Final Report

NASA CR-92123

PART I

RTCC REQUIREMENTS FOR MISSION G: TRAJECTORY COMPUTERS FOR TLI AND MCC PROCESSORS

William E. Moore
Analytical Mechanics Associates, Inc.

Brody O. McCaffety
Bernard F. Morrey
NASA Manned Spacecraft Center

Report No. 68-4
Contract NAS 9-4036
April 1968

ANALYTICAL MECHANICS ASSOCIATES, INC.
57 OLD COUNTRY ROAD
WESTBURY, N. Y. 11590

NASA CR 921

TABLE OF CONTENTS

Section	Page
SUMMARY AND INTRODUCTION	1
The Trajectory Computers	2
Variables, Stopping Conditions	2
Lunar Orbits	3
ABBREVIATIONS	3
SUBROUTINES	4
Subroutine BURN	5
Subroutine CTBODY	8
Subroutine DGAMMA	14
Subroutine EBETA	16
Subroutine EPHM	17
Subroutine ELEMT	17
Subroutine FCOMP	19
Subroutine LOPC	21
Subroutine LIBRAT	23
Subroutine PATCH	25
Subroutine RBETA	28
Subroutine RNTSIM	31
Subroutine RTASC	33
Subroutine RVIO	35
Subroutine SCALE	36
Subroutine TLIBRN	38
Subroutine TLMC	38
Subroutine XBETA	39
FLOW CHARTS	42
1 Real-Time Applications of the Generalized Iterator	42
2 Translunar Midcourse First Guess Trajectory Computer	44
3 Functional Flow of Analytical Trajectory Computer for Conic Mission	45
4 Integrating Trajectory Computer	51
TABLE I - Independent Variables for the Trajectory Computers	57
TABLE II - Dependent Variables for the Trajectory Computers	58
TABLE III - Basic Modules used in Trajectory Computers	60
REFERENCES	61

RTCC REQUIREMENTS FOR MISSION G: TRAJECTORY COMPUTERS FOR TLI AND MCC PROCESSORS

By Brody O. McCaffety, Bernard F. Morrey, and William E. Moore

SUMMARY AND INTRODUCTION

This note is the last of a series documenting the Generalized Iterator as used in the RTCC translunar injection and midcourse correction processors for Mission G. The mathematical formulation of the iterator itself is completely general and is documented in reference 1; the program setups giving the various mission options provided by the processors have been documented in references 2 and 3. This note gives the trajectory computers of the translunar injection and midcourse correction processors.

The term "Generalized Iterator" as used here refers to the whole program - supervisor, trajectory computer, and iterator. The iterator is a general formulation that applies to any problem involving the solution of a minimum or maximum value of a given function. The technique has other RTCC applications in addition to its use in the TLI and MCC processors. The supervisor sets the dependent and independent variables in such a way as to solve a desired problem. The trajectory computer indicates the sequence of events or computations needed to generate the desired trajectory. This note gives the functional and detailed information about the trajectory computer, the subroutines used in constructing a trajectory, their function, and their algorithms.

There are basically five types of trajectories generated by the TLI and MCC processors:

1. Elliptical trajectories generated out of earth orbit (i.e., E-type ellipses and hybrid ellipses).

2. x, y, z, and t return-to-nominal trajectories generated during translunar coast.

3. Free-return trajectories generated from EPO or translunar coast.

4. Free-return, BAP reoptimized trajectories generated during translunar coast.

5. Nonfree-return, BAP reoptimized trajectories generated during translunar coast.

Flow chart 1 shows these possibilities.

The calculation of each type involves the use of analytical and integrated computations. Conic, or analytical, trajectories are used in first guess routines to generate initial conditions and in optimizations to shorten computation time. Integrated calculations are necessary to provide precision target conditions. An explanation of how these computation modes are used together is contained in references 2 and 3.

The Trajectory Computers

Separate trajectory computers are used in providing first guesses for the midcourse correction, for the conic, and for the precision trajectory computations.

The MCC first guess trajectory computer essentially solves Lambert's problem. Subroutine TLMC computes the first guess trajectory for the MCC. The flow diagram is shown in flow chart 2.

A functional flow diagram of the analytic trajectory computer for conic mission calculations is set forth in flow chart 3. This merely shows the general flow indicating the sequence of state vector calculations, the entry of the appropriate independent variables, the calculation of the dependent variables, and the sequence of the mass history calculations.

The precision propagation of an arc is done using the Herrick-Beta technique documented in the appendix of reference 4. Flow chart 4 shows the functional flow of the precision trajectory computer.

Variables, Stopping Conditions

Independent and dependent variables for the different trajectory computers are shown in tables I and II.

Although the stopping conditions for the computers are indicated implicitly in the flow diagrams, it is worth mentioning them at this time. Integrated trajectories returning to the nominal x, y, and z of the LOI node stop at the time of the node; the nonfree-return BAP options also integrate the same arc and stop on the time of the node obtained from the conic optimization. However, the precision transearth trajectory used in the lunar flyby stops on a flight-path angle of reentry as a function of return velocity instead of a time as is sometimes the case with other processors. Finally, during the iteration process, before the height of LOI is completely correct, the position and velocity vectors at the start of LPO are scaled as shown in subroutine SCALE. These vectors are used to compute the rest of the trajectory, thus retaining the integrity of those independent variables based on the desired height of the orbit; e.g., ΔT_{lo}, ΔT_{lls}.

Lunar orbits.- The initial lunar orbit may be either an ellipse or a circle. Since, after a certain number of revolutions, the spacecraft will be maneuvered into a circular orbit anyway, the program will simulate the ellipse by a circular arc. There are slight differences in the methods of calculation which pertain to the simulation of lunar orbit insertion and of the elliptic orbit itself. These differences do not relate to whether the orbit is integrated or not.

The trajectory computer furnishes the input velocity magnitude at the pericynthion of the ellipse to subroutine BURN; for the circular orbit BURN computes a circular velocity at the current distance. Since in either case BURN reduces the flight-path angle to zero, the ellipse always has its pericynthion at lunar orbit insertion.

Since the state is always related to a circular orbit after lunar orbit insertion, the only other difference is an adjustment of the time to account for the discrepancy in orbital period between the ellipse and the circle used to represent it. This time is the accumulated time difference during the required revolutions before the spacecraft is maneuvered into a circular orbit at the time the LM separates; it will be provided as an input to the program.

ABBREVIATIONS

LAEG	lunar analytical ephemeris generator
BAP	best adaptive path
EOI	earth orbit insertion
EMP	earth-moon plane
EPO	earth parking orbit
LLM	lunar landing mission
LOI	lunar orbit insertion
LOPC	lunar orbit plane change prior to lunar module ascent
LPO	lunar parking orbit
MCC	midcourse correction
RTCC	Real-Time Computer Complex
TEI	transearth injection
TLI	translunar injection

SUBROUTINES

The subroutines and computation modules used in the trajectory computers are listed in table III. The subroutines involved include

1. BURN - simulates impulsive thrusting for application of a delta velocity magnitude, delta azimuth, and delta flight-path angle in the topocentric reference frame.

2. CTBODY - used for propagation of a conic state for a specified time interval.

3. DGAMMA - determines the universal conic variable from periapsis to the nearest specified flight-path angle.

4. EBETA - determines the interval in the universal conic variable from a given state to periapsis.

5. ELEMT - calculates a set of orbital elements from a given state vector, time, and central body constant.

6. EPHM - obtains earth and moon states relative to each other, solar position, and a precession-nutation-libration direction cosine matrix from the magnetic tape ephemeris.

7. FCOMP - evaluates the universal conic functions for a specified value of the universal conic variable.

8. LIBRAT - performs librations upon an input state vector and does a reference transformation.

9. LOPC - computes the size and effect of the lunar orbit plane change (CSM2).

10. PATCH - accomplishes patching of the geocentric and selenocentric vehicle states at the sphere of action of the moon.

11. RBETA - determines the value of the universal conic variable to propagate from a given state to a specified radial magnitude.

12. RNTSTM - determines the reentry and landing conditions, delta time of reentry, and obtains the longitude of landing.

13. RTASC - determines right ascension of the Greenwich meridian.

14. RVIO - transforms a given set of coordinates in Cartesian or spherical form to the other form.

15. SCALE - transforms the actual state vector after LOI to a circular state at a given height.

16. TLIBRN - simulates the translunar injection thrusting maneuver by evaluating precomputed polynomials.

17. TLMC - in control when first guesses for delta azimuth, delta velocity, and delta flight-path angle are determined for translunar abort or midcourse maneuvers.

18. XBETA - propagates a given state through a specified universal conic β to a desired state. The β is the stopping condition for XBETA.

The remaining text of this internal note will be devoted to a detailed description of the input, output, and the mathematics needed for each of the subroutines listed above. All lunar orbit computations will be computed using the lunar radius at the landing site and not the mean radius of the moon.

Subroutine BURN

Function.- Subroutine BURN simulates impulsive thrusting of the vehicle. The ideal velocity equation is used to determine propellant consumption. This subroutine is used for the midcourse, LOI and TEI burns.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
v_c	O	circular velocity
Δv_R	O	characteristic delta velocity
Δv		change in scalar velocity during burn
v_{pc}	I	velocity at pericynthion of the desired ellipse (if ellipse is required)
$\Delta\gamma$	I	change in flight-path angle during burn

Symbols	Input (I), output (O)	Definition
$\Delta\psi$	I	change in azimuth during burn
I_{sp}	I	specific impulse
m_f/m_o	O	ratio of mass after burn to mass before
μ_e	I	constant used to convert pounds force to pounds mass
μ	I	gravitational constant of current reference body
R	I	initial position vector
$\dot{R}$	I	initial velocity vector
γ_o	I	initial flight-path angle
$\dot{R}_1$, $\dot{R}_2$	I	intermediate velocity vectors
R_f	O	final position vector
$\dot{R}_f$	O	final velocity vector

Method.- The vector R_f is the same as R; that is, the routine assures that the position does not change during the maneuver. Compute

$$r = \sqrt{R \cdot R}$$

$$v = \sqrt{\dot{R} \cdot \dot{R}} \ .$$

If a circular state after the burn is specified, put

$$v_c = \sqrt{\frac{\mu}{r}}$$

$$\Delta v = v_c - v$$

$$\Delta\gamma = - \gamma_o \ .$$

If an elliptical state is specified, put

$$\Delta v = v_{pc} - v$$

$$\Delta\gamma = -\gamma_o \quad .$$

In the other more general option v, Δγ, and Δψ are all inputs. Compute

$$d = R \cdot \dot{R}$$

$$h = |\dot{R} \times \dot{R}| \;.$$

$$\dot{R}_1 = \dot{R} \cos \Delta\gamma + \frac{v^2 R - d\dot{R}}{h} \sin \Delta\gamma$$

$$\dot{R}_2 = \frac{2R\,(R \cdot \dot{R}_1)}{r^2} \sin^2 \frac{\Delta\psi}{2} + \dot{R} \cos \Delta\psi - \frac{R \times \dot{R}_1}{r} \sin \Delta\psi$$

$$\dot{R}_f = \dot{R}_2 \left(1 + \frac{\Delta v}{v}\right),$$

which is the velocity vector part of the state S_f after the burn.

$$(\Delta v_R)^2 = \Delta v^2 + 4v\,(v + \Delta v)\left(\sin^2 \frac{\Delta\gamma}{2} + \frac{h^2 \cos \Delta\gamma - hd \sin \Delta\gamma}{r^2\, v^2} \sin^2 \frac{\Delta\psi}{2}\right)$$

which furnishes the characteristic velocity.

Finally, the mass ratio is

$$\frac{m_f}{m_o} = \exp\left(\frac{-\sqrt{(\Delta v_R)^2}}{I_{sp}\mu_e}\right) .$$

Subroutine CTBODY

Function.- Subroutine CTBODY determines the propagated state at a specified time, Δt, from a given epoch state. This is the classical problem of Kepler and must be solved iteratively due to the transcendental relationship between time and the anomalies.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
K	I	central body indicator
r_o	I	position vector magnitude
v_o	I	velocity vector magnitude
μ	I	gravity constant
α	O	square of universal variable divided by semimajor axis
F_1, F_2, F_3, F_4	O	functions of the universal variable
a		semimajor axis
R_o	I	initial position vector
$\dot{R}_o$	I	initial velocity vector
t_o	I	initial time
R_f	O	final position vector
$\dot{R}_f$	O	final velocity vector
r_m		radius of moon
J		$2/3\ J_2$, second harmonic of moon's gravity
t_f	I	final time

Method.- Determine the interval of propagation

$$\Delta t = t_f - t_o .$$

If $|\Delta t| < 10^{-13}$, the final state is just the initial state, and the operation is complete; if not

$$\frac{1}{a} = \frac{2}{r_o} - \frac{v_o}{\mu}$$

$$D_o = R_o \cdot \dot{R}_o .$$

A first guess of the universal variable for the Newton-Raphson iteration is made as

$$\beta = \frac{1}{5}\left(\Delta t \frac{\sqrt{\mu}}{r_o}\right)$$

$$\alpha = -\frac{\beta^2}{a} . \qquad (1)$$

Subroutine FCOMP is entered to obtain F_1, F_2, F_3, and F_4, and the time equation is evaluated

$$t = \left[\beta^2 F_1 + \frac{D_o}{\sqrt{\mu}} \beta F_2 + r_o F_3\right] \frac{\beta}{\sqrt{\mu}}$$

$$r = \frac{D_o}{\sqrt{\mu}} \beta F_3 + \beta^2 F_2 + r_o F_4 .$$

Increment β:

$$\beta = \beta + (\Delta t - t) \frac{\sqrt{\mu}}{r} . \qquad (2)$$

The time equation is again evaluated with the new value of β, and the Newton-Raphson iteration, equation (2), continues until the covergence tolerance of 1×10^{-12} is met:

$$\left|\frac{t - \Delta t}{\Delta t}\right| < 10^{-12} . \qquad (3)$$

Exit with an error message if no convergence is obtained after, say, 10 iterations. (See flow chart on next page.)

