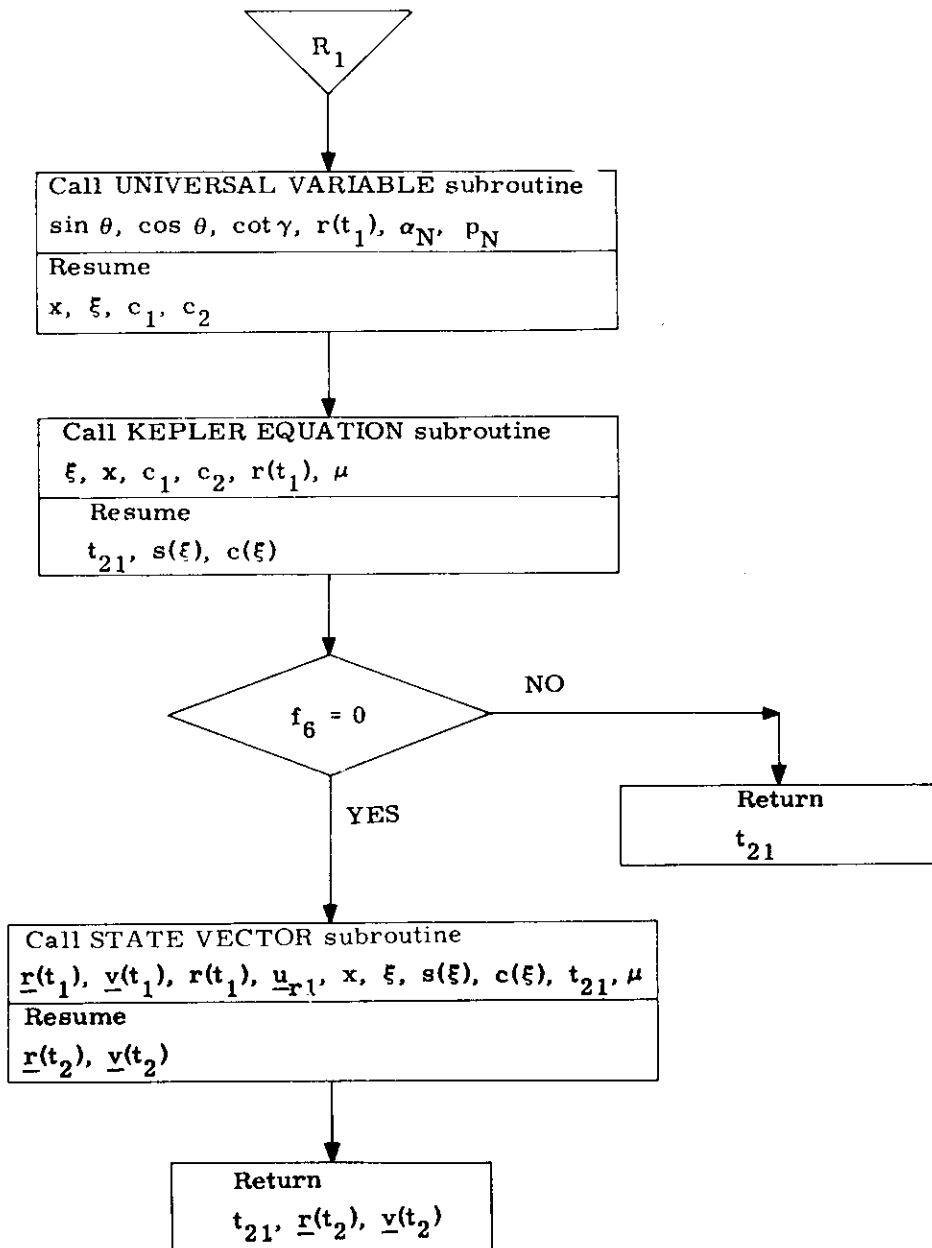


Figure 5.8-2 Time-Radius Subroutine

### 5.4.3 ABORT TARGETING

#### 5.4.3.1 Aborts from Powered Landing

##### 5.4.3.1.1 General Objectives

The objective of the DPS and APS Abort Targeting and Control Programs, P70 and P71 respectively, is to control the abort maneuver such that a suitable injection trajectory is achieved from which a rendezvous profile can be accomplished.

The abort targeting is not fixed, as in the ascent. Instead, the desired down-range speed,  $\dot{Z}_D$ , is computed as a function of the LM-CSM phase angle and the cutoff altitude. If the phase angle of abort initiation is too large ( $\theta > \theta_c$ ), the LM requires an extra revolution to make the rendezvous. In the P70/P71 initialization, this test is made and the appropriate solution to the phasing problem chosen.

For the cutoff conditions,  $\theta$  is approximated by the current value, and the perigee radius ( $R_p$ ) used is an approximate integration of the radial thrust component equation. For an initial value, the nominal cutoff radius ( $R_D$ ) is used.

The initialization operations are shown in Fig. 4.3-1. The iterative guidance computations are shown in Fig. 3.5-3 along the path followed for P7071FLG=1. This flag is set at the beginning of P70 and P71 to cause the guidance equations to operate in the abort target mode.

The cross range translation to be achieved by the abort maneuver is automatically limited by the P70/P71 target initialization to a maximum of 1/2 deg (approximately 8 nm).

The DPS Abort Program, P70, is called by the LGC upon receipt of an abort discrete bit, while P71 is called automatically when an abort stage discrete bit is sensed by the LGC. Either program may be called by the astronaut using a V37 DSKY entry.

When P70 or P71 is called to supersede any of the landing maneuver programs (P63 through P66), their initialization and operation are very similar, both using the Powered Ascent Guidance Equations to control the vehicle to the desired injection conditions. The input parameters required for the P70 and P71 abort programs when used in this mode, which are supplied by the landing programs are:

1.  $t_{IG}(\text{Descent})$  - time of DPS landing maneuver ignition
2.  $t, \underline{r}(t), \underline{v}(t)$  - current LM state vector
3.  $\underline{r}_c, \underline{v}_c$  - any recent CSM state vector
4.  $m$  - vehicle mass (used only by P70)

All other input parameters are prestored in the LGC.

