SPACE TRAJECTORIES
ERROR ANALYSIS PROGRAMS
VERSION II

VOLUME II: PROGRAMMER'S MANUAL

(NASA-CR-132868) SIMULATED TRAJECTORIES
ERROR ANALYSIS PROGRAM, VERSION 2.
VOLUME 2: PROGRAMMER'S MANUAL Final
Report (Martin Marietta Corp.) 623 p HC
$33.25

March 1971
Space Navigation Technology
Martin-Marietta Corporation
Denver, Colorado

Unclas

https://ntrs.nasa.gov/search.jsp?R=19740003544 2020-01-08T01:49:02+00:00Z

https://ntrs.nasa.gov/search.jsp?R=19740003544 2020-01-08T01:49:02+00:00Z
Volume II of Three Volumes
Final Report
Contract NAS 5-11795

Computer Program for Mission Analysis of Lunar and Interplanetary Missions

Prepared by
E. D. Vogt
G. L. Adams
M. M. Working
J. B. Ferguson
M. R. Bynum

Space Navigation Technology Section
Martin Marietta Corporation
Denver, Colorado

For
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

March, 1971
Acknowledgements

The authors wish to acknowledge the invaluable support of several persons whose help was instrumental in the completion of this contract. Dr. Al Bradt, serving as Program Manager, efficiently solved the many administrative problems that arose during the course of the contract. Joanne Spofford did her usual excellent job of typing the voluminous documents. Deborah Bower was responsible for all the graphics work involved with the production of the flowcharts and mathematical analyses. A utility computer program developed by Jim Schnelker aided greatly in the program conversion from CDC single precision to IBM double precision. All of their work was greatly appreciated.
FOREWORD

STEAP II is a series of three computer programs developed by the Martin-Marietta Corporation for the mathematical analysis of the navigation and guidance of lunar and interplanetary trajectories.  STEAP is an acronym for Space Trajectory Error Analysis Programs. The first series of programs under this name was developed under contract NAS 1-8743 for Langley Research Center and was documented in two volumes (STEAP User's Manual, STEAP Analytical Manual) as NASA Contract Report 66818. Under contract NAS 5-11793 the STEAP series was extensively modified and expanded for Goddard Space Flight Center. This second generation series of programs is referred to as STEAP II.

STEAP II is composed of three independent yet related programs: NOMINAL, ERAN, and SIMUL. All three programs require the integration of n-body trajectories for both interplanetary and lunar missions. The virtual mass technique is the scheme used for this purpose in all three programs.

The first program named NOMINAL is responsible for the generation of n-body nominal trajectories (either lunar or interplanetary) performing a number of deterministic guidance events. These events include initial or injection targeting, midcourse retargeting, and orbit insertion. A variety of target parameters are available for the targeting events. The actual targeting is done iteratively either by a modified Newton-Raphson algorithm or by a steepest descent-conjugate gradient scheme. Linear and nonplanar strategies are available for the orbit insertion computation. All maneuvers may be executed either by a simple impulsive model or by a pulsing sequence model.

ERAN, the second program of STEAP II, is used to conduct linear error analysis studies along specific targeted trajectories. The targeted trajectory may however be altered during flight by retargeting events (computed either by linear or nonlinear guidance) and by an orbit insertion event. Knowledge and control covariances are propagated along the trajectory through a series of measurements and guidance events in a totally integrated fashion. The knowledge covariance is processed through measurements using an optimal Kalman-Schmidt filter with arbitrary solve-for/consider augmentation. Execution errors at guidance events may be modeled either by an impulsive approximation or by a pulsing sequence model. The resulting knowledge and control covariances may be analysed by the program at various events to determine statistical data including probabilistic midcourse correction sizing and effectiveness, probability of impact, and biased aimpoint requirements.