As the iterations proceed, β will move in the same direction until it is very close to the answer. To protect against the tolerance of 10^{-12} in equation (3) being too tight, the signs of successive values of $\Delta t - t$ are compared. If two successive iterations should have different signs before equation (3) is satisfied, β is replaced by the average of the two values associated with these iterations, and the process is repeated until the relative difference between two values being averaged is less than 10^{-14}.

With the universal variable determined, the state at the final time is built.

$$f = 1 - \frac{\beta^2 F_2}{r_o}$$

$$g = t - \frac{\beta^3 F_1}{\sqrt{\mu}}$$

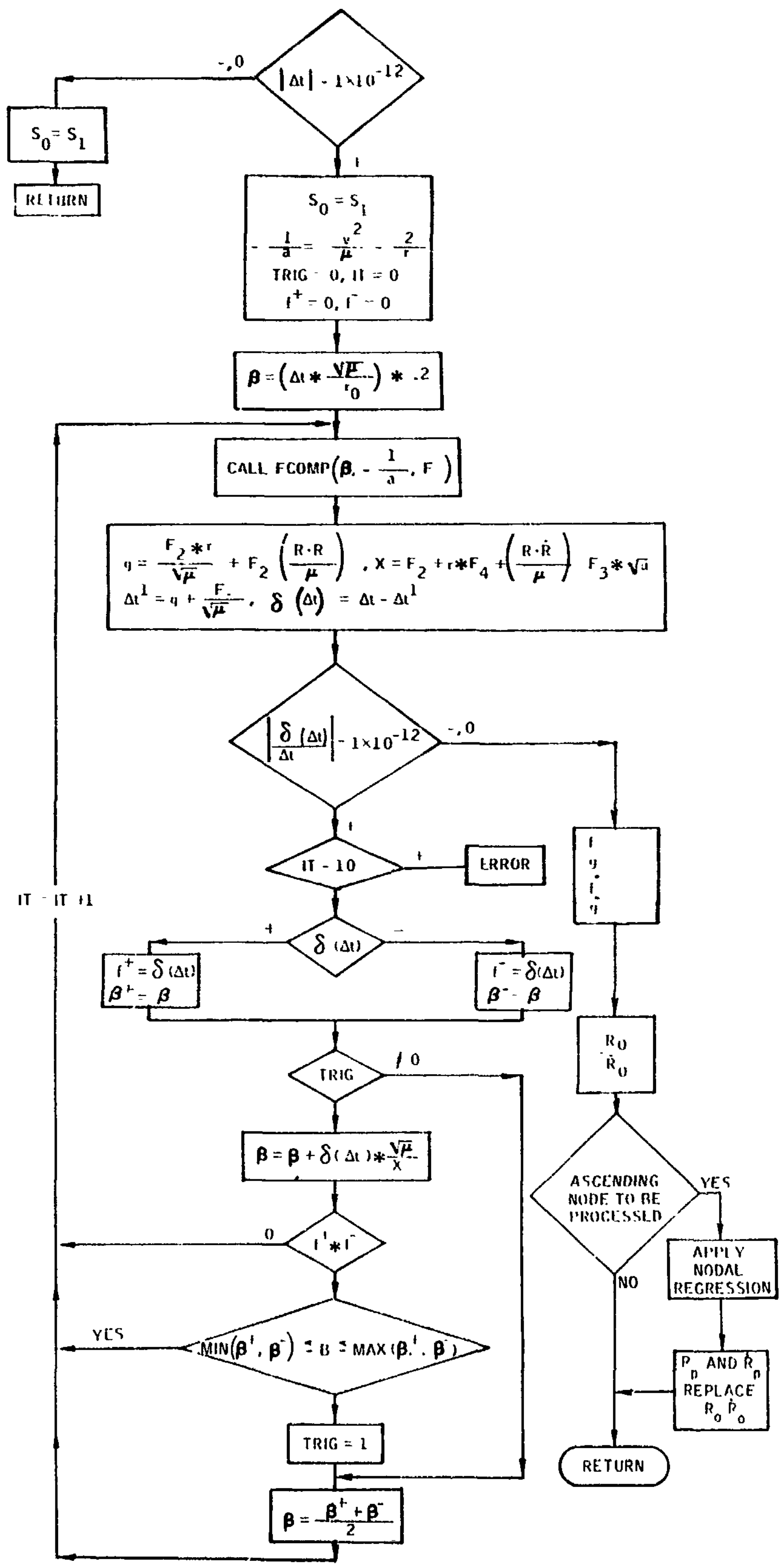
$$\dot{f} = - \frac{\sqrt{\mu}\beta F_3}{r_o r}$$

$$\dot{g} = 1 - \frac{\beta^2 F_2}{r}$$

$$R_f = fR_o + g\dot{R}_o$$

$$\dot{R}_f = \dot{f}R_o + \dot{g}\dot{R}_o$$

Check to see if the ascending node is to be precessed or not. If not, R, $\dot{R}$ are output. Otherwise (using time t_f) rotate R, $\dot{R}$ to selenographic coordinates G_o, $\dot{G}_o$. Let the components of G_o and $\dot{G}_o$ be x, y, z and $\dot{x}$, $\dot{y}$, $\dot{z}$, respectively.


-,0
S0 = S1
RETURN
S0 = S1
TRIG = 0, IT = 0
CALL FCOMP
-,0
IT - 10
ERROR
IT = IT +1
TRIG
/ 0
0
YES
TRIG = 1
ASCENDING NODE TO BE PROCESSED
YES
NO
APPLY NODAL REGRESSION
REPLACE
RETURN

Compute

$$n_1 = \dot{z}x - z\dot{x}$$

$$n_2 = \dot{z}y - z\dot{y}$$

$$n = \sqrt{n_1^2 + n_2^2} \quad .$$

If $n \leq 10^{-12}$, return without precessing the node. Otherwise compute

$$\cos \Omega = \frac{n_1}{n}$$

$$\sin \Omega = \frac{n_2}{n}$$

$$H = \frac{G_o \times \dot{G}_o}{|G_o \times \dot{G}_o|} \quad .$$

Let the components of H be h_1, h_2, h_3. Then

$$\cos i = h_3$$

and

$$\sin i = \sqrt{h_1^2 + h_2^2} \quad .$$

Compute

$$\Delta\Omega = - J r_m^{\ 2} \sqrt{\mu} \cos i \left(\frac{1}{a}\right)^3 \left(\frac{1}{a}\right)^{\frac{1}{2}} \Delta t$$

$$= - 1.14161 \times 10^{-5} \cos i \left(\frac{1}{a}\right)^3 \left(\frac{1}{a}\right)^{\frac{1}{2}} \Delta t$$

$$d = G_o \cdot \dot{G}_o$$

$$r^2 = R^2 = G_o^{\ 2}$$

$$v^2 = \dot{R}^2 = \dot{G}_o{}^2$$

$$N = \begin{bmatrix} \cos \Omega \cos \Delta\Omega - \sin \Omega \sin \Delta\Omega \\ \sin \Omega \cos \Delta\Omega + \cos \Omega \sin \Delta\Omega \\ 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -\cos i \; (\sin \Omega \cos \Delta\Omega + \cos \Omega \sin \Delta\Omega) \\ \cos i \; (\cos \Omega \cos \Delta\Omega - \sin \Omega \sin \Delta\Omega) \\ \sin i \end{bmatrix}$$

$$G = \frac{\dot{z}r^2 - zd}{n} N + \frac{z \; |G_o \times \dot{G}_o|}{n} M$$

$$G = \frac{\dot{z}d - zv^2}{n} N + \frac{\dot{z} \; |G_o \times \dot{G}_o|}{n} M .$$

Using the same time t_f, rotate G, $\dot{G}$ back into selenocentric coordinates R_p, $\dot{R}_p$.

Subroutine DGAMMA

Function.- Subroutine function DGAMMA determines the value of the universal variable necessary to obtain a state at a desired flight-path angle, given the initial position magnitude and the reciprocal of the semimajor axis.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
r_o	I	magnitude of position vector at periapsis
$1/a$	I	reciprocal of semimajor axis
γ	I	flight-path angle
H	O	hyperbolic eccentric anomaly
E	O	elliptic eccentric anomaly
β	O	universal variable
e	O	eccentricity

Method.- Since the given state is at periapsis,

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

$$c = \sqrt{\left|\frac{2r_o}{a} - \frac{r_o^2}{a^2}\right|} .$$

If $\frac{1}{a} < 0$, the orbit is hyperbolic:

$$H = \ln\left[\frac{c}{e}\tan\gamma + \sqrt{1 + \left(\frac{c\tan\gamma}{e}\right)^2}\right]$$

$$\beta = H\sqrt{|a|} .$$

If $\frac{1}{a} > 0$, the orbit is elliptic:

$$\sin E = \frac{c \tan \gamma}{e}$$

$$E \quad \tan^{-1}\left(\frac{\sin E}{\sqrt{1 - \sin^2 E}}\right) \quad \text{when } -\frac{\pi}{2} < E < \frac{\pi}{2}$$

$$\beta = E\sqrt{a} \ .$$

If $\frac{1}{a} = 0$, the orbit is parabolic:

$$\beta = (\sin \gamma/\cos\gamma) \sqrt{2r_o} \ .$$

Remarks.- On an ellipse, the eccentric anomaly is double-valued with respect to the flight-path angle. As is apparent from the equation for E, the algorithm always gives the solution nearer periapsis.

This formulation does not provide for optimizing the same trajectory arc from a hyperbolic energy through parabolic to an elliptical energy.

For the elliptic case, γ may be such that $|\sin E| > 1$. In this instance, γ cannot be achieved, and there is an error.

Subroutine EBETA

Function.- Determines the universal variable necessary to obtain the state at periapsis.

Nomenclature.-

Symbol	Input (I), output(O)	Definition
$1/a$	O	reciprocal of the semimajor axis
R_o	I	initial position vector
$\dot{R}_o$	I	initial velocity vector
r_o		magnitude of initial position vector
v_o		magnitude of initial velocity vector
β	O	universal variables
μ	I	gravitational constant
E		elliptical eccentric anomaly
H		hyperbolic eccentric anomaly
e		eccentricity

Method.- The universal variable and the state at periapsis are determined by

$$D_o = R_o \cdot \dot{R}_o$$

$$1/a = 2/r_o - v_o^2/\mu \ .$$

If $a > 0$, orbit is elliptic:

$$e \cos E = 1 - \frac{r_o}{a}$$

$$e \sin E = D_o/\sqrt{\mu a}$$

$$E = \tan^{-1}(e \sin E / e \cos E)$$

$$\beta = - E\sqrt{a} \ .$$

If 1/a = 0, the orbit is parabolic:

$$\beta = - \frac{D_o}{\sqrt{\mu}} \ .$$

If a < 0, the orbit is hyperbolic:

$$e \cosh H = 1 - r_o/a$$

$$e \sinh H = D_o / \sqrt{\mu |a|}$$

$$H = \ln\left[\frac{e \cosh H + e \sinh H}{|(e \cosh H)^2 - (e \sinh H)^2|}\right]$$

$$\beta = - H\sqrt{|a|} \ .$$

This formulation does not provide for optimizing the same trajectory arc from a hyperbolic energy through parabolic to an elliptic energy.

Subroutine EPHM

Function.- Ephermis subroutines locate, transmit into core, and interpolate data from an ephemeris tape. From this data, earth and moon states relative to each other, solar position, and a precession-nutation-libration direction cosine matrix are obtained.

Remarks.- The ephemeris subroutines used in the RTCC will be system subroutines.

Subroutine ELEMT

Function.- Calculates a set of orbital elements from a given state vector, time, and central body constant.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
R	I	position vecotr
$\dot{R}$	I	velocity vector

Symbol	Input (I), output (O)	Definition
r	I	magnitude of position vector
v	I	magnitude of velocity vector
t	I	initial time
H	O	angular momentum vector per unit mass
μ	I	gravity constant
a	O	semimajor axis
e	O	eccentricity
Δt	O	time to periapsis
i	O	inclination of conic
ω_p	O	argument of periapsis
Ω	O	right ascension of the ascending node
n	O	mean motion
P	I	period
η	O	true anomaly

Method.- Given R, V, t, μ the following items are calculated:

$$\frac{1}{a} = \left(\frac{2}{|R|} - \frac{|\dot{R}|^2}{\mu}\right)$$

$$e = \sqrt{\left(1 - \frac{|R|}{a}\right)^2 + \frac{(R \cdot V)^2}{\mu a}}$$

$$H = R \times \dot{R}$$

$$i = \cos^{-1}\left(\frac{h_z}{|H|}\right)$$

$$n = \frac{\mu^{1/2}}{|a|^{3/2}}$$

$$\eta = \tan^{-1}\left(\frac{|H|(R \cdot \dot{R})}{|H|^2 - \mu|R|}\right)$$

$$P = \frac{2\pi a\sqrt{a}}{n} .$$

All equations but the ones for η and P apply for all conics; the equation for η does not apply to circular orbits and the equation for P does not apply to parabolas and hyperbolas.