The primary outputs of P70 and P71 are the same as those for the Powered Ascent Guidance Program, P12, of Section 5.3.5, namely: LM Digital Autopilot (DAP) attitude commands, engine-off signals, and display parameters for the Vertical Rise and Ascent Guidance Phases (Section 5.3.5.3).

#### 5.4.3.1.2 DPS Abort Program, P70

The functional logic for P70 is shown in Fig. 4.3-1, together with the Ascent Guidance Computations, Sec. 5.3.5.9. When a DPS powered abort is initiated, the thrust filter must be initialized. Because of the decrease of vehicle mass during the descent burn preceding the abort, the vehicle dynamics must be related to the current mass of the vehicle. In addition, the engine performance itself varies because of erosion which takes place during burning. The main effect of the erosion is to increase the mass flow rate,  $\dot{m}$ , leaving the exhaust velocity,  $V_e$ , relatively unchanged. The initialization of P70 assumes a constant  $\dot{m}$  which introduces an error into the initialization of the engine performance and the thrust magnitude filter. This error will decrease and eventually vanish. The engine performance initialization takes place as follows:



#### 5. 5. 10      MISCELLANEOUS SUBROUTINES

There are, as part of the Conic Trajectory Sub-routines, three subroutines which are useful in their own right. These are the Conic Parameters, the Geometric Parameters and the Iterator Subroutines which are depicted in Figs. 5. 10-1, 5. 10-2 and 5. 10-3, respectively.

The Conic Parameters and Geometric Parameters Subroutines are self explanatory.

The Iterator Subroutine serves several purposes. It is used when flag  $f_4$  is set to zero to solve for the value of the independent variable which drives the error in the dependent variable to zero, provided the function is monotonically increasing. To improve convergence for functions whose derivative changes rapidly, the limits are reset as shown in the block diagram.

With  $f_4$  set to 1, the Iterator seeks a minimum of the function, provided the first derivative is single-valued between the limits. The inputs are redefined so that "y" is the derivative of the independent variable with respect to the dependent variable, and "x" is the value at which the derivative was computed or approximated. Since the desired value of y is zero,  $y_{ERR} = -y$ .

Since the Iterator uses the "Regula Falsi" technique, it requires two sets of variables to begin iteration. If only one set is available, flag  $f_3$  must be set to 1, causing the iterator to generate the independent variable increment from a percentage of the full range.

In addition to the above subroutines there are three other subroutines of primary interest to the five basic conic subroutines described in Sections 5.5.5 to 5.5.9. These are the Universal Variable Subroutine, the Kepler Equation Subroutine, and the State Vector Subroutine shown in Figs. 5.10-4, 5.10-5 and 5.10-6, respectively.

The Universal Variable Subroutine is utilized by the Lambert, the Time-Theta and the Time-Radius Subroutines to compute the universal parameter  $x$  required for the time equation. There are two different formulations required according to the size of the parameter  $w$ .

If the input to the subroutine requires the physically impossible solution that the trajectory "close through infinity", the problem will be aborted, setting flag  $f_7$ .

The Kepler Equation Subroutine computes the transfer time given the variable  $x$  and the conic parameters.

The State Vector subroutine computes the position and velocity vectors at a point along the trajectory given an initial state vector, the variable  $x$  and the transfer time.

The final miscellaneous subroutine, the SETMU Subroutine, is depicted in Fig. 5.10-7. It sets  $\mu$  to the appropriate primary body gravitational constant consistent with the estimated CSM or LM state vector as defined in Section 5.2.2.6.