The third and final program in the STEAP II series is the simulation program SIMUL. SIMUL is responsible for the testing of the mathematical models used in the navigation and guidance process. An "actual" dynamic model is used to propagate an "actual" trajectory. Noisy measurements from this "actual" trajectory are then sent to the estimation algorithm. Here the actual measurement, the statistics associated with that measurement, and an "assumed" dynamical model are blended together to generate the filter estimate of the trajectory state. This process is repeated continually through the measurement schedule. At guidance events corrections are computed
based on the estimate of the current state. These corrections are then corrupted by execution errors and added to the "actual" trajectory. The statistics and augmentation of the filter, the mismatches in the "actual" and "assumed" dynamics, and the execution errors and measurement biases may then be varied to determine the effects of these parameters on the navigation and guidance process.

The documentation for STEAP II consists of three volumes: the Analytic, Programmer's, and User's Manuals. Each of these documents is self-contained.

The Analytic Manual consists of two major divisions. The first section provides a unified treatment of the mathematical analysis of the STEAP II programs. The general problem description, formulation, and solution are given in a tutorial manner. The second section of this report supplies the detailed analysis of those subroutines of STEAP II dealing with technical tasks.

The Programmer's Manual provides the reader with the information he needs to effectively modify the programs. Both the overall structure of the programs as well as the computational flow and analysis of the individual subroutines is described in this manual.

The User's Manual contains the information necessary to operate the programs. The input and output quantities of the programs are described in detail. Example cases are also given and discussed.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOREWORD</td>
<td>111</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. NOMENCLATURE</td>
<td>2</td>
</tr>
<tr>
<td>3. SUMMARY OF MODES</td>
<td>7</td>
</tr>
<tr>
<td>3.1 The Virtual Mass Propagator VMP</td>
<td>7</td>
</tr>
<tr>
<td>3.2 The Nominal Trajectory Targeter NOMINAL</td>
<td>8</td>
</tr>
<tr>
<td>3.3 The Error Analysis Program ERRAN</td>
<td>9</td>
</tr>
<tr>
<td>3.4 The Simulation Program SIMUL</td>
<td>10</td>
</tr>
<tr>
<td>4. DESCRIPTION OF SUBROUTINES</td>
<td>13</td>
</tr>
<tr>
<td>4.1 Index of Subroutines</td>
<td>13</td>
</tr>
<tr>
<td>4.2 VMP Subroutine Hierarchy</td>
<td>13</td>
</tr>
<tr>
<td>4.3 NOMINAL Subroutine Hierarchy</td>
<td>14</td>
</tr>
<tr>
<td>4.4 ERRAN and SIMUL Hierarchy</td>
<td>14</td>
</tr>
<tr>
<td>5. COMMON VARIABLE DEFINITIONS</td>
<td>25</td>
</tr>
<tr>
<td>5.1 Common Variables by Blocks</td>
<td>26</td>
</tr>
<tr>
<td>5.2 Common Variables in Alphabetical Order</td>
<td>65</td>
</tr>
<tr>
<td>5.3 Large Array Definitions</td>
<td>94</td>
</tr>
<tr>
<td>6. INDIVIDUAL SUBROUTINE DOCUMENTATION</td>
<td>97</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>615</td>
</tr>
</tbody>
</table>
1. **INTRODUCTION**

This Programmer's Manual is intended to supply the reader with sufficient information about the STEAP II programs to enable him to efficiently modify them. Both the overall structure of the programs and the computational flow of the individual subroutines are described in this manual.

This section describes the contents of the Programmer's Manual. Following this discussion the nomenclature used throughout the report is presented.

The third section of this manual describes the four basic components of STEAP II: the n-body trajectory propagation package, the nominal trajectory generator NORMAL, the error analysis program ERRAN, and the simulation program SIMUL. The general purpose and capability of each of the programs is briefly summarized.

Chapter 4 of this volume examines the STEAP II programs from a more detailed viewpoint. The operational structure of each of the main components is described at the subroutine level. The individual subroutines are defined and cross-referenced according to the three main programs of STEAP II.