Subroutine FCOMP

Function.- Subroutine FCOMP determines the functions of the universal variable necessary to express two-body state quantities given an epoch state. The functions are well defined by circular and hyperbolic functions except as the universal variable approaches zero. To avoid numerical difficulty, a series expansion must be used. To avoid discontinuities, the same expansion is always used. FCOMP is used by XBETA and CTBODY to evaluate the functions of the universal constant.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
F_i	O	functions of the universal variable
α	I	parameter needed to obtain F

Method.-

$$F_j = \sum_{i=0}^{\infty} \frac{\alpha^i}{(2i + 4 - j)!} \quad j = 1,2 .$$

This formulation for the series is used to compute F_1 and F_2. F_3 and F_4

are computed by the formulas:

$$F_3 = \alpha F_1 + 1$$

$$F_4 = \alpha F_2 + 1\,.$$

The number n of the term to be used in the series is determined as follows:

If $\lvert\alpha\rvert <$	n equals
2^{-7}	5
2^{-5}	6
2^{-3}	7
2^{-2}	8
2^{-1}	9
1	10
2	11
4	13
8	15
16	18
32	21
64	25
128	30
256	38
512	46

The series are summed backward;

$$m = 2n + 1$$

$$F_1^{(0)} = F_2^{(0)} = 0$$

$$F_1^{(k)} = \left(\alpha F_1^{(k-1)} + 1\right) \frac{1}{(m - 2k + 2)(m - 2k + 1)} \text{ for } k=1, \cdots, n$$

$$F_2^{(k)} = \left(\alpha F_2^{(k-1)} + 1\right) \frac{1}{(m - 2k + 1)(m - 2k)} \text{ for } k=1, \cdots, n.$$

Finally,

$$F_1 = F_1^{(n)} \text{ and } F_2 = F_2^{(n)}.$$

Coefficients of the form $\frac{1}{m(m - 1)}$ can be precomputed once for all and can be stored with the program.

If $\alpha < -4\pi^2$, the computation may be shortened as follows:

$$\Theta = \text{DMOD } (\sqrt{-\alpha}, 2\pi)$$

$$\overline{\alpha} = - \Theta^2.$$

Use $\overline{\alpha}$ in the series instead of α, obtaining $F_i\ (\overline{\alpha})$. Finally

$$F_4(\alpha) = F_4(\overline{\alpha})$$

$$F_3(\alpha) = F_3(\overline{\alpha})\ \Theta/\sqrt{-\alpha}$$

$$F_2(\alpha) = F_2(\overline{\alpha})\ \Theta^2/(-\alpha)$$

$$F_1(\alpha) = \frac{F_1(\overline{\alpha})\ \Theta^3 + \Theta - \sqrt{-\alpha})}{-\alpha(\sqrt{-\alpha})}.$$

Subroutine LOPC

Function.- Determines the size and effect of the lunar orbit plane change maneuver (CSM2).

Nomenclature.-

Symbol	Input (I), output (O)	Definition
m	I	number of revolutions from first pass over lunar landing site (LLS) to (CSM2 + 1/4)
n	I	number of revolutions from (CSM2 + 1/4) to second pass over LLS.
P	I	period of orbit
S_o	I	state vector at lunar landing
t_o	I	time at lunar landing
Δt_L	I	time from start of lunar orbit to first pass over LLS
Δt_1		time from first pass over LLS to CSM2
S_1		state before CSM2
Δt_2		time from first pass over LLS to second pass over LLS
S_2		predicted state at second pass over LLS
t_L		time of second pass over LLS if no CSM2
S_3	O	state after CSM2
t_3	O	time of CSM2
$\frac{m_f}{m_o}$	O	mass ratio of CSM2 maneuver
R_2		position vector at second pass over LLS in selenographic coordinates

Symbol	Input (I), output (O)	Definition
$\dot{R}_2$		velcoity vector at second pass over LLS in selenographic coord.
L	I	selenographic components of unit vector pointing to the LLS

Method.- Compute

$$\Delta t_1 = \left(m - \frac{1}{4}\right)P.$$

Use CTBODY regressed to propagate S_o from t_o to $(t_o + \Delta t_1)$ obtaining S_1. Then compute

$$t_L = t_o + \Delta t_2 = t_o + (m + n)P.$$

Use CTBODY regressed to propagate S_o from t_o to t_L obtaining S_2. Call LIBRAT at time t_2 to transform S_2 to selenographic coordinates R_2, $\dot{R}_2$.

$$\Delta\psi = -\sin^{-1}\left(\frac{R_2 \times \dot{R}_2}{|R_2 \times \dot{R}_2|} \cdot L\right).$$

Call BURN to get S_3 and $\frac{m_f}{m_o}$, using S_1, $\Delta\psi$ and I_{sp} (where the last two are the only nonzero parameters).

Subroutine LIBRAT

Function.- Subroutine LIBRAT obtains an appropriate transformation matrix and transforms an input state vector in moon reference.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
R	I and O	position vector
$\dot{R}$	I and O	velocity vector

Symbols	Input (I), output (O)	Definition
t	I	time of state vector
K	I	indicator
ME		moon with respect to earth

Method.- Six options exist for converting state vectors to different coordinate systems:

K = 1-Earth-moon plane to selenographic

K = 2-Selenographic to earth-moon plane

K = 3-Earth-moon plane to selenocentric

K = 4-Selenocentric to earth-moon plane

K = 5-Selenocentric to selenographic

K = 6-Selenographic to selenocentric

When the earth-moon plane is involved, a matrix is used to convert either to or from this coordinate system. The formation of this matrix is as follows:

Given the position R_{ME} and velocity V_{ME} of the moon with respect to the earth at each given time,

$$\vec{i} = -\frac{R_{ME}}{\left|R_{ME}\right|}.$$

$$\vec{k} = \frac{R_{ME} \times \dot{R}_{ME}}{\left|R_{ME} \times \dot{R}_{ME}\right|}.$$

$$\vec{j} = \vec{k} \times \vec{i}.$$

Set $A = (\vec{i}, \vec{j}, \vec{k})$ noting that $\vec{i}, \vec{j}, \vec{k}$ are taken as column vectors. Let A^T denote the transpose of A. Then if the selenocentric coordinates in the equatorial system are R, $\dot{R}$, we can say

$$R^1 = A^T R \quad \text{and} \quad \dot{R}^1 = A^T \dot{R}$$

and

$$R = AR^1 \quad \text{and} \quad \dot{R} = A\dot{R}^1.$$

When converting from the selenocentric coordinate system to the selenographic (moon-fixed) coordinate system, the libration matrix is used.

Given the precession-nutation-libration matrix, B, at each given time, and the selenocentric coordinates, R, $\dot{R}$, transform to the selenographic coordinates R", $\dot{R}$" by the following:

$$R'' = BR \quad \text{and} \quad \dot{R}'' = B\dot{R}$$

and

$$R = B^T R'' \quad \text{and} \quad \dot{R} = B^T \dot{R}''.$$

A combination of the two preceding techniques can be used to transform vectors from moon orbit plane to selenographic and the reverse.

Subroutine PATCH

Function.- This subroutine finds a point at which the spacecraft is at a given ratio between the earth and the moon and changes reference bodies at that point.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
R	I and O	position vector
$\dot{R}$	I and O	velocity vector

Symbols	Input (I), output (O)	Definition
t	I and O	time of vector
r		magnitude of position vector
i		reference body subscript: i = 1, primary body i = 2, secondary body
Q	I	direction of patch in time
ERROR	O	error return
μ	I	gravitational constant
a	I	acceleration with respect to body i
β	I	universal variable
KREF	I	primary reference indicator
R_{21}		position of the secondary body with respect to the primary body
r_{21}		magnitude of R_{21}

Method.- In the following, if KREF = 1 (earth reference input), we refer to the earth as the "primary body" and to the moon as the "secondary body". If KREF = 2 (moon reference input), the moon is "primary" and the earth is "secondary".

Subscripts 1 and 2 indicate primary and secondary bodies respectively. Define

$$\text{Ratio} = \frac{r_2}{r_1} = \frac{\text{distance of spacecraft from secondary body}}{\text{distance of spacecraft from primary body}} .$$

Then for a given two-body orbit, Ratio is a function of the orbital parameters, the universal variable β, and moon earth ephemeris data. The procedure is to calculate a second order Taylor's expansion giving

Ratio in terms of the first and second partial derivatives,

$$\frac{d\ \text{Ratio}}{d\beta} \quad \text{and} \quad \frac{d^2\ \text{Ratio}}{d\beta^2},$$

an initial value of β, a corresponding initial value of Ratio, and an increment Δβ to β. Setting Ratio (β + Δβ) equal to the desired value of Ratio we solve the quadratic for Δβ. If the discriminant is less

than zero we set d^2 Ratio/$d\beta^2$ = 0 and solve the linear equation instead. Starting with a guessed value of β, we propagate the initial state (by XBETA) to a final state at the patch with respect to the primary reference body. The position of the secondary body with respect to the primary is obtained from EPHM. A reference change is made, and r_2 and d_2 are calculated.

$$\Delta\text{Ratio} = R - \text{Ratio} \qquad (4)$$

$$\text{where } R = \begin{cases} \dfrac{1}{0.275} & \text{if the moon is the primary body} \\ 0.275 & \text{if the earth is the primary body} \end{cases}$$

$$\frac{d\ \text{Ratio}}{d\beta} = \frac{1}{r_2\sqrt{\mu_1}}\left(d_2 - \frac{r_2^{\,2} d_2}{r_1^{\,2}}\right) \qquad (5)$$

$$\text{where } d_i = R_i \cdot \dot{R}_i \qquad i = 1,2.$$

$$\frac{d^2\ \text{Ratio}}{d\beta^2} = \frac{r_1^{\,2}}{\mu_1}\,\frac{v_2^{\,2} + R_2 \cdot A_2}{r_2} - \frac{d_1\, d_2}{\mu_1\, r_1\, r_2}$$

$$- \frac{d_2^{\,2}\, r_1}{\mu_1\, r_2^{\,3}} - \frac{r_2\, v_1^{\,2}}{\mu_1\, r_1} + \frac{r_2}{r_1^{\,2}} + \frac{2\, d_1^{\,2}\, r_2}{\mu_1\, r_1^{\,3}} \qquad (6)$$

where $v_i^2 = \dot{R}_i \cdot \dot{R}_i$, $i = 1, 2$, and $A_2 = -\frac{\mu_1 R_1}{r_1^3} + \frac{(\mu_1 + \mu_2)}{r_{21}^3} R_{21}$.

$$\Delta\beta = \frac{2\Delta\text{Ratio}}{\frac{d\ \text{Ratio}}{d\ \beta} + \text{sign}\frac{(d\ \text{Ratio})}{d\ \beta}\sqrt{\frac{(d\ \text{Ratio})^2}{d\ \beta} + 2\ \Delta\text{Ratio}\frac{(d^2\ \text{Ratio})}{d\ \beta^2}}}.$$

Replace β by $\beta + \Delta\beta$ and repeat the process until $\Delta\text{Ratio} < 1 \times 10^{-12}$. The last state and time with respect to the secondary body is the output state and time.

The initial first guesses for the earth and moon as primary bodies are the value of β needed to propagate to 50 and 15 e.r., respectively.

Upon further reference to the routine using a given primary body, the last value of distance in that particular primary body is used to derive a first guess for β. This implies that two distances are saved, one for each primary body.

Remarks.- Error returns or indicators – the last variable in the calling sequence is an error indicator which is a logical variable and will return a value of .TRUE. when an error has occurred in the routine. There are three ways that the error indicator can be set up to .TRUE.:

1. If the patch iterative procedure fails to converge within 10 iterations.

2. If the ephemeris data table has not been initialized or the time calculated withing the routine is outside the range of the ephemeris data.

3. If the magnitude of the input position vector is greater than 40 e.r. when the earth is the primary body or 10 e.r. when the moon is the primary body and the conic defined by the input state vector is such that the radius of periapsis is greater than 40 e.r. when the earth is the primary body or 10 e.r. when the moon is the primary body.

Subroutine RBETA

Function.- RBETA determines the universal variable necessary to obtain a state vector at a desired radial magnitude, given an initial state.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
R_o	I	initial position state vector
$\dot{R}_o$	I	initial velocity state vector
r_o	I	magnitude of initial position vector
v_o	I	magnitude of initial velocity vector
Q	I	direction indicator
β	O	universal variable
E		elliptic eccentric anomaly
H		hyperbolic eccentric anomaly
ERROR	O	indicator of error return
μ		gravity constant of reference body
r	I	desired radius magnitude
a		semimajor axis
e		eccentricity

Method.- Subroutine RBETA is restricted to cases where the desired radius magnitude is greater than the initial magnitude. If an orbit is circular, the subroutine gives a return with the error indicator set .TRUE. since any β would suffice if the desired distance is the radius of the circle, and no β exists if the desired radius is not the circular radius. In general, the solution for a desired radius is double-valued; therefore an indicator Q is provided to select the desired solution. If Q = +1, the solution will be ahead of the initial position with respect to the direction of motion; if Q = -1, the solution will be behind the initial position.