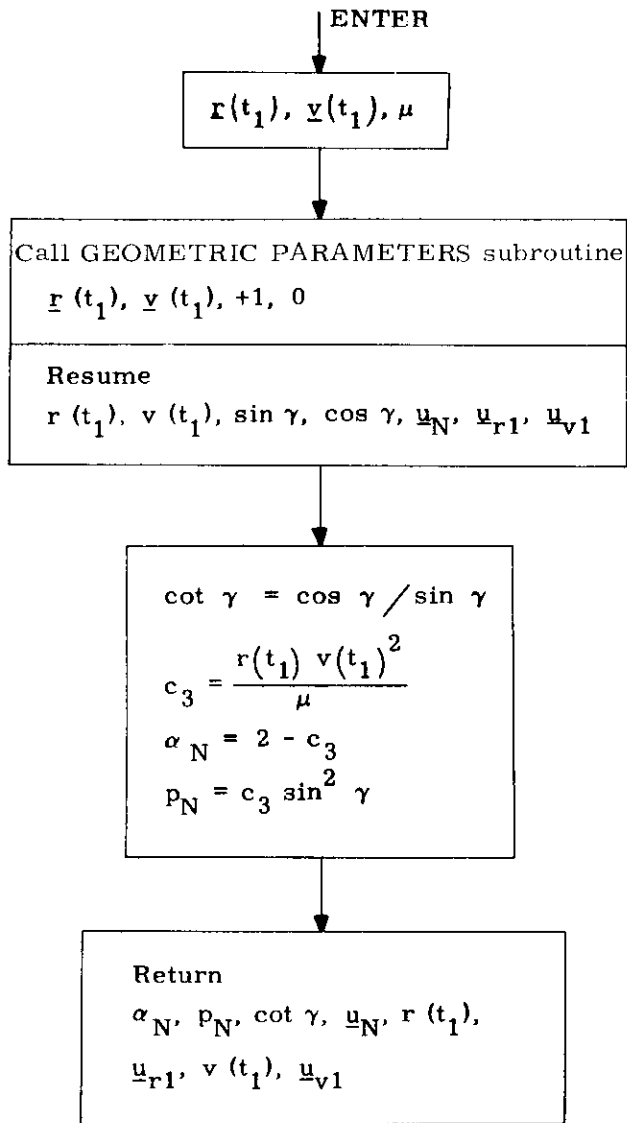


Figure 5.10-1 Conic Parameters Subroutine

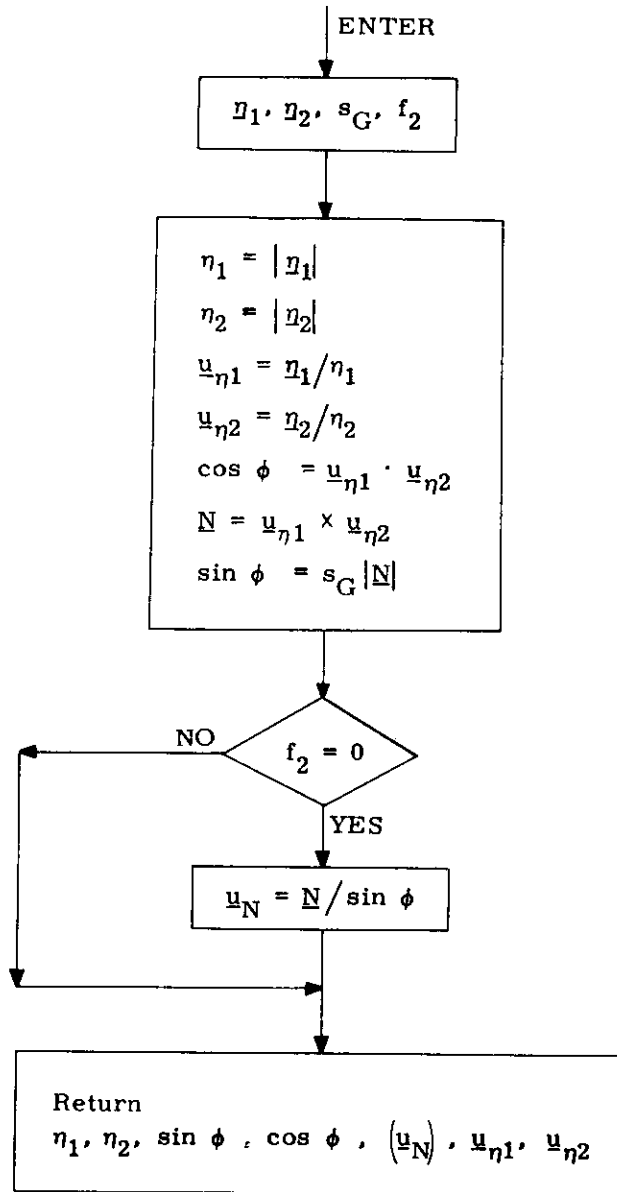


Figure 5.10-2 Geometric Parameters Subroutine

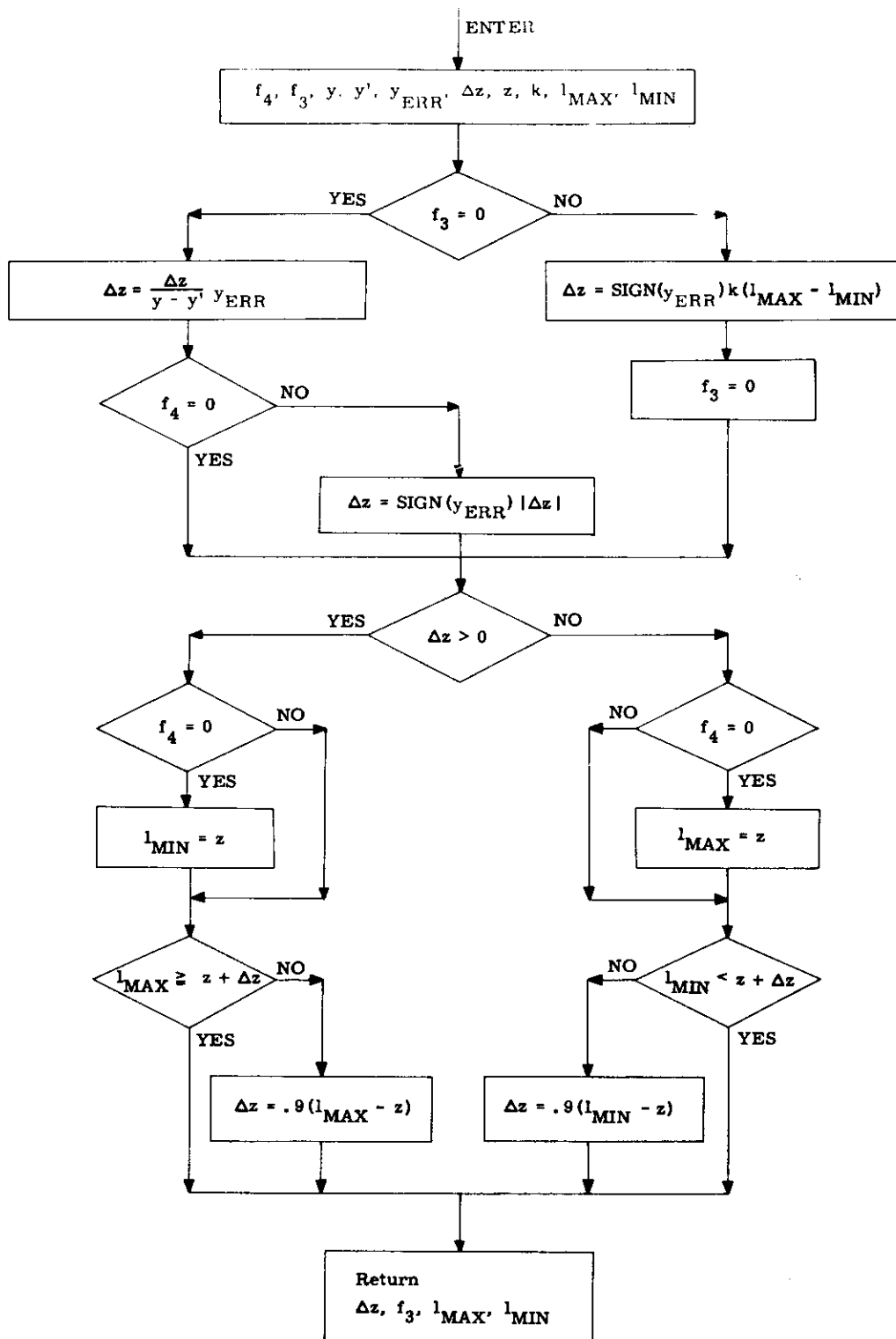


Figure 5.10-3 Iterator Subroutine

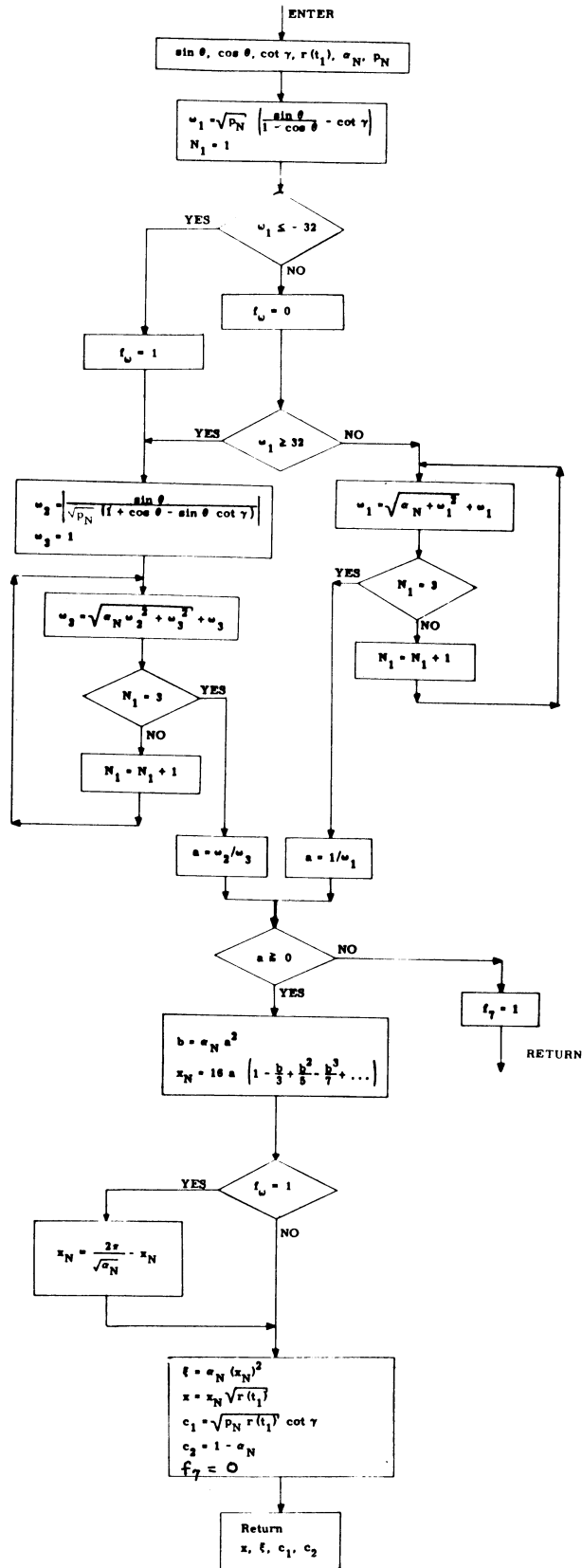


Figure 5.10-4 Universal Variable Subroutine

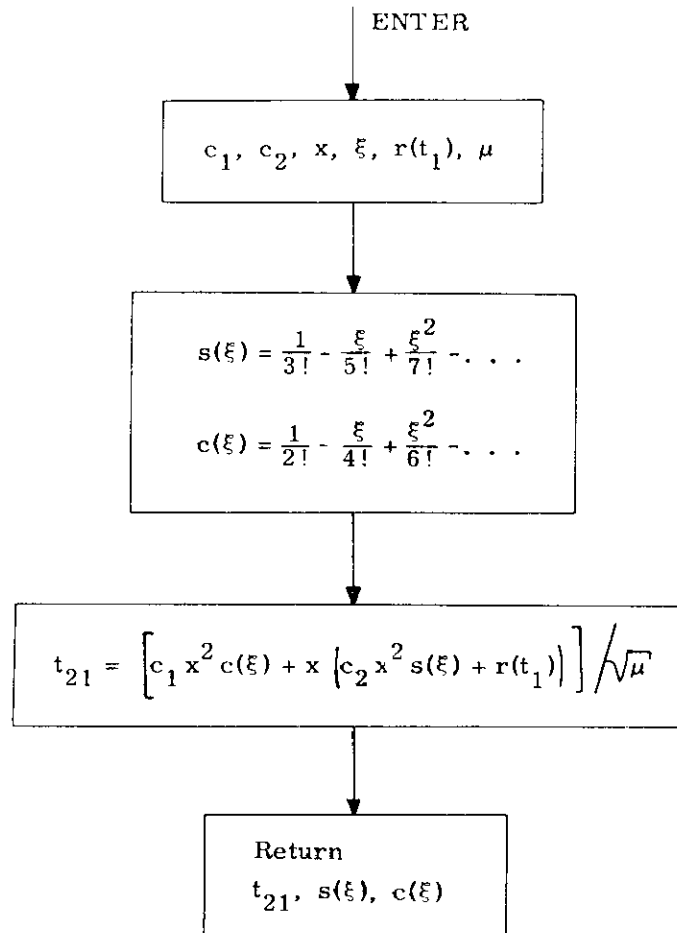


Figure 5.10-5 Kepler Equation Subroutine

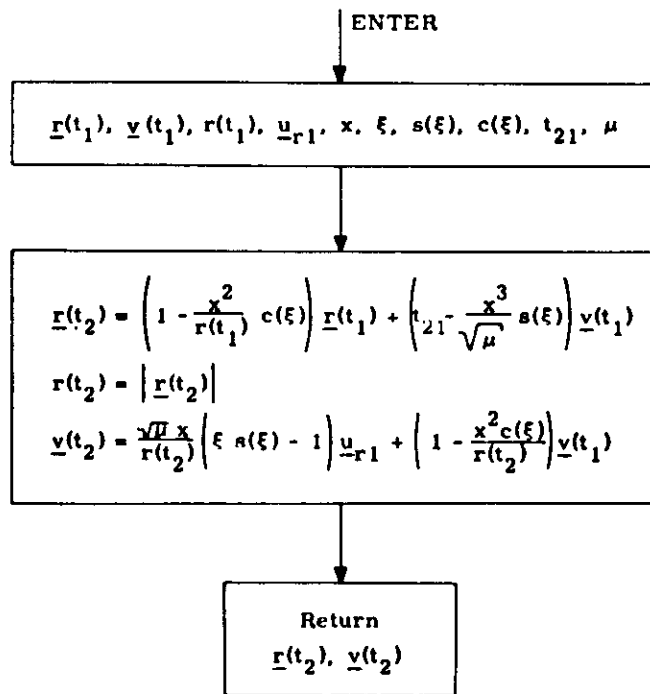


Figure 5.10-6 State Vector Subroutine



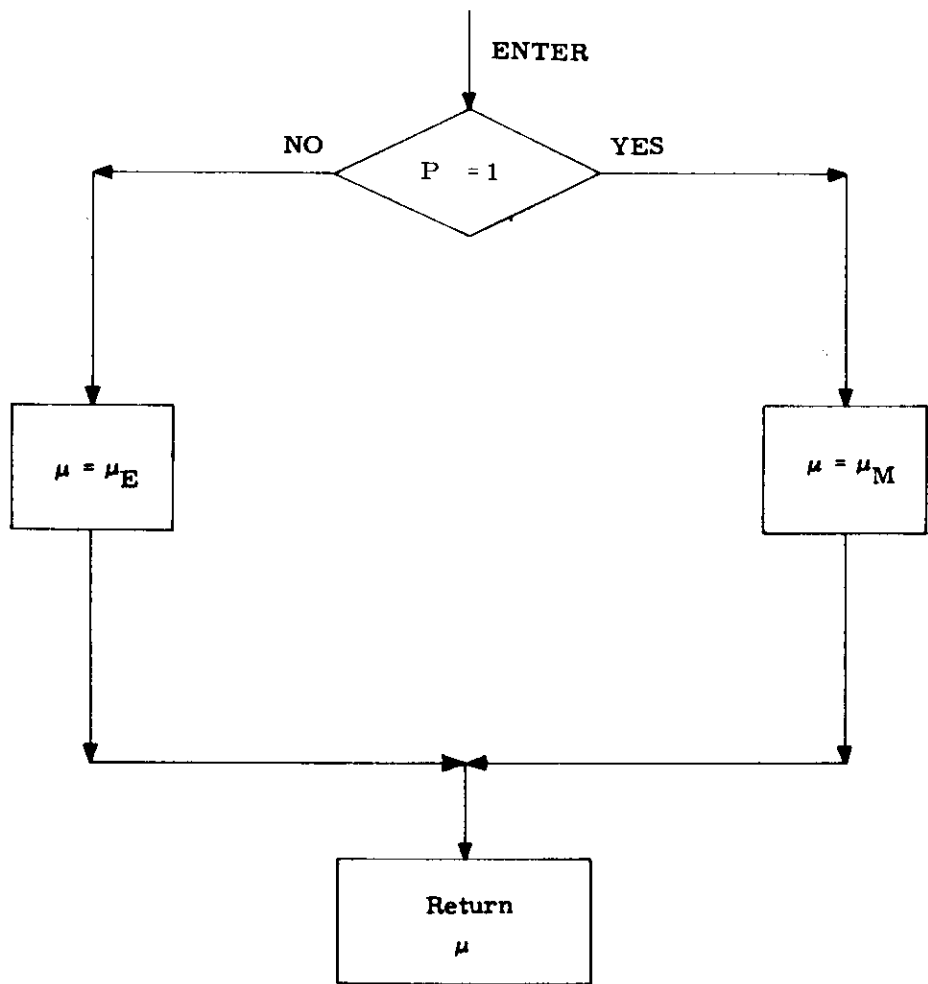


Figure 5.10-7 SETMU Subroutine

#### 5.5.11 INITIAL VELOCITY SUBROUTINE

The Initial Velocity Subroutine computes the required initial velocity vector for a trajectory of specified transfer time between specified initial and target position vectors. The trajectory may be either conic or precision depending on an input parameter (namely, number of offsets). In addition, in the precision trajectory case, the subroutine also computes an "offset target vector", to be used during pure-conic cross-product steering. The offset target vector is the terminal position vector of a conic trajectory which has the same initial state as a precision trajectory whose terminal position vector is the specified target vector.

In order to avoid the inherent singularities in the  $180^\circ$  transfer case when the (true or offset) target vector may be slightly out of the orbital plane, the Initial Velocity Subroutine rotates this vector into a plane defined by the input initial position vector and another input vector (usually the initial velocity vector), whenever the input target vector lies inside a cone whose vertex is the origin of coordinates, whose axis is the  $180^\circ$  transfer direction, and whose cone angle is specified by the user.