Chapter 5 contains the definitions of the variables appearing in common blocks throughout the programs. The variables are first listed according to the common blocks to which they belong. The programs requiring each of these common blocks are also noted. Following this all the common variables are listed in alphabetical order with their common blocks referenced. Tables detailing the definitions of large, frequently referenced common arrays are also provided.

Chapter 6 comprises the bulk of this volume. Each of the subroutines is documented in detail in alphabetical order. The purpose of the subroutine is supplied. Subroutines supported or required by the subroutine are listed. Arguments and interval variables of the subroutines are defined and usage of common variables is noted. The mathematical analysis upon which the subroutine is based is then discussed in full. Finally a flow chart of the computational flow of the subroutine is provided.
### 2. NOMENCLATURE

#### A. Arabic symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Semi-major axis of conic</td>
</tr>
<tr>
<td>B·T</td>
<td>Impact plane parameter</td>
</tr>
<tr>
<td>B·R</td>
<td>Impact plane parameter</td>
</tr>
<tr>
<td>Cₓₓₒ</td>
<td>Correlation between position/velocity state and solve-for parameters</td>
</tr>
<tr>
<td>Cₓₓᵤ</td>
<td>Correlation between position/velocity state and dynamic consider parameters</td>
</tr>
<tr>
<td>Cₓₓᵥ</td>
<td>Correlation between position/velocity state and measurement consider parameters</td>
</tr>
<tr>
<td>Cₓₛᵤ</td>
<td>Correlation between solve-for parameters and dynamic consider parameters</td>
</tr>
<tr>
<td>Cₓₛᵥ</td>
<td>Correlation between solve-for parameters and measurement consider parameters</td>
</tr>
<tr>
<td>e</td>
<td>Eccentricity of conic</td>
</tr>
<tr>
<td>E</td>
<td>Eccentric anomaly</td>
</tr>
<tr>
<td>f</td>
<td>True anomaly on conic</td>
</tr>
<tr>
<td>G</td>
<td>Observation matrix relating observables to dynamic consider parameter state</td>
</tr>
<tr>
<td>H</td>
<td>Observation matrix relating observables to position/velocity state</td>
</tr>
<tr>
<td>i</td>
<td>Inclination of conic (reference body equatorial)</td>
</tr>
<tr>
<td>J</td>
<td>Measurement residual covariance matrix</td>
</tr>
<tr>
<td>K</td>
<td>Kalman gain constant for position/velocity state</td>
</tr>
<tr>
<td>L</td>
<td>Observation matrix relating observables to measurement consider parameter state</td>
</tr>
<tr>
<td>M</td>
<td>Mean longitude</td>
</tr>
<tr>
<td>H</td>
<td>Observation matrix relating observables to solve-for parameter state</td>
</tr>
<tr>
<td>Mean anomaly</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$n_1$</td>
<td>Dimension of solve-for parameter state</td>
</tr>
<tr>
<td>$n_2$</td>
<td>Dimension of dynamic consider parameter state</td>
</tr>
<tr>
<td>$n_3$</td>
<td>Dimension of measurement consider parameter state</td>
</tr>
<tr>
<td>$p$</td>
<td>Semilatus rectum of conic</td>
</tr>
<tr>
<td>$P$</td>
<td>Probability density function</td>
</tr>
<tr>
<td>$\hat{P}$</td>
<td>Position/velocity covariance matrix</td>
</tr>
<tr>
<td>$P_s$</td>
<td>Unit vector to periapsis of conic</td>
</tr>
<tr>
<td>$Q$</td>
<td>Solve-for parameter covariance matrix</td>
</tr>
<tr>
<td>$\hat{Q}$</td>
<td>Dynamic noise covariance matrix</td>
</tr>
<tr>
<td>$\hat{\hat{Q}}$</td>
<td>Execution error matrix</td>
</tr>
<tr>
<td>$\hat{\hat{Q}}$</td>
<td>Unit vector in plane of motion normal to $P$</td>
</tr>
<tr>
<td>$r$</td>
<td>Radius</td>
</tr>
<tr>
<td>$r_{CA}$</td>
<td>Radius of closest approach</td>
</tr>
<tr>
<td>$r_{SI}$</td>
<td>Radius of sphere of influence</td>
</tr>
<tr>
<td>$R$</td>
<td>Measurement noise covariance matrix</td>
</tr>
<tr>
<td>$\hat{R}$</td>
<td>Actual noise covariance matrix</td>
</tr>
<tr>
<td>$\hat{\hat{R}}$</td>
<td>Unit vector normal to $T$ in plane perpendicular to approach asymptote directed south ($R = S \times T$)</td>
</tr>
<tr>
<td>$R_c$</td>
<td>Target planet capture radius</td>
</tr>
<tr>
<td>$S$</td>
<td>Kalman gain constant for solve-for parameters</td>
</tr>
<tr>
<td>$S_j$</td>
<td>Velocity correction covariance matrix</td>
</tr>
<tr>
<td>$\hat{S}$</td>
<td>Approach or departure asymptote</td>
</tr>
<tr>
<td>$t_{CA}$</td>
<td>Time of closest approach to target body</td>
</tr>
<tr>
<td>$t_{SI}$</td>
<td>Time of intersection with sphere of influence of target body</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time interval</td>
</tr>
<tr>
<td>$\hat{T}$</td>
<td>Unit vector lying in ecliptic plane normal to $\hat{S}$. ($\hat{T} = \frac{\hat{S} \times \hat{R}}{</td>
</tr>
<tr>
<td>$U_o$</td>
<td>Dynamic consider parameter covariance matrix</td>
</tr>
</tbody>
</table>
Velocity