Determine the dot product of R_o and $\dot{R}_o$, semimajor axis, and eccentricity.

$$D_o = R_o \cdot \dot{R}_o .$$

$$\frac{1}{a} = \frac{2}{r_o} - \frac{v_o^2}{\mu a} .$$

$$e = \sqrt{\left(1 - \frac{r_o}{a}\right)^2 + \frac{D_o^2}{\mu a}}$$

If 1/a < 0, the orbit is hyperbolic.

$$\cosh H_o = \frac{1}{e}\left(1 - \frac{r_o}{a}\right) .$$

$$\cosh H = \frac{1}{e}\left(1 - \frac{r}{a}\right) .$$

$$H_o = \pm \ln\left(\cosh H_o + \sqrt{\cosh^2 H_o - 1}\right)$$

where the sign is chosen to be the sign of D_o.

$$H = \ln\left(\cosh H + \sqrt{\cosh^2 H - 1}\right) .$$

$$\theta = H_o - QH .$$

$$\beta = Q \, |\theta| \, \sqrt{|a|} .$$

If 1/a > 0, the orbit is elliptic.

$$\cos E_o = \frac{1}{e}\left(1 - \frac{r_o}{a}\right) .$$

$$\cos E = \frac{1}{e}\left(1 - \frac{r}{a}\right) .$$

$$E_o = \pm \tan^{-1} \sqrt{\frac{1 - \cos^2 E_o}{\cos E_o}}$$

where the sign is chosen to be the sign of D_o.

$$E = \tan^{-1} \frac{\sqrt{1 - \cos^2 E}}{\cos E} .$$

$$\theta = E_o - QE.$$

$$\beta = Q \; |0| \; \sqrt{a} .$$

If $1/a = 0$, orbit is parabolic.

$$\beta = \frac{D_o}{\sqrt{\mu}} + Q \sqrt{\frac{D_o{}^2}{\mu} + 2 \; (r - r_o)} .$$

Remarks.- If any of the radicands involving r is less than zero, the distance r is impossible, and the calculation is suspended with error indicator set .TRUE.

Subroutine RNTSIM

Function.- This subroutine determines the reentry and landing conditions of delta time from reentry to landing and longitude of landing.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
λ	O	computed longitude of landing
λ_L	I	longitude of landing
$\Delta\lambda$	O	error in longitude of landing
R	I	position vector at reentry
$\dot{R}$	I	velocity vector at reentry
r	I	magnitude of position vector at reentry
v	I	magnitude of velocity vector at reentry
t	I	time of reentry
RR	I	reentry range, n. mi.

Symbol	Input (I), output (O)	Definition
Δt	I	time from reentry to landing
ϕ_L	O	latitude at landing
α_L	O	right ascension at landing
α_G	O	greenwich right ascension at time of landing
γ	I	flight-path angle at reentry
θ	O	central angle between reentry and landing

Method.- Given R, $\dot{R}$, and RR

$$P = \frac{\dot{R}}{v} \frac{1}{\cos \gamma} - \frac{R}{r} \tan \gamma .$$

$$\theta = RR/3443.933585.$$

$$S = \frac{R}{r} \cos \theta + P \sin \theta .$$

where S is the position at landing.

$$\phi_L = \tan^{-1} \frac{S_z}{\sqrt{S_x^2 + S_y^2}} .$$

$$\alpha_L = \tan^{-1} \frac{S_y}{S_x} .$$

Call RTACS at time t + Δt to get α_G. Then

$$\Delta\lambda = \alpha_L - \alpha_G - \lambda_L .$$

$$\lambda = \alpha_L - \alpha_G$$

Reduce $\Delta\lambda$ by any excess multiples of 2π. If the result is $> \pi$, subtract 2π; if the result is $\leq -\pi$, add 2π. Thus, finally $-\pi < \Delta\lambda \leq \pi$.

To allow partial derivatives to be obtained correctly despite the discontinuities inherent in this scheme, the following procedure is applied when computing the trajectories involved in partial derivities calculations. After each nominal trajectory computation, the value of $\Delta\lambda$ is retained. During the perturbed trajectory computations, this value, called $\Delta\lambda_o$, is compared with the current value of $\Delta\lambda$. If $(\Delta\lambda - \Delta\lambda_o) < -\pi$, then $\Delta\lambda$ is replaced by $\Delta\lambda + 2\pi$; if $(\Delta\lambda - \Delta\lambda_o) > \pi$, $\Delta\lambda$ is replaced by $\Delta\lambda - 2\pi$.

Subroutine RTASC

Function.- Subroutine RTASC determines right ascension of the Greenwich meridian.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
X	I	epoch year
Y	I	year of base time in universal time
d	I	day of base time in universal time
h	I	hours of base time in universal time
ψ	O	right ascension of Greenwich at base time

Method.- The following steps will be used in the initialization to determine the right ascension.

1. Compute number of leap years between 1950 and x, not counting x. n = integral part of $\frac{x - 1949}{4}$

2. Compute the beginning of the Besselian year

$$d_{BY} = 0.923329 + 0.2421947\ (x - 1950) - 3.08 \times 10^{-8}(x - 1950)^2 - n$$

3. Compute the daily precessional rate at the epoch year.

$$m = 6.11907 \times 10^{-7} + 3.70 \times 10^{-12} (x - 1950)$$

4. Compute the precession from Jan 0.0 to the beginning of the Besselian year.

$$\Delta = md_{BY}$$

5. Compute the number of days from Jan 1.0, 1950, Jan 0.0 of the epoch year.

$$d^* = 365 (x - 1950) + n - 1$$

6. Compute the right ascension of Greenwich with respect to the mean equinox at Jan 0.0 of the epoch.

$$\psi_E = 1.7294449276386 - 0.0041554274551 (x - 1950) + 0.0172027914513 n$$
$$+ 5.0640897 \times 10^{-15} d^{*2}$$

7. Compute the number of days, d', from Jan 0.0 of the epoch year to base time.

$$d' = d \quad \text{if } x = y$$

$$d' = \left.\begin{matrix} d - 365 & \text{if y is not a leap year} \\ d - 366 & \text{if y is a leap year} \end{matrix}\right\} \quad \text{and } x \neq y$$

8. Compute the Greenwich hour angle at base time with respect to the mean equinox fixed at Jan 0.0 of the epoch year.

$$\psi_{BO} = (\psi_E + 0.0172021795 43\, d' + 0.2625161452801) \bmod 2\pi$$

9. Correct to mean equinox fixed at the beginning of the Besselian year.

$$\psi_B = \psi_{BO} + \Delta$$

The general computation after initialization is as follows:

1. Input h - hours from base time (universal time)

2. Compute the integral number of days and the hours remaining in the fractional part of a day.

$$d = \text{integral part of } \left(\frac{h}{24}\right)$$
$$h' = h - 24d$$

3. Compute the right ascension of Greenwich at base time

$$\psi = (\psi_B = 0.017202179543\ d + 0.2625161452801\ h') \bmod 2\pi$$

Remarks.- The constant term is the right ascension of Greenwich at Jan 0.0, 1950. The coefficient of (x - 1950) is the difference between a full revolution and 365 times the daily rate.

Subroutine RVIO

Function.- Transform a given set of coordinates in Cartesian or spherical form to the other form.

Nomenclature.-

Symbol	Input (I), output (O)	Definition
R	I and O	position vector
$\dot{R}$	I and O	velocity vector
r	I and O	position magnitude
v	I and O	velocity magnitude
x	I and O	x component of position vector
y	I and O	y component of position vector
z	I and O	z component of position vector
$\dot{x}$	I and O	$\dot{x}$ component of position vector
$\dot{y}$	I and O	$\dot{y}$ component of position vector
$\dot{z}$	I and O	$\dot{Z}$ component of position vector
θ	I and O	latitude
ϕ	I and O	right ascension angle
γ	I and O	flight-path angle
ψ	I and O	azimuth angle

Method.-

Spherical to Cartesian Transformation

$$x = r \cos\phi \cos\theta$$

$$y = r \cos\phi \sin\theta$$

$$z = r \sin\phi$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & -\sin\theta & -\sin\phi\cos\theta \\ \cos\phi\sin\theta & \cos\theta & -\sin\phi\sin\theta \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} v\sin\gamma \\ v\cos\gamma\sin\psi \\ v\cos\gamma\cos\psi \end{bmatrix}$$

Cartesian to Spherical Transformation

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\phi = \sin^{-1}\frac{z}{r}$$

$$\theta = \tan^{-1}\frac{y}{x}$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

$$\gamma = \sin^{-1}\frac{R \cdot \dot{R}}{rv}$$

$$\psi = \tan^{-1}\left(\frac{h_z}{\dot{z}r - zv_r}\right) = \tan^{-1}\left(\frac{x\dot{y} - y\dot{x}}{\dot{z}r - z\frac{R \cdot \dot{R}}{r}}\right)$$

Subroutine SCALE

Function.- Subroutine SCALE transforms the actual state vector after [illegible] to the state of a circular orbit at a given height.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
R_o	I	position vector after LOI
r_o	I	magnitude of position vector after LOI
$\dot{R}_o$	I	velocity vector after LOI
v_o	I	magnitude of velocity vector after LOI
R_f	O	scaled position vector at the beginning of the lunar circular orbit
$\dot{R}_f$	O	scaled velocity vector at the beginning of the lunar circular orbit
h	I	height of scaled lunar circular orbit
μ	I	gravitational constant of the moon
r_m	I	radius of the moon at the landing site

Method.-

$$R_f = R_o\left(\frac{h + r_m}{r_o}\right).$$

$$\dot{R}_f = \sqrt{\frac{\mu}{h + r_m}} .$$

Subroutine TLIBRN

Function.- Subroutine TLIBRN simulates the translunar injection thrusting maneuver by use of a precomputed polynomial.

Remarks.- The method of this subroutine is contained in reference 4.

Subroutine TLMC

Function.- Subroutine TLMC determines the first guesses for delta azimuth, delta velocity, and delta flight-path angle for a translunar state at abort or midcourse.

Nomenclature.-

Symbol	Input(I), output (O)	Definition
S	I and O	state vector
t	I	time of state vector S
t_p	I	nominal time of node
x	I and O	x component of position vector
y	I and O	y component of position vector
z	I and O	z component of position vector
r	I	desired radius at the node
λ	I and O	longitude of node in earth-moon plane system
v	I and O	velocity magnitude at node
γ	I	flight-path angle at node
ψ	I and O	azimuth of node in earth-moon system
Δt	I	amount of change in t_n (for nonfree-return)
t_n	O	adjusted time of node
ERROR	O	flag indicating an error in TLMC

Method.- Compute the adjusted time of node: $t_n = t_p + \Delta t$. The earth-moon plane (EMP) matrix is obtained once for all, by subroutine LIBRAT at the time t_m for use in transforming the EMP coordinates at the node to the selenocentric system.

The next step sets the dependent variable limits, weights, and weight cuts. Three dependent variables, x, y, z, are defined as the components of the position vector at abort or midcourse. They are designated Class 1 variables. The minimum and maximum required values of the position components are found by adding and subtracting a small tolerance (10^{-5} e.r.) to the abort position components.

Having described the dependent variables, the independent variables are set up and given a first guess. The first independent variable is the EMP longitude of node. The first guess for the longitude is $(3.1 - 0.025\ \Delta t)$ radians. This guess places the node behind the moon in the vicinity of the earth-moon line. The second independent variable is the velocity at the node, and the first guess is $\sqrt{0.184 + 0.553/r} - 0.0022\ \Delta t$ e.r./hr. The third independent variable is the azimuth at the node, and the first guess is $-\frac{\pi}{2}$ to obtain a retrograde lunar approach hyperbola.

By forcing the node to lie at the required EMP latitude and to have the required height and flight-path angle, the above independent variables determine the state at the node in the EMP. Subroutine CTBODY is called, and the trajectory is propagated backward to the initial time, t. The generalized iterator then finds the set of independent variables necessary to obtain the dependent variables at abort; that is, the abort position components. Once converged, the differences between the azimuth, flight-path angle, and velocity before abort, and the values after abort necessary to obtain the above node conditions are determined. These values become first guesses for the MCC maneuver.

Subroutine XBETA

Function.- Determines the state vector relative to the initial state for a desired value of the universal variable.