The Initial Velocity Subroutine is depicted in Fig. 5.11-1. The Lambert Subroutine, Section 5.5.6, is utilized for the conic computations; and the Coasting Integration Subroutine, Section 5.2.2, is utilized for the precision trajectory computations.

The Initial Velocity Subroutine sets the maximum number of iterations allowable inside the Lambert Subroutine to 20 in all cases.

## Nomenclature for the Initial Velocity Subroutine

$\underline{r}(t_1)$	Initial position vector.
$\underline{v}(t_1)$	Vector (usually the actual initial velocity vector) used to determine whether the transfer from the initial position vector to the target vector is through a central angle of less or greater than $180^\circ$ , and also used in certain cases to specify the transfer plane (see text).
$\underline{r}_T(t_2)$	Target Vector (True target vector if $N_1 > 0$ , or Offset target vector if $N_1 = 0$ ).
$t_D$	Desired transfer time from initial position vector to target vector.
$N_1$	Number of offsets to be used in calculating the offset target vector from the true target vector. ( $N_1 = 0$ implies conic calculations only with offset target vector input).
$\epsilon$	Cone Angle of a cone whose vertex is the coordinate origin and whose axis is the $180^\circ$ transfer direction (i. e., the negative initial position direction). The cone angle $\epsilon$ is measured from the axis to the side of the cone.
$f_1$	Switch set to 0 or 1 according to whether a guess of $\cot \gamma$ is input or not.
{ $\cot \gamma$ }	Guess of $\cot \gamma$ .
$\underline{v}_T(t_1)$	Required initial velocity vector of a precision [a conic] trajectory which passes through the true [or offset] target vector, or the rotated true [or offset] target vector if the original target vector was in the cone, at the end of the desired transfer time, if $N_1 > 0$ [or $N_1 = 0$ ].
$\underline{r}(t_2)$	Computed offset target vector.
$\underline{v}_T(t_2)$	Final precision [conic] velocity vector resulting from a precision [conic] update of the initial position vector and the required initial velocity vector $\underline{v}_T(t_1)$ , if $N_1 > 0$ , [or $N_1 = 0$ , respectively]
$\underline{r}_T(t_2)$	Final precision position vector.
$\cot \gamma$	Value to which the Lambert Subroutine converged (for later use as guess to minimize computation time).
$f_2$	Switch set to 0 or 1 according to whether the input (true or offset) target vector was not or was in the cone, and consequently was not or was rotated into the plane.

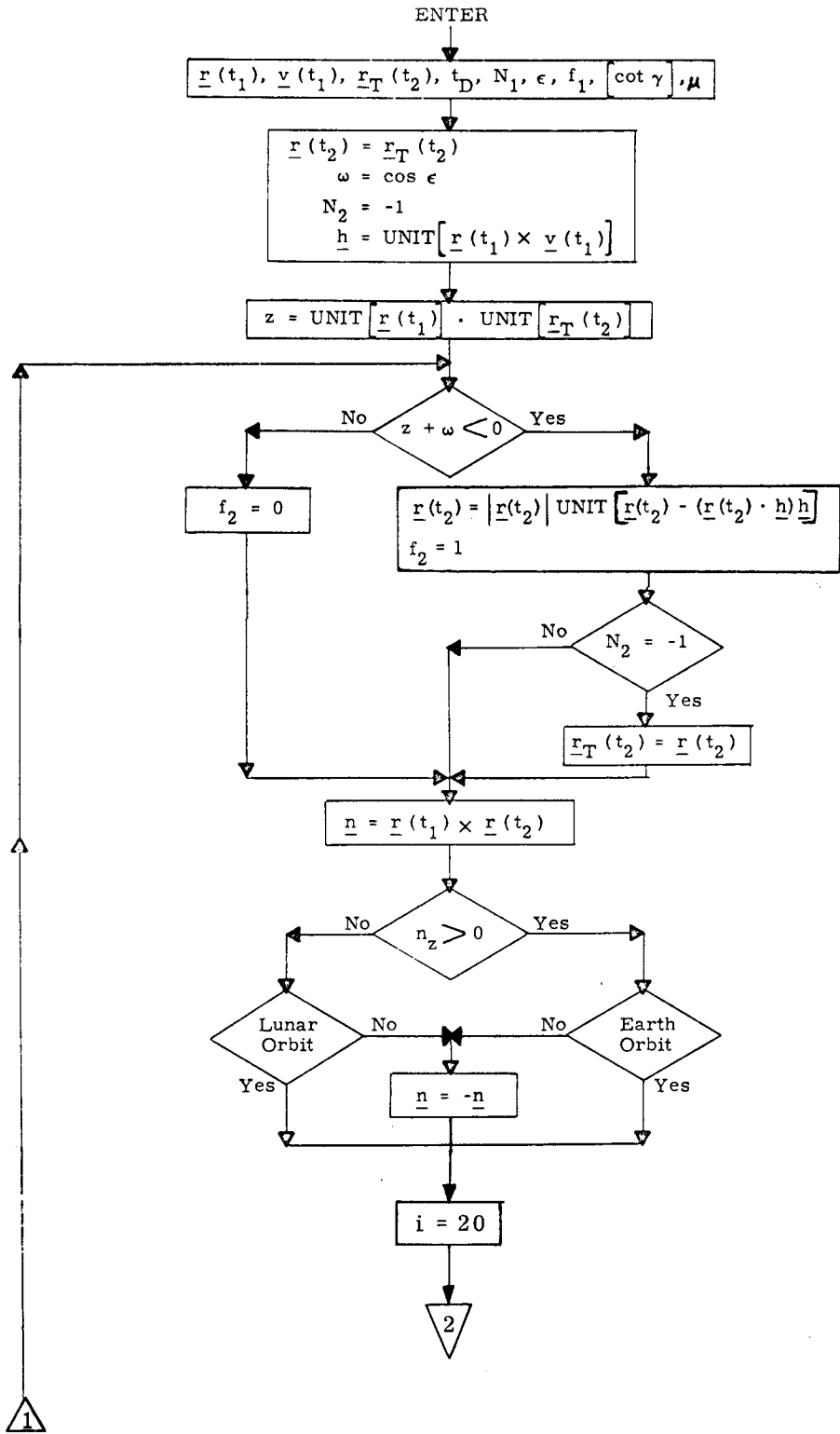


Figure 5.11-1 Initial Velocity Subroutine  
(page 1 of 2)