\( V \)

Measurement consider parameter covariance matrix

\( V_0 \)

Target parameter covariance matrix

\( \hat{W} \)

Unit normal to orbital plane

\( X \)

Actual position/velocity state

\( \bar{X} \)

Targeted nominal position/velocity state

\( \tilde{X} \)

Most recent nominal position/velocity state

B. Greek Symbols

\( \alpha \)

Auxiliary parameters

\( \Gamma \)

Guidance matrix

\( \gamma \)

Illumination path angle

\( \delta \)

Declination of vector

\( \Delta v \)

Velocity increment

\( \epsilon \)

Measurement residual

Errors in target parameters

\( \eta \)

Variation matrix relating position/velocity variations to target conditions

\( \Theta_{xx} \)

State transition matrix partition associated with solve-for parameters

\( \Theta_{xx} \)

State transition matrix partition associated with dynamic consider parameters

\( \Theta \)

Longitude or right ascension

\( A_j \)

Projection of target condition covariance matrix \( W \) into the impact plane

\( \mu \)

Gravitational constant of body

\( \mu \)

Biased aimpoint

\( \nu \)

Sampled measurement noise

True anomaly
Magnitude of gaussian approximation for midcourse correction
Correlation coefficient

\( \rho \)

Standard deviation

\( \sigma \)

Launch azimuth

\( \Sigma \)

Target parameters

\( \Phi \)

Targeting matrix
State transition matrix for position/velocity state
Latitude

\( \chi \)

Sensitivity matrix

\( \psi \)

Matrix relating guidance corrections to target condition variations

\( \Omega \)

Longitude of ascending node

\( \omega \)

Argument of periapsis

\( \tilde{\omega} \)

Longitude of periapsis

C. Subscripts

C  Control variable \((P^C)\)

CA  Closest approach \((r^CA)\)

f  Final variable \((t_f)\)

i  Initial variable \((t_i)\)

j  Index of current guidance event \((P_j)\)

k  Index of current measurement \((P_k)\)

K  Knowledge variable \((P_K)\)

s  Solve-for parameter \((x_s)\)

SI  Sphere of influence \((t_{SI})\)

D. Superscripts

A  Augmented variable \((\Phi^A)\)

T  Matrix transpose \((\Phi^T)\)

-1  Matrix inverse \((\Phi^{-1})\)
Variable immediately before instant \( (P_k^- \text{ or } v^-) \)