Nomenclature.-

Symbols	Input (I), output (O)	Definition
β	I	universal variable
K	I	central body indicator
F_i		functions of the universal variable
μ		gravity constant
R_o	I	initial position vector
$\dot{R}_o$	I	initial velocity vector
r_o	I	magnitude of initial position vector
v_o	I	magnitude of initial velocity vector
t_o	I	initial time
R	O	fixed position vector
$\dot{R}$	O	fixed velocity vector
t	O	final time

Method.- From the initial state vector, the final state is determined as a function of β.

$$D_o = R_o \cdot \dot{R}_o$$

$$1/a = 2/r_o - v_o^2/\mu$$

$$\alpha = -\beta^2/a$$

Call subroutines FCOMP and determine the functions of the universal variable.

$$t = \left(\beta^2 F_1 + \frac{D_o \beta F_2}{\sqrt{\mu}} + r_o F_3\right) \beta\sqrt{\mu} \; .$$

$$t_f = t_o + t.$$

$$r = \left(\frac{D_o F_3}{\sqrt{\mu}} + \beta F_2\right)\beta + r_o F_4 .$$

$$f = 1 - \frac{\beta^2 F_2}{r_o} .$$

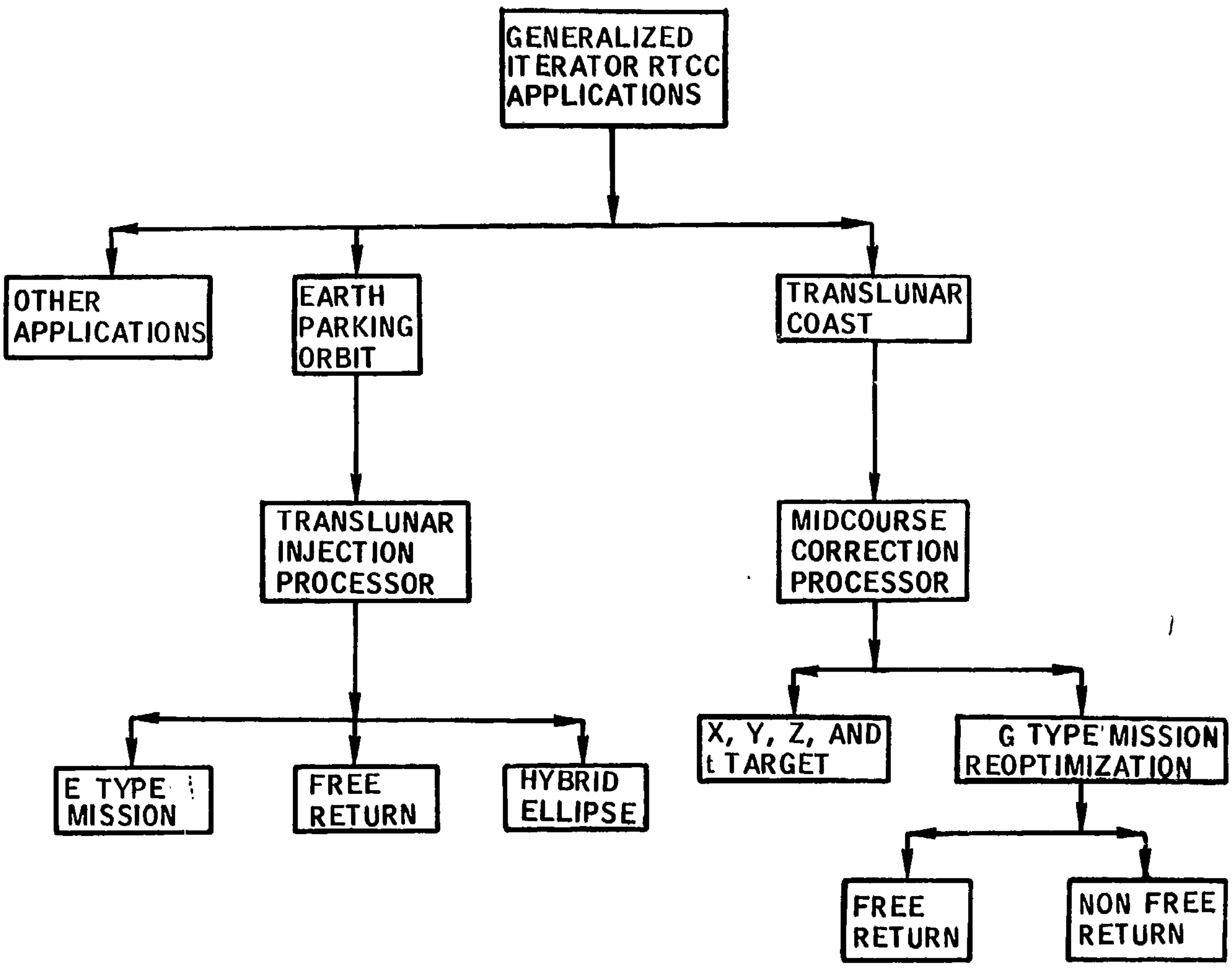
$$g = t - \beta^3 F_1 \sqrt{\mu} .$$

$$\dot{f} = - \sqrt{\mu}\, \beta F_3 / r_o\, r.$$

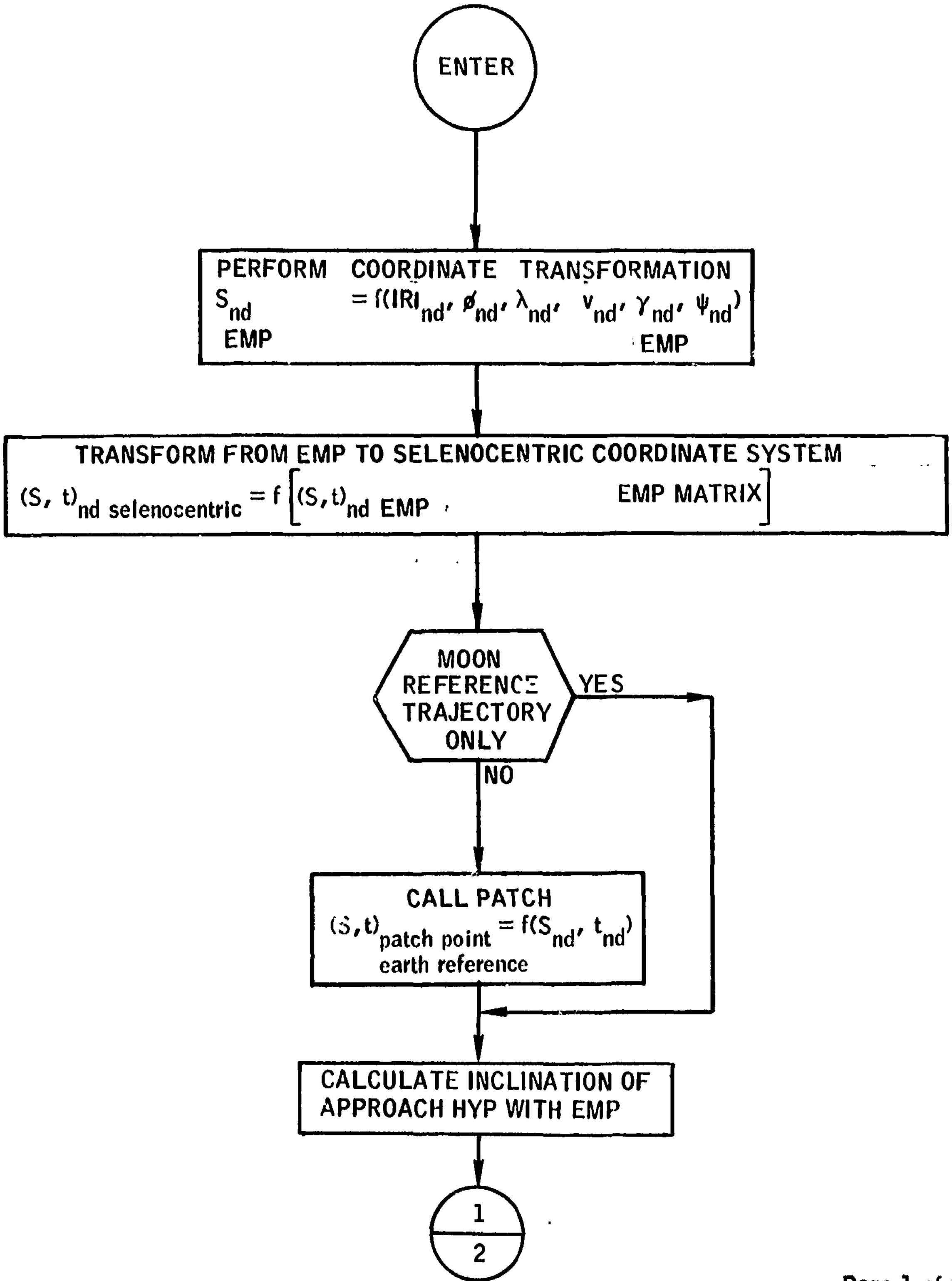
$$\dot{g} = 1 - \beta^2 F_2 / r .$$

$$R = f R_o + g \dot{R}_o .$$

$$\dot{R} = \dot{f} R_o + \dot{g} \dot{R}_o .$$

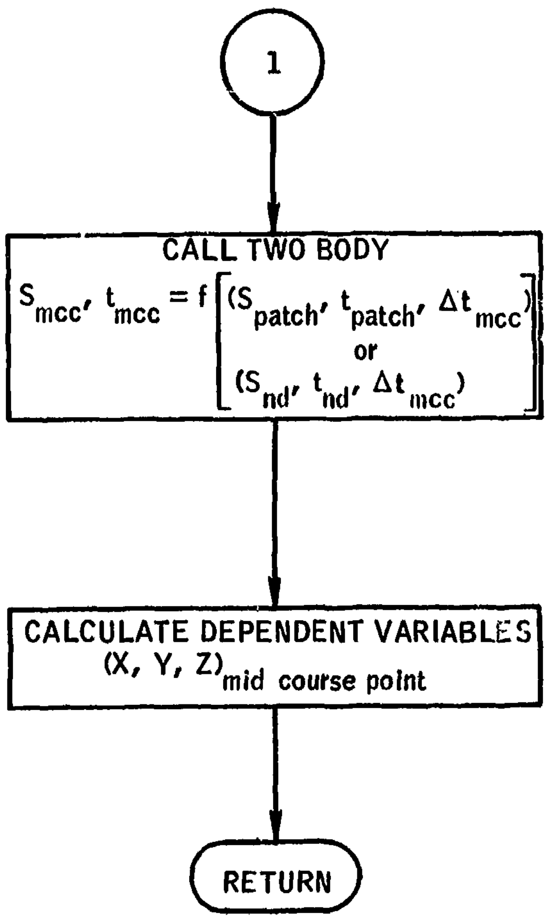


Flowchart 1.- Real time applications of the generalized iterator.



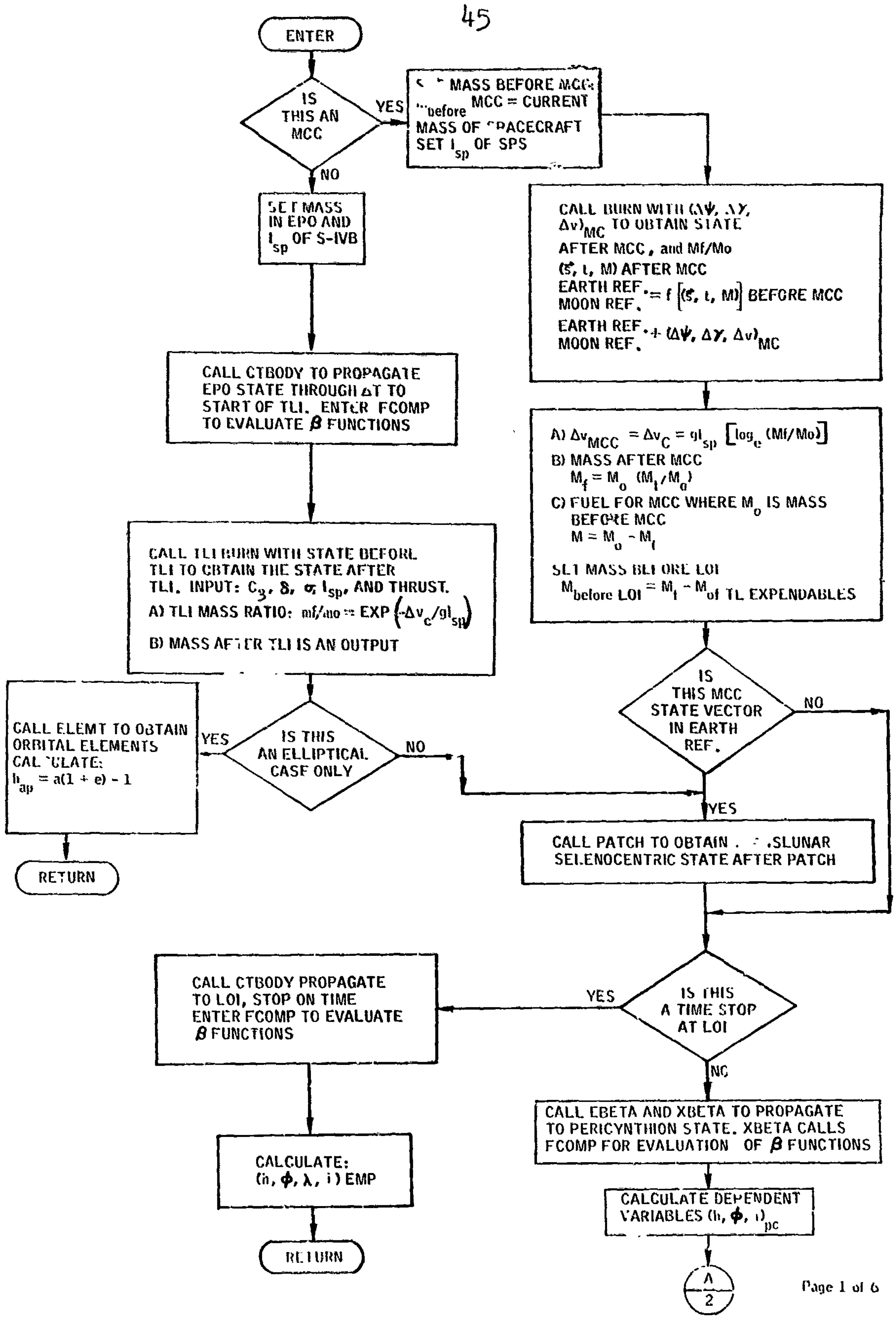
Page 1 of 2

Flow chart 2.- Translunar midcourse first guess trajectory computer.