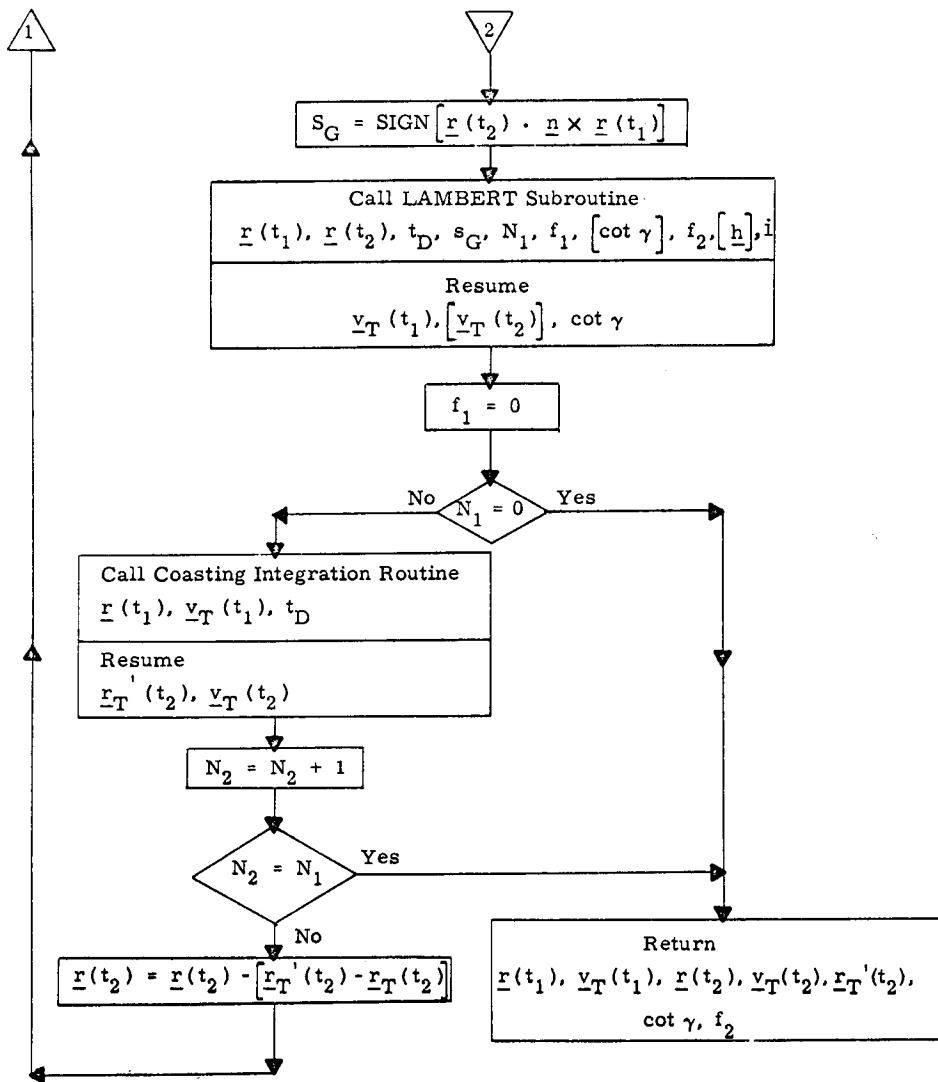


Figure 5.11-1 Initial Velocity Subroutine  
(page 2 of 2)

### 5.5.13 LOSSEM SUBROUTINE

The LOSSEM Subroutine computes the lines-of-sight of the Sun, Earth, and Moon with respect to the spacecraft in the Basic Reference Coordinate System. This data is used by the IMU alignment programs whenever the astronaut elects to sight on the Sun, Earth, or Moon instead of a star for purposes of IMU alignment. The data is also used by the Star Selection Routine (Section 5.6.4) when testing for star occultation. In addition, this subroutine computes the sizes of the occultation cones used in the Star Selection Routine.

The unit vectors  $\underline{u}_S$ ,  $\underline{u}_E$ , and  $\underline{u}_M$  specifying the lines-of-sight to the Sun, Earth, and Moon, respectively, in the Basic Reference Coordinate System are computed as follows:

$$\underline{u}_S = \begin{cases} \underline{u}_{ES} & \text{if } P = E \\ \text{UNIT} (\underline{u}_{ES} - \rho \underline{u}_{EM}) & \text{if } P = M \end{cases} \quad (5.13.1)$$

$$\underline{u}_E = \begin{cases} -\text{UNIT} (\underline{r}_L) & \text{if } P = E \\ -\text{UNIT} (r_{EM} \underline{u}_{EM} + \underline{r}_L) & \text{if } P = M \end{cases} \quad (5.13.2)$$

$$\underline{u}_M = \begin{cases} \text{UNIT} (r_{EM} \underline{u}_{EM} - \underline{r}_L) & \text{if } P = E \\ -\text{UNIT} (\underline{r}_L) & \text{if } P = M \end{cases} \quad (5.13.3)$$

where P, E, M, S, and L respectively denote the primary body, Earth, Moon, Sun, and Lunar Module,  $\underline{r}_L$  is the position vector of the LM with respect to the primary body,  $\underline{u}_{EM}$  and  $\underline{u}_{ES}$  are the unit position vectors of the Moon and Sun with respect to the Earth obtained from the Lunar and Solar Ephemerides Subroutine of Section 5.5.4,  $r_{EM}$  is the mean distance (384402 km) between the Earth and Moon, and  $\rho$  is the ratio (0.00257125) of  $r_{EM}$  over the mean distance between the Sun and Earth. The line-of-sight vectors are determined for a time specified by the calling program or routine. When LOSSEM is used by the sighting mark routines it is not called until just after the astronaut finishes his optical marks on the Sun, Earth, or Moon, and the time used in this case is that recorded for the last mark. When LOSSEM is used by the Auto Optics Positioning