Variable immediately after instant \( (P_k^+ \text{ or } v^+) \)

E. Abbreviations

- **AU**: Astronomical unit
- **CA**: Closest approach to reference body
- **ERRA**: Error analysis program
- **FTA**: Fixed time of arrival guidance policy
- **GHA**: Greenwich hour angle
- **J.D.**: Julian date (referenced either 0° or 1900°)
- **km**: Kilometers
- **M/C**: Midcourse correction
- **NOMINAL**: Nominal trajectory generation program
- **POI**: Probability of impact
- **Q-L**: Quasilinear filter event
- **S/C**: Spacecraft
- **SF/C**: Solve-for/consider
- **SIMUL**: Simulation program
- **SOI**: Sphere of influence
- **STM**: State transition matrix
- **STEAP**: Space Trajectories Error Analysis Programs
- **VM**: Virtual Mass
- **2VBP**: Two variable B-plane guidance policy
- **3VBP**: Three variable B-plane guidance policy
3. SUMMARY OF MODES

The Space Trajectory Error Analysis Programs (STEAP) consist of four subprograms or operational modes. The first mode, used as a subroutine by each of the other three programs, is the trajectory mode VMP by which an n-body trajectory (lunar or interplanetary) is propagated by the virtual mass technique. The second mode is the nominal trajectory generator or targeter (NOMNA.L) by which a lunar or interplanetary trajectory meeting specified conditions is determined. The third mode is the error analysis program ERRAN in which the navigation and guidance characteristics of a nominal trajectory are analyzed by linearly propagating knowledge and control covariances along the trajectory. Finally, the simulation mode SIMUL tests the mathematical models used in the navigation and guidance processes by modeling the tracking and correction of an "actual" trajectory. In this chapter a general description of each of these modes will be provided.

3.1 The Virtual Mass Propagator VMP

The dynamic model used by STEAP is supplied by the trajectory propagation package. The only external forces acting upon the spacecraft are assumed to be the gravitational forces of the celestial bodies considered in the integration. Both the spacecraft and the gravitational bodies are assumed to be point masses so neither spacecraft attitude nor planet asphericities are considered.

The celestial bodies to be in the integration are specified by the user and may include the sun, any of the nine planets, and the earth's moon. The motion of the planets about the sun and the moon about the earth are modeled by using mean ecliptic elements of date. If the user desires, each of the planets can be set in a fixed ellipse referenced to some epoch for speedier computation.

The coordinate system used in the integration is also specified by the user. The options available are either heliocentric ecliptic or barycentric ecliptic (nominal for lunar trajectories).

The actual scheme used in the propagation of the trajectory is the virtual mass or varicentric technique (see reference 15). No actual integration is performed by the trajectory mode; the key idea of the virtual mass technique is to build up an n-body trajectory by using a sequence of conic sections around a moving effective force center called the virtual mass. At each instantaneous moment along the trajectory, the combined effects of all the gravitational bodies can be viewed as resulting from a fictitious body of unique magnitude and position which is called the virtual mass. The computational pro-
procedure then assumes that over a small time interval the motion of the spacecraft can be represented by a two-body conic section arc relative to this virtual mass. The complete trajectory is thus generated by a series of small arcs pieced together in steps while updating the position and magnitude of the effective force center. The main advantage of the virtual mass technique is that the tedious numerical integration of the differential equation is avoided.

Another significant feature of the virtual mass technique is its flexibility. By varying a simple parameter called the "accuracy level" related to the true anomaly increment of each step, trajectories ranging from a sequence of relatively few conic section arcs corresponding to a very approximate solution to those requiring a large number of arcs corresponding to highly accurate solutions may be generated.

1.2 The Nominal Trajectory Targeter NOMINAL

NOMINAL is responsible for the generation of a nominal trajectory for either lunar or interplanetary missions. The method of propagation in either case is the virtual mass n-body integrator. The trajectory may be processed through a series of deterministic maneuvers including initial or injection targeting, subsequent retargeting, and finally orbit insertion. A variety of target parameters are available for the targeting events. Both coplanar and noncoplanar strategies are permitted in the orbit insertion maneuver.