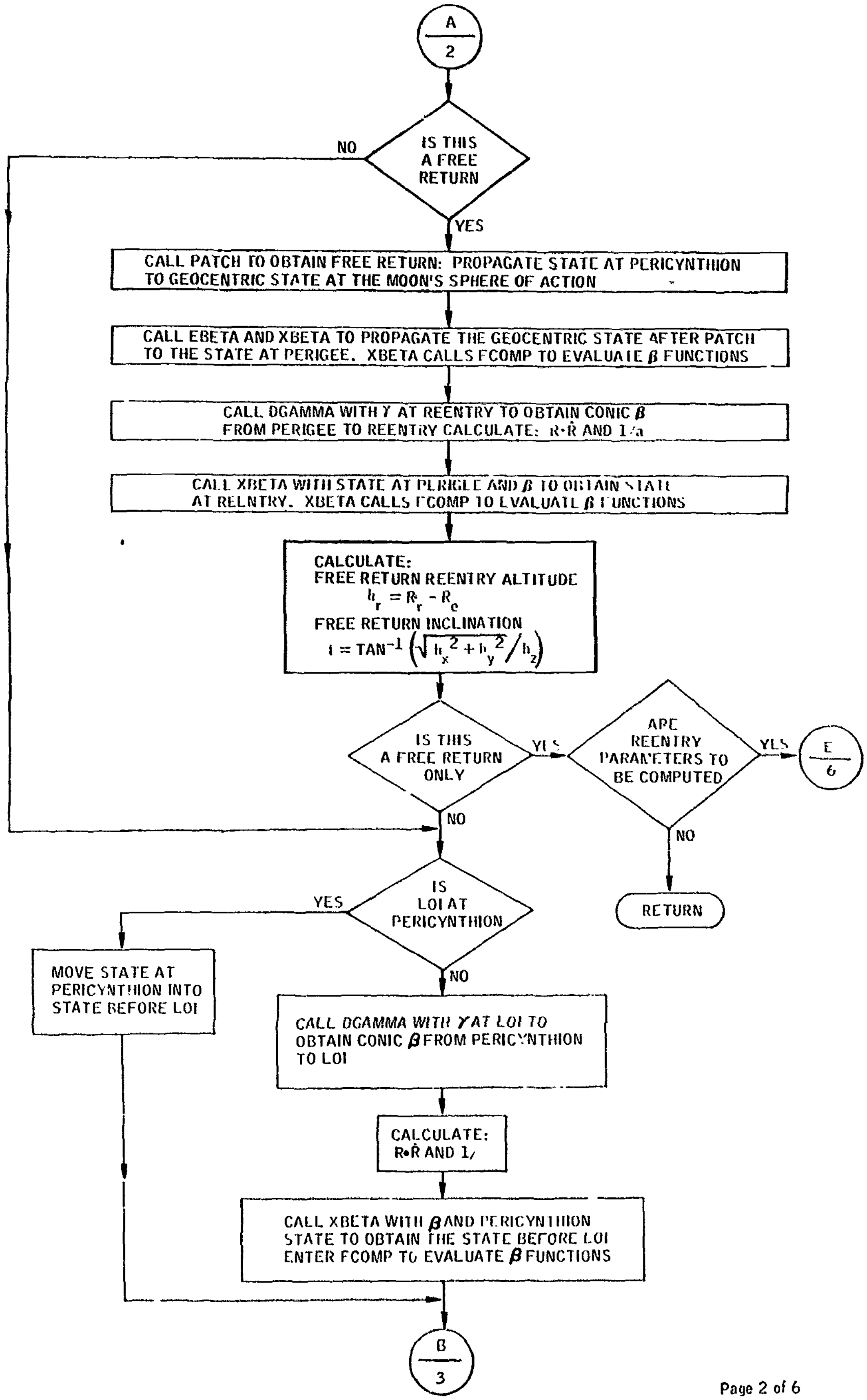


Page 2 of 2

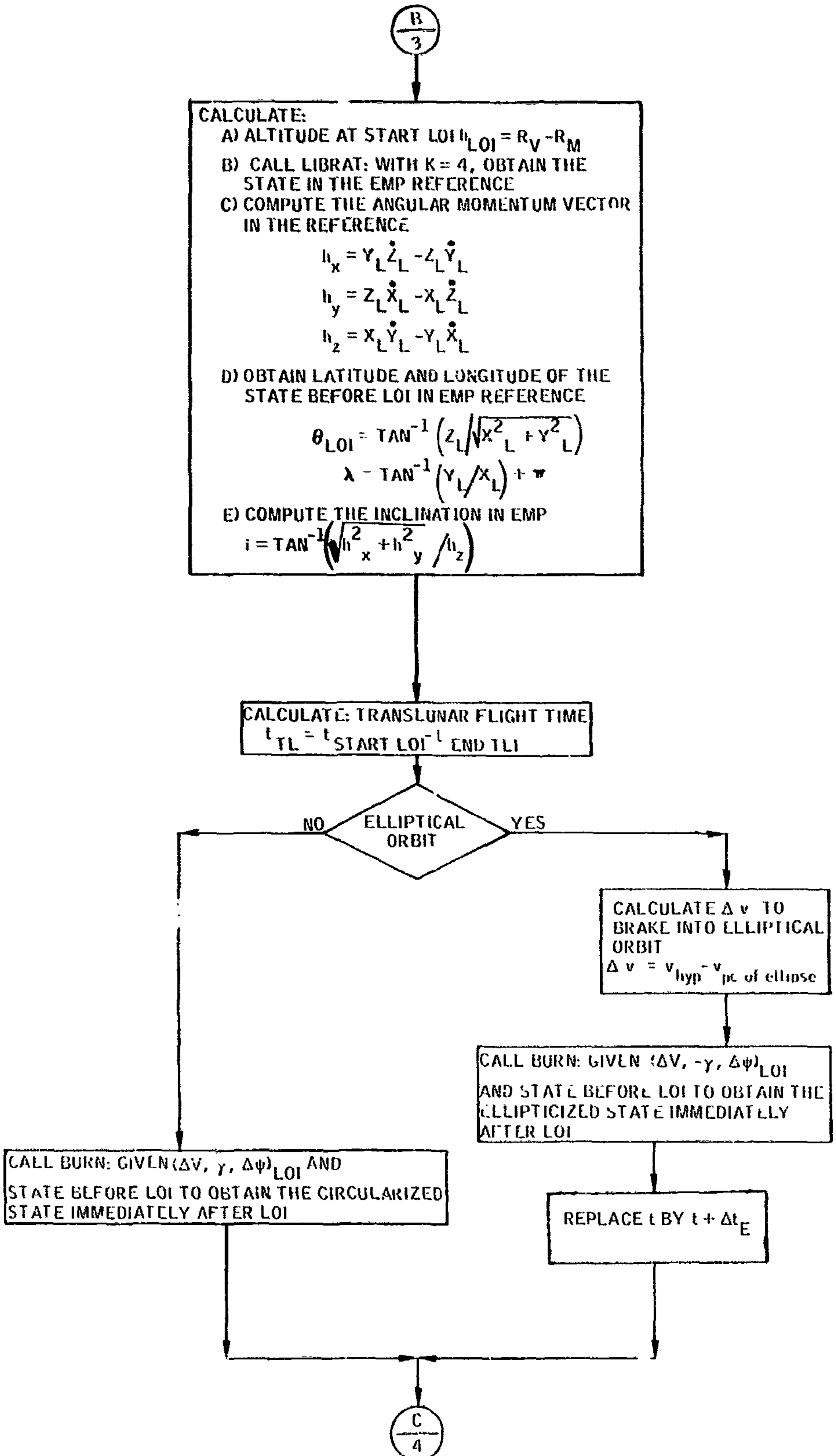
Flowchart 2.- Translunar midcourse first guess trajectory computer - Concluded.



Flow chart 3.- Functional flow of analytical trajectory computer for conic mission.



Flowchart 3.- Funtional flow of analytical trajectory computer for conic mission - Continued.



Flow chart 3.- Functional flow of analytical trajectory computer for conic mission - Continued.

C / 4

A) LOI MASS RATIO: WHERE M_0 IS MASS BEFORE MANEUVER, WHERE M_f IS MASS AFTER

$$M_f / M_0 = \mathrm{EXP}\left[-\left(\Delta v_c / g I_{sp}\right)\right]$$

B) MASS OF SC BEFORE LM SEPARATION

$$M_{SC} = M_{\text{before LOI}}\,(mf / mo) - M_{\text{OF CIRCULARIZATION}} - M_{\text{LUNAR ORBIT EXPENDABLES}}$$

C) ALTITUDE AT START OF LPO: $h_{LO} = R_0 - R_M$

↓

A) CALL SCALE TO CONVERT ACTUAL VELOCITY AND POSITION VECTORS TO SPECIFIED PARKING ORBIT VALUES STATE AT START OF LPO IS OBTAINED BY SCALING STATE AFTER LOI

B) TIME OF SIMULATED LM LUNAR LANDING:

$$T_{LL} = T_{\text{IMMEDIATELY AFTER LOI}} + \Delta T_{\text{FIRST PASS OVER LLS}}$$

↓

CALL CT BODY (REGRESSED) WITH STATE AT START OF LPO AND TIME OF LUNAR LANDING TO OBTAIN CSM STATE AT LANDING. ENTER FCOMP TO EVALUATE β FUNCTIONS.

↓

CALL LIBRAT, WITH K = 5, AND USE THE PNL MATRIX TO ROTATE THE CSM POSITION VECTOR AT FIRST PASS OVER THE LUNAR LANDING SITE FROM SELENOCENTRIC INTO THE SELENOGRAPHIC, R_s.

THE SELENOGRAPHIC LATITUDE AND LONGITUDE OF THE CSM AT LM LANDING TIME:

$$\phi_s = \mathrm{TAN}^{-1}\left(Z_s / \sqrt{X_s^2 + Y_s^2}\right) \qquad L = \frac{R_s}{|R_s|}$$

$$\lambda_s = \mathrm{TAN}^{-1}\left(Y_s / X_s\right)$$

↓

CALL ELEMT COMPUTE PERIOD

↓

CALL LOPC COMPUTE TIME OF CSM_2 PLANE CHANGE

↓

D / 5

Flow chart 3.- Functional flow of analytical trajectory computer for conic mission - Continued.

D / 5

CALCULATE MASS AFTER CSM_2 AND LM RENDEZVOUS

$$M = (M_1 - M_{LM} - \Delta M_1)(Mf/Mo)_{CSM_2} + \Delta M_2$$

WHERE

M_{LM} = MASS OF LM

ΔM_1 = MASS OF ASTRONAUTS AND THEIR EQUIPMENT BEFORE LANDING

ΔM_2 = MASS OF ASTRONAUTS AND EQUIPMENT AFTER RETURN TO CSM

M_1 = MASS BEFORE LM SEPARATION

↓

CALL CT BODY (REGRESSED) PROPAGATE TO STATE BEFORE TEI. ENTER FCOMP TO EVALUATE FUNCTIONS.