Routine (R-52) it is called just before positioning the spacecraft and the specified time is the present time. When LOSSEM is called by the IMU Realignment Program (P-52) just prior to calling the Star Selection Routine, the specified time is the present time plus an additional amount ( $T_{SS}$ ) in order to project it into the middle of the sighting mark process. This is done to insure that the line-of-sight vector of the primary body used in the occultation test of the Star Selection Routine is that which occurs during the sighting mark process.

The occultation cones used in the Star Selection Routine for the Sun, Earth, and Moon are computed as follows:

$$c_S = \cos 60^\circ \quad (5.13.4)$$

$$c_E = \begin{cases} \cos \left[ 5^\circ + \sin^{-1} \left( \frac{R_E}{r_L} \right) \right] & \text{if } P = E \\ \cos 5^\circ & \text{if } P = M \end{cases} \quad (5.13.5)$$

$$c_M = \begin{cases} \cos 5^\circ & \text{if } P = E \\ \cos \left[ 5^\circ + \sin^{-1} \left( \frac{R_M}{r_L} \right) \right] & \text{if } P = M \end{cases} \quad (5.13.6)$$

where  $c$  is the cosine of one half the total angular dimension of a cone and represents a more convenient way of treating the dimension of a cone in the Star Selection Routine,  $r_L$  is the magnitude of the LM position vector,  $R_E$  is the equatorial radius (6378,166 km) of the Earth, and  $R_M$  is the mean radius (1738.09 km) of the Moon.

#### 5.5.14 PERICENTER - APOCENTER (PERIAPO) SUBROUTINE

The Pericenter - Apocenter Subroutine computes the two body apocenter and pericenter altitudes given the position and velocity vectors for a point on the trajectory and the primary body.

This subroutine is depicted in Fig. 5.14-1.



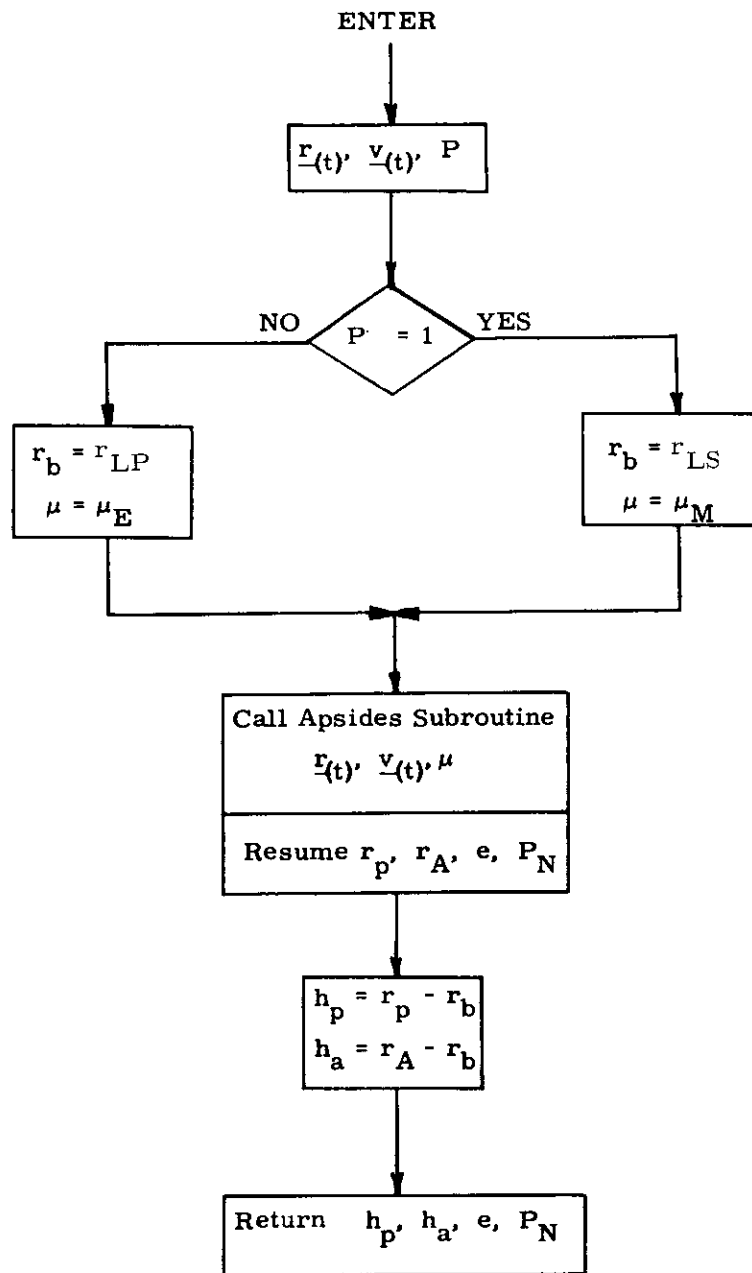


Figure 5.14-1 PERICENTER - APOCENTER SUBROUTINE