If an initial state for the problem is known, this may be read in to start the trajectory. Otherwise NOMINAL generates its own zero iterate. In interplanetary missions this involves solving the Lambert time of flight equation for the massless planet trajectory that connects the desired initial and final positions in the specified time interval. Four options are available in describing these reference points:

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>Final Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch Planet</td>
<td>Target Planet</td>
</tr>
<tr>
<td>Launch Planet</td>
<td>Specified Point</td>
</tr>
<tr>
<td>Specified Point</td>
<td>Target Planet</td>
</tr>
<tr>
<td>Specified Point</td>
<td>Specified Point</td>
</tr>
</tbody>
</table>

If the initial point is referenced to the launch planet, a launch profile is consulted to generate a realistic set of injection condition consistent with the heliocentric trajectory.

For lunar trajectories a slightly different procedure is used. The required data for the lunar zero iterate includes specification of the desired semimajor axis with respect to the moon, radius and time of
closest approach to the moon, and inclination to the lunar equator. Then the generation of the zero iterate is accomplished by first targeting a patched conic trajectory and then a multi-conic trajectory to the desired conditions.

A targeting event may be processed immediately after obtaining a zero iterate state or at any point along the nominal trajectory. At a targeting event the current velocity is refined to yield a trajectory satisfying target parameter constraints. The possible target parameters are:

1. TRF
2. TSH
3. TCS
4. TCA
5. B-T
6. B-R
7. ECA
8. INC
9. SMA (Lunar)
10. XP
11. YF
12. ZF

The targeting method to be used is specified by the user. Either a modified Newton-Raphson algorithm or a steepest descent/conjugate gradient technique may be used.

Orbit insertion events are also available in NOMINAL. At a specified time the spacecraft state relative to the target body is computed. The resulting conic trajectory relative to the target body is then compared with the desired orbit to determine the optimal time to make the insertion and the required correction. At the proper time the velocity correction is then implemented. Two strategies are permitted in the orbit insertion computation:

Coplanar - The desired semimajor axis, eccentricity, and periapsis shift of a coplanar orbit are specified.

Nonplanar - The desired plane of the post-insertion state is specified along with nominal values of the orbit elements.

The targeted correction, orbit insertion correction, or an externally supplied correction may be executed if desired. Two models are available for this implementation; a simple impulsive addition or a more complex multiple pulse model.

The program will integrate and record the periodically-corrected nominal trajectory until reaching a termination time specified by the user.

1.3 The Error Analysis Program ERRAN

The error analysis program ERRAN is a preflight mission analysis tool used primarily to propagate covariance matrices along selected
The guidance event is the most complex event and yields much useful information for preflight mission analysis. Several types of guidance events are available in ERRAN. At a midcourse guidance event the user can choose from three midcourse guidance policies. The midcourse guidance event can also be constrained to satisfy planetary quarantine requirements. At an orbital insertion guidance event the user can choose from two insertion policies. Options are also available for changing target conditions in mid-flight and re-targeting the trajectory using nonlinear techniques, or for simply applying an externally-supplied or precomputed ΔW at some arbitrary trajectory time. Two thrust models are available: impulse and impulse series. Execution error statistics are generated using an error model defined by a proportionality error, a resolution error, and two pointing angle errors. At a midcourse guidance event in ERRAN we also compute a statistical ΔW and the target condition covariance matrix both before and after the midcourse correction.

3.4 The Simulation Program SIMUL

The simulation program SIMUL is the most complex program in the STEAP set of programs. In SIMUL the validity of the navigation and guidance process is examined by simulating an actual mission. Spacecraft state estimates are generated in SIMUL, as well as covariance matrices. The results given by the error analysis program ERRAN become meaningful only when SIMUL shows that the estimated spacecraft trajectory converges, within reasonable bounds specified by the covariance matrix, to the simulated actual trajectory.