$t_{TEI} = t_{LOI} + \Delta t_{LPO}$

MASS BEFORE TEI = $M_{after\ CSM_2} - M_{LO\text{-}EXPENDABLE}$

↓

CALL BURN WITH $(\Delta v, \Delta\psi, \Delta\gamma)_{TEI}$ AND T_{TEI} TO COMPUTE STATE AFTER TEI

MASS AFTER TEI = $M_{before\ TEI}\,(mf/mo)_{TEI}$

↓

CALL PATCH WITH STATE AFTER TEI TO OBTAIN STATE AFTER TE-PATCH

↓

CALL EBETA AND XBETA TO OBTAIN PERIGEE STATE VECTOR. XBETA CALLS FCOMP TO EVALUATE β FUNCTIONS

↓

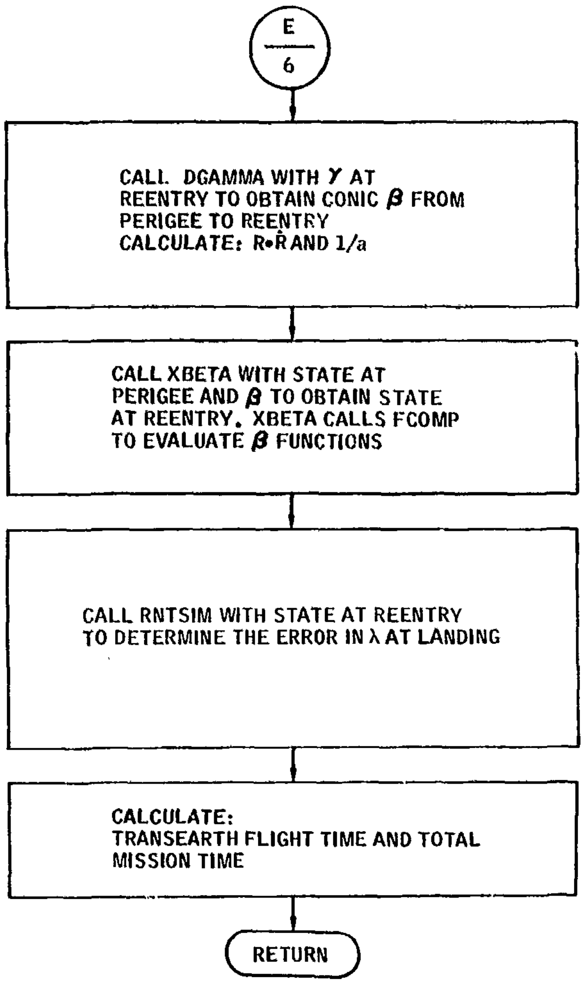
COMPUTE:

INCLINATION OF RETURN

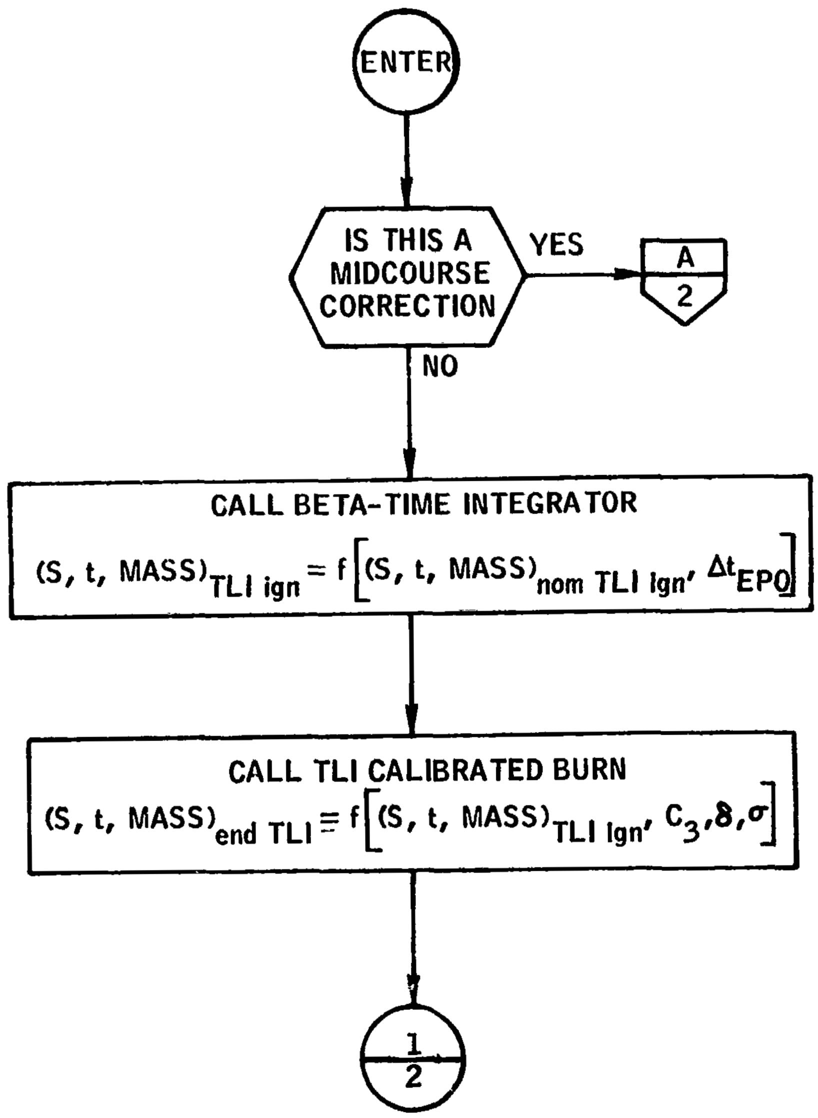
$$= TAN^{-1}\left(\sqrt{h_x^2 + h_y^2}/h_z\right)$$

E / 6

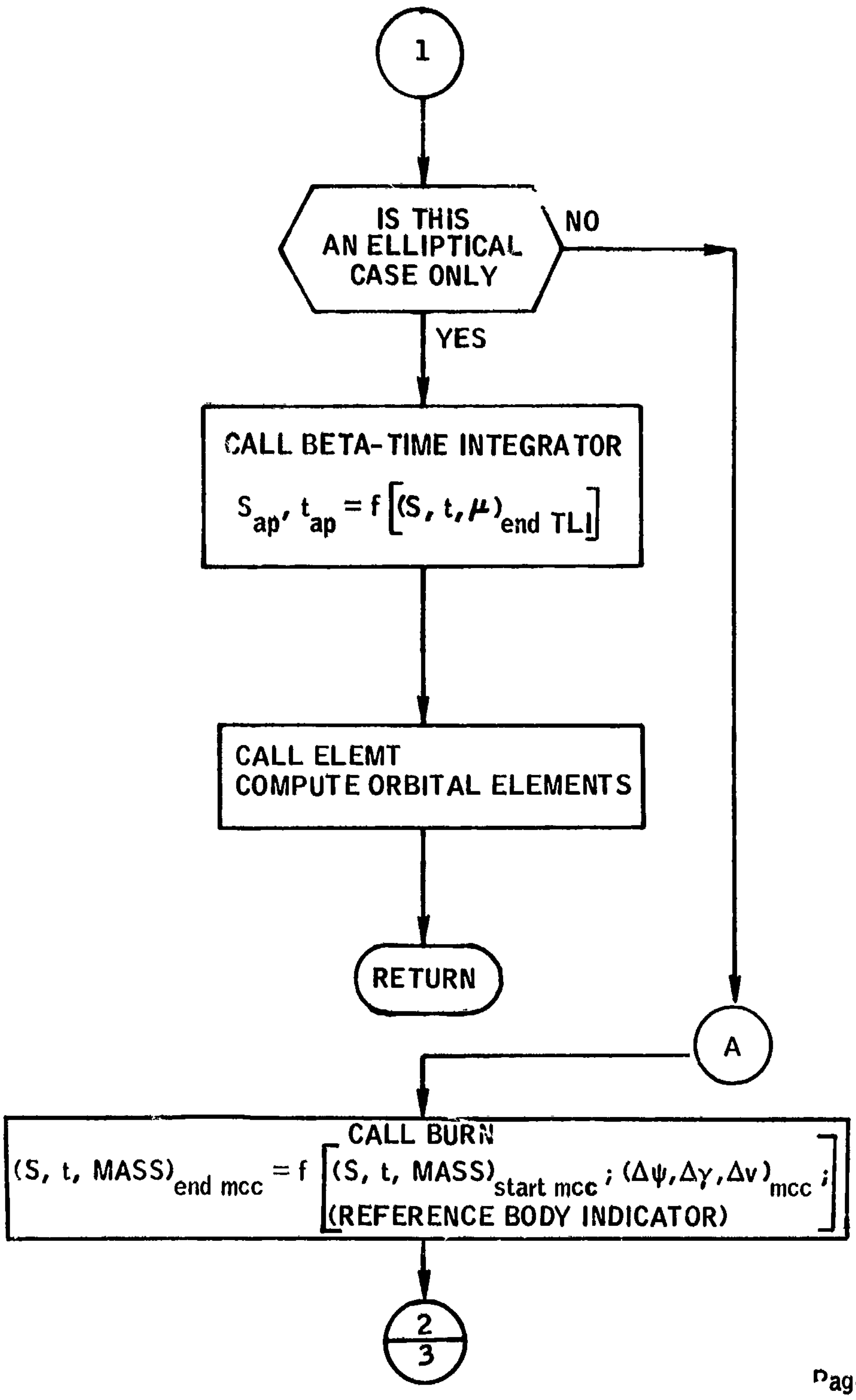
Flowchart 3.- Functional flow of analytical trajectory computer for conic mission - Continued.



Flow chart 3.- Functional flow of analytical trajectory computer for conic mission - Concluded.

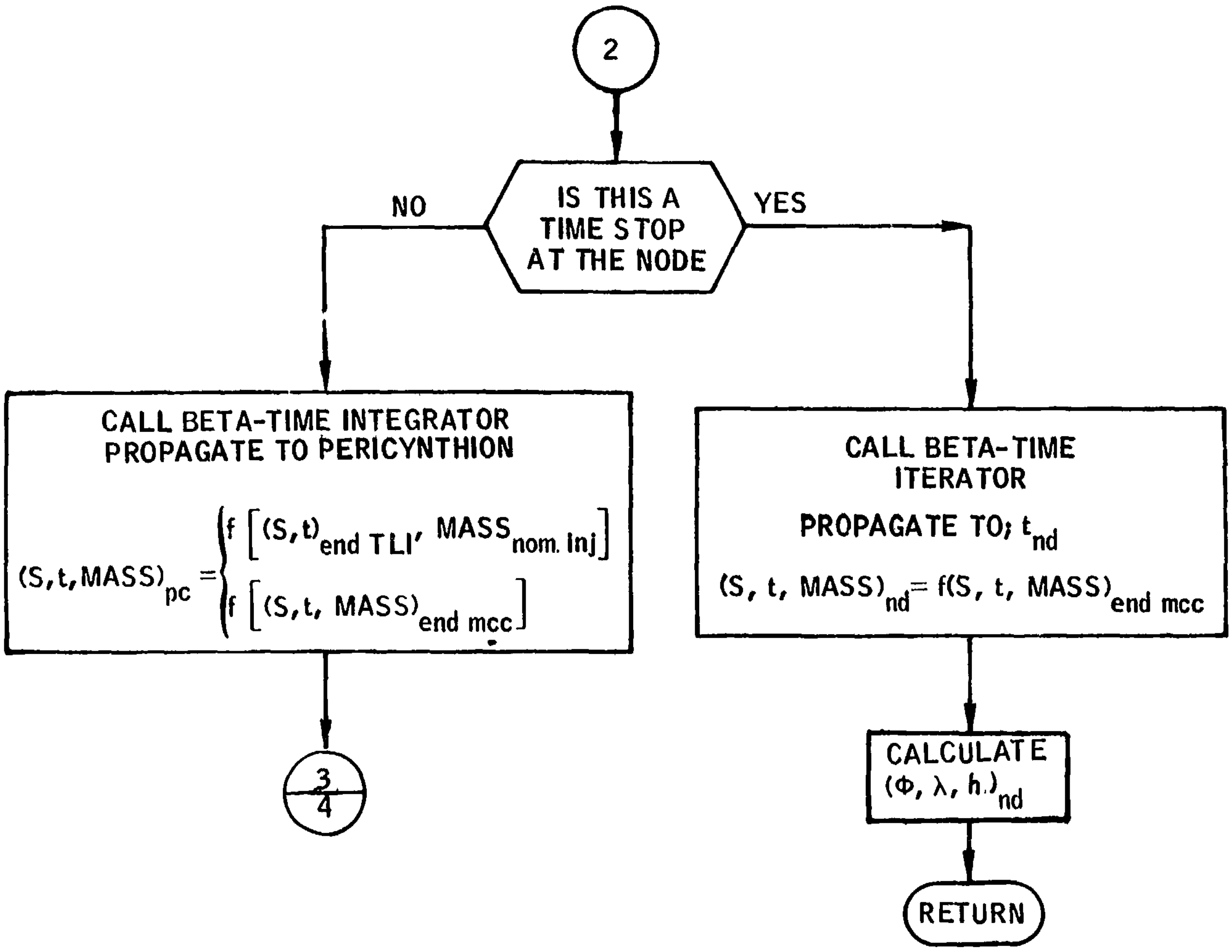


Flowchart 4.- Integrating trajectory computer.

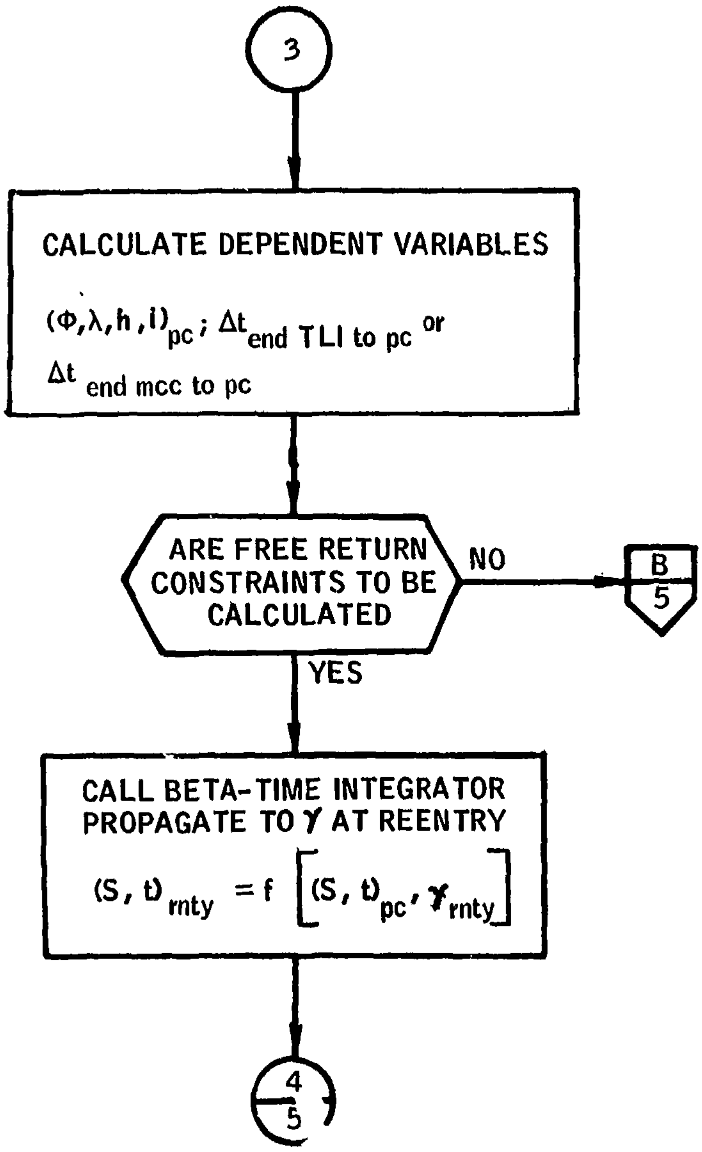


Page 2 of 6

Flowchart 4.- Integrating trajectory computer - Continued.