All state transition matrix, parameter augmentation, and measurement options described in section 3.3 are also available in SIMUL. As in ERRAN, the computational procedure in SIMUL is divided into basic cycle computations and event computations. The SIMUL basic cycle is concerned with the generation of state estimates and an actual trajectory, together with all quantities generated in the ERRAN basic cycle. Eigenvector and prediction events in SIMUL involve all computations performed in the corresponding ERRAN events. In addition, the SIMUL prediction event propagates state estimates forward to the time to which we are predicting.

All options available in the ERRAN guidance event (see section 3.3) are also available in the SIMUL guidance event. The treatment of the midcourse guidance event, however, is different in several respects. First, since an estimated spacecraft state is generated in SIMUL, an actual midcourse ΔW can be computed, rather than a statistical ΔW as in ERRAN. Also, all linear midcourse ΔW's computed in SIMUL can be recomputed using nonlinear techniques. Finally, since an actual trajectory is generated in SIMUL, actual target errors after the midcourse correction are also computed.
interplanetary or lunar trajectories. Three main quantitative results are available from ERRAN: (a) knowledge covariances, which provide a measure of how well the actual trajectory is known after each measurement is processed; (b) control covariances, which when propagated forward to the target provide a measure of how well the nominal target conditions will be satisfied by the actual trajectory; and (c) statistical midcourse ΔV's.

State transition matrices are required to propagate covariance matrices over an arbitrary interval of time. Three methods are available for computing the 6 x 6 position/velocity state transition matrix. The first two methods, which are analytical methods, are analytical patched conic and analytical virtual mass. The third method uses numerical differencing to compute the state transition matrix. To increase the accuracy of the analytical techniques over long time intervals a state transition matrix cascading option is also available. Augmented parameter state transition matrices are always computed using numerical differencing.

Measurements are processed in an optimal recursive consider filter. Up to 23 dynamic and measurement parameters may be solved-for or considered. The dynamic parameters include biases in gravitational constants of the Sun and the target planet and biases in the 6 orbital elements of the target planet. Measurement biases include biases in the locations of the 3 earth-based tracking stations, and biases in all measurements. Available measurement types are range, range-rate, star-planet angles, and apparent planet diameter measurements. Measurement noise for each measurement type is assumed to be constant.

The computational procedure in ERRAN is divided into basic cycle computations and event computations. Basic cycle computations are concerned with the propagation of covariances forward to a measurement time and processing the measurement. Events refer to a set of specialized computation, not directly concerned with measurement processing, which can be scheduled to occur at arbitrary times along the trajectory.

The three events available in ERRAN are eigenvector events, prediction events, and guidance events. At an eigenvector event the position and velocity partitions of the knowledge covariance matrix are diagonalized to reveal geometric information about the size and orientation of the position and velocity navigation uncertainties. Associated hyperellipsoids are also computed. At a prediction event the most recent covariance matrix is propagated forward to some critical trajectory time to determine predicted navigation uncertainties in the absence of further measurements.
A quasi-linear filtering event, not defined in ERBAN, is also available in SIMUL. At a quasi-linear filtering event the most recent nominal trajectory is updated by using the most recent state estimate. This permits more accurate computation of state transition and observation matrices which in turn helps prevent the occurrence of divergence of the state estimate.
4. DESCRIPTION OF SUBROUTINES

4.1 Index of Subroutines

The subroutines making up the STEAP programs are listed according to category in Table 4.1 following. The programs are divided into three general classes: the subroutines making up the virtual mass propagation package used by the three basic programs, the additional subroutines required by NOMINAL and then the additional subroutines used in ERRAN and SIMUL. In Table 4.2 the subroutines are listed again by category with a brief summary of their purpose. Thus Table 4.2 can be used to track down the subroutine in which a specific task is performed. The individual subroutines are then documented in detail in alphabetical order in Chapter 6.