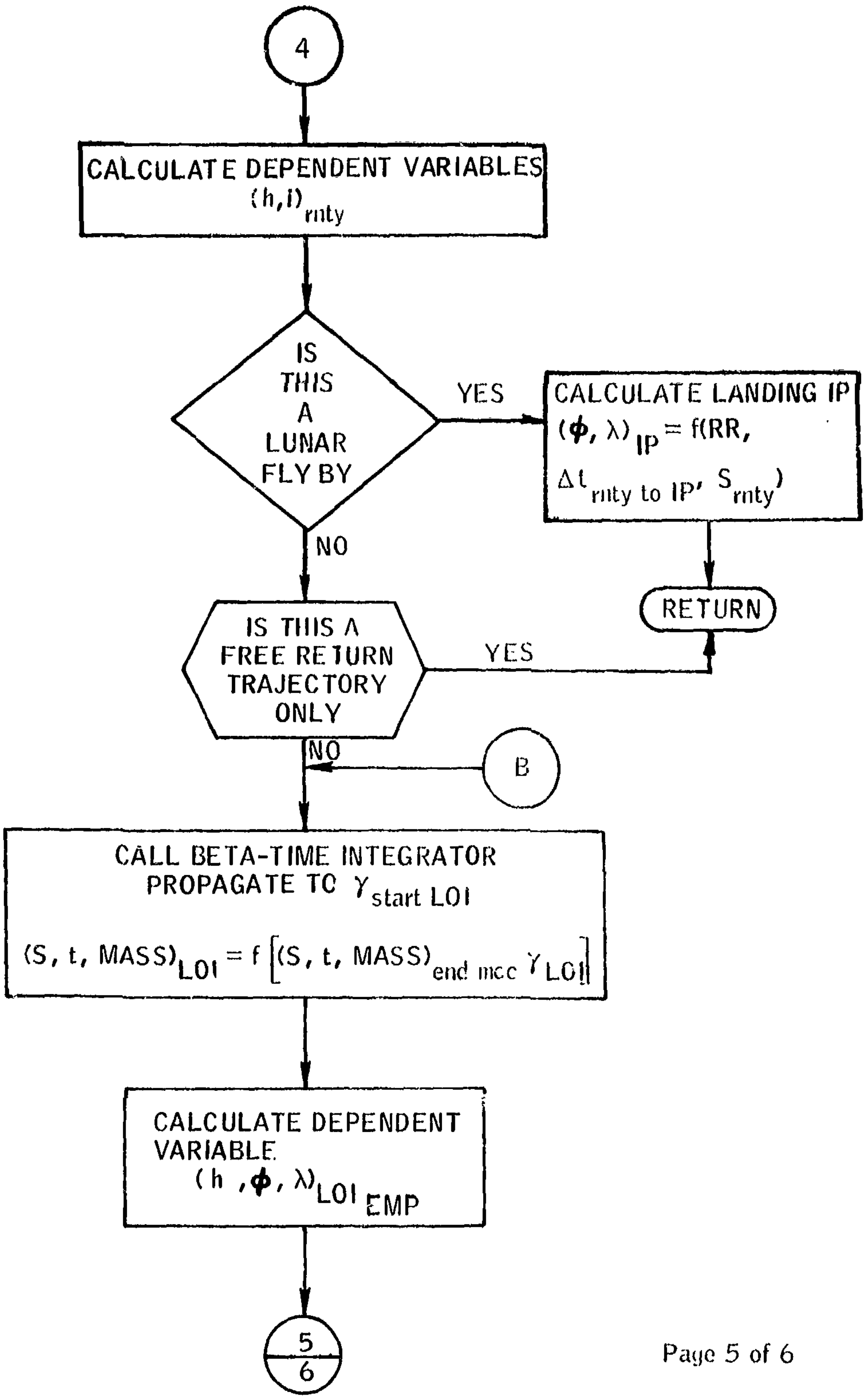


Flowchart 4.- Integrating trajectory computer - Continued.

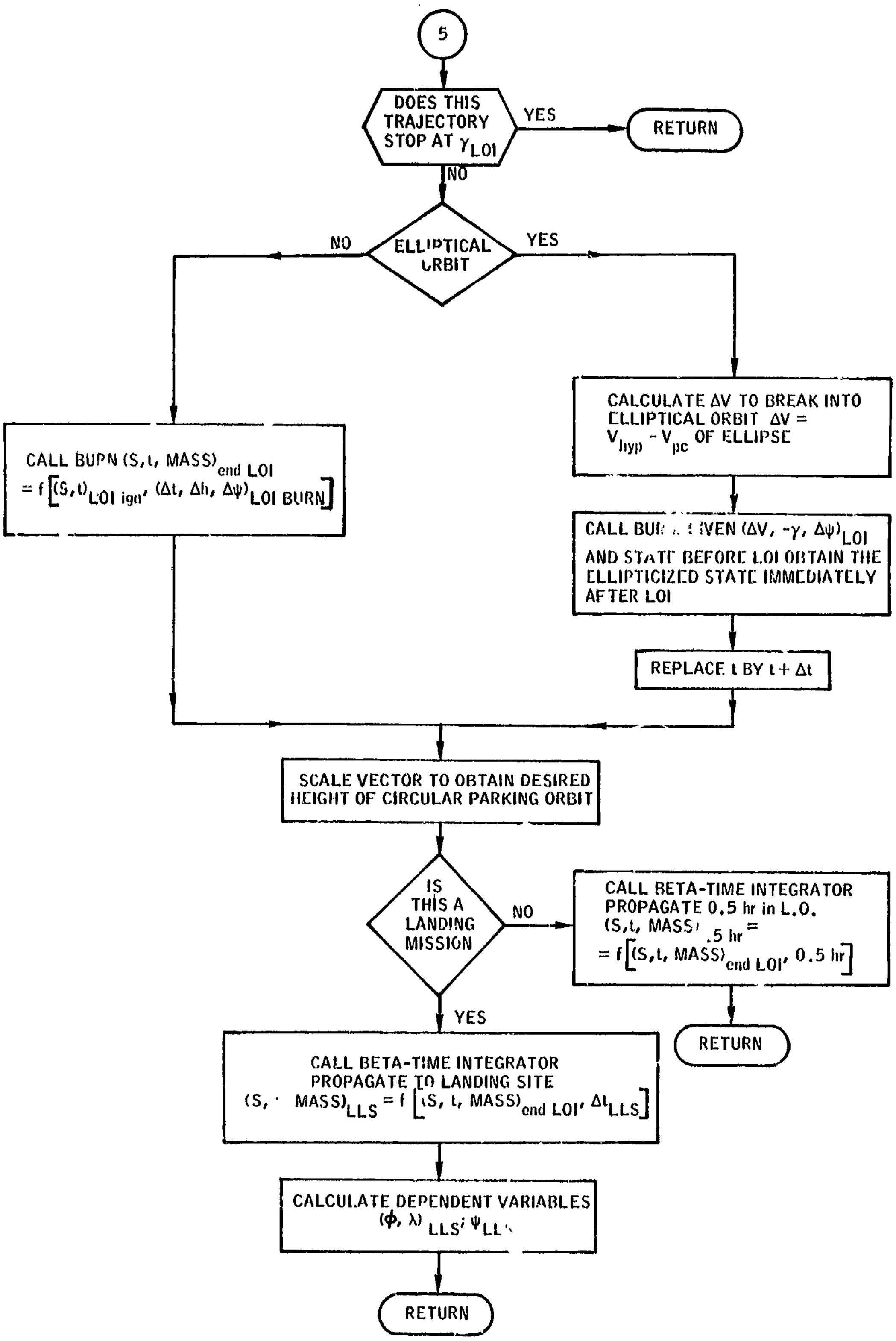


Page 4 of 6

Flow chart 4.- Integrating trajectory computer - Continued.



Flow chart 4.- Integrating trajectory computer - Continued.



Page 6 of 6

Flow chart 4.- Integrating trajectory computer - Concluded.

TABLE I.- INDEPENDENT VARIABLES FOR

THE TRAJECTORY COMPUTERS

Variable	Reference frame	Use		
		Analytic MCC 1st guess	Analytic trajectory computer	Integrating trajectory computer
V_{pc}		✓		
λ_{pc}	EMP	✓		
ψ_{pc}	EMP	✓		
C_3			✓	✓
ΔT_{EPO}			✓	✓
δ			✓	✓
σ			✓	✓
ΔV_{MCC}			✓	✓
$\Delta \gamma_{MCC}$			✓	✓
$\Delta \psi_{MCC}$			✓	✓
$\Delta \psi_{LOI}$			✓	✓
γ_{LOI}			✓	
$\Delta t_{1st\ pass\ LLS}$			✓	✓
$T_{in\ lunar\ orbit}$			✓	
$\Delta \psi_{TEI}$			✓	
ΔV_{TEI}			✓	

TABLE II.- DEPENDENT VARIABLES FOR

THE TRAJECTORY COMPUTERS

Variable	Reference frame	Use		
		Analytic MCC 1st guess	Analytic trajectory computer	Integrating trajectory computer
x_{mcpt}	GC or SC	√		
y_{mcpt}	GC or SC	√		
z_{mcpt}	GC or SC	√		
$MASS_{TLI}$			√	√
$\Delta t_{TL\ Coast}$			√	√
H_{ap}	GC		√	√
H_{pc}	SC		√	√
I_{pc}	EMP		√	√
ϕ_{pc}	EMP		√	√
$H_{fr-rtny}$	GG		√	√
I_{fr}	EEP		√	√
H_{nd}	SG		√	√
ϕ_{nd}	EMP		√	√
λ_{nd}	EMP		√	√
H_{LPO}	SG		√	

TABLE II.- DEPENDENT VARIABLES FOR

THE TRAJECTORY COMPUTERS - Concluded

Variable	Reference frame	Use		
		Analytic MCC 1st guess	Analytic trajectory computer	Integrating trajectory computer
ϕ_{LLS}	SG		✓	✓
λ_{LLS}	SG		✓	✓
ψ_{LLS}	SG		✓	
$MASS_{TEI}$			✓	
$\Delta t_{TE\ Coast}$			✓	

TABLE III.- BASIC MODULES USED IN TRAJECTORY COMPUTERS

MCC first guess trajectory computer	TLI/MCC analytic trajectory computer	TLI/MCC integrated trajectory
EPHM (ephemeris)	DGAMMA	Integrator
RVIO (Cartesian to spherical, etc.)	XBETA (BETA series summation)	Forcing function
PATCH (both ways)	BURN-impulsive	Runge Kutta
EBETA	PATCH (both ways)	Predictor-corrector
RBETA	EBETA	Editor
XBETA (BETA series summation)	XBETA (BETA series summation).	EPHM
EPHM	RBETA	BETA series summation
CTBODY (BETA series summation)	EPHM (ephemeris)	Right ascension of Greenwich
	CTBODY	TLI BURN calibrated
	LIBRAT	LOI BURN calibrated
	EPHM	BURN impulsive
	TLI BURN (calibrated)	LIBRAT
	ELEMT (orbital)	ELEMT (orbital)
	CTBODY (BETA series summation)	RUIO (Cartesian to spherical, etc.)
	EBETA	
	RTASC	
	RVIO (Cartesian to spherical, etc.)	
	SCALE	

REFERENCES

1. Moore, William E.: AS-503/504 Requirements for the RTCC: The Generalized Iterator. NASA IN 66-FM-131, November 4, 1966.

2. Morrey, Bernard F.; and McCaffety, Brody O.: Mission G Requirements for the RTCC: The Translunar Midcourse Correction Processor. MSC IN 68-FM-60, February 1, 1968.

3. McCaffety, Brody O.; and Scheffman, David S.: E/G Requirements for the RTCC: Translunar Injection Processor. MSC IN 67-FM-90, February 1, 1968.

4. Moore, William E.: The Generalized Forward Iterator. MSC IN 66-FM-55, June 15, 1966.

5. Martin, D. T.; and Redwine, W. J.: Empirical Simulation of the Nominal and Alternate Mission Translunar Injection Burn Fit for an Extended Range of Independent Variables. TRW Note No. 67-FMT-506 (A-77), April 14, 1967.

6. Yencharis, J. D.: AS-503A/AS-504A Requirements for the RTCC: Empirical Equations for Simulating the Translunar Injection and Lunar Orbit Insertion Maneuvers. MSC IN 67-FM-96, July 14, 1967.