4.2 VMP Subroutine Hierarchy

The executive program for the virtual mass n-body trajectory propagator is named VMP. The reader should investigate the detailed analysis and flow chart of VMP in the individual subroutine documentation in Chapter 6. The summaries of the subroutines of VMP are given in the first part of Table 4.2. The subroutines are conveniently divided into four general classes:

- Conic: Subroutines based on conic approximations
- Ephemeris: Subroutines used to compute the positions and velocities of the gravitational bodies at different times along the trajectory
- Propagation: Subroutines used in the direct computation of the trajectory of the spacecraft moving under the influence of all the gravitational bodies
- Input/Output: Subroutines processing either the input or output from the virtual mass trajectory propagation

The calling hierarchy of the virtual mass programs is given in Figure 4.1. All subroutines within a given block are at an identical level relative to the calling hierarchy unless they are enclosed by parentheses. Subroutines within parentheses are called by the preceding subroutine. Otherwise calls to subroutines are indicated by arrows. Thus all subroutines within blocks connected directly to VMP are called directly from VMP.
4.3 NOMCOL Subroutine Hierarchy

The first of the three independent programs of STEAP is the nominal trajectory targeter NOMCOL. The main program controlling the processing of the program goes under the same name. Reference is made to the complete documentation of NOMCOL in Chapter 6. The subroutine hierarchy of NOMCOL is provided in Figure 4.2. BLOCK DATA loads the planetary constants used by many of the subroutines; it is therefore available to all subroutines of NOMCOL. PRELIM reads the input data and calls ZERIT for the computation of a zero iterate if necessary. ZERIT in turn calls HELIO or LUNA for the actual computation of the interplanetary or lunar zero iterate respectively. NOMCOL calls TRJTRY for the propagation of the nominal trajectory between guidance maneuvers. TRJTRY of course calls the VMP package described in Figure 4.1. NOMCOL calls GIDANS for the actual processing of any guidance event. GIDANS calls VMP to initialize arrays for the other events. If a targeting event requires a zero iterate computation ZERIT is called. Subroutine TARGET controls the targeting events; INSERS controls the insertion decision computations. NOMCOL calls EXECUTE for the execution of either type event.

4.4 ERRAN and SIMUL Hierarchy

The calling hierarchy of the subroutines used in ERRAN and SIMUL is shown in Figures 4.3 and 4.4, respectively. The similar structure of ERRAN and SIMUL is apparent from these two figures. All subroutines can be classified under one or more of the following categories: input, output, basic cycle (measurement processing), or events. The calling hierarchy of the subroutines is indicated by the level of the subroutine in figures 4.3 and 4.4. A given subroutine calls all those subroutines which are directly connected to the subroutine and are located on the next lower level. For the purposes of clarity, the lowest level subroutine on a given branch is enclosed in parentheses. BLOCK DATA is shown connected to the main hierarchy with a dashed line to indicate that the constants stored in BLOCK DATA are available to all subroutines.

The complete documentation of all subroutines used in ERRAN and SIMUL is given in Chapter 5 of this document.
SPACE TRAJECTORIES
ERROR ANALYSIS PROGRAMS
VERSION II

VOLUME II: PROGRAMMER'S MANUAL

March 1971
Space Navigation Technology
Martin-Marietta Corporation
Denver, Colorado
Table 4.1 STEAP II Subroutines

I. Virtual Mass Subroutines

<table>
<thead>
<tr>
<th>A. Conic</th>
<th>B. Ephemeris</th>
<th>C. Propagation</th>
<th>D. Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CAREL</td>
<td>1. TIME</td>
<td>1. VMP</td>
<td>1. TRAPAR</td>
</tr>
<tr>
<td>2. ELCAR</td>
<td>2. BLOCK DATA</td>
<td>2. ESTMT</td>
<td>2. INPUTZ</td>
</tr>
<tr>
<td>3. IMPACT</td>
<td>3. ORB</td>
<td>3. VECTOR</td>
<td>3. PRINT</td>
</tr>
<tr>
<td></td>
<td>4. EPHEM</td>
<td>4. VMASS</td>
<td>4. SPACE</td>
</tr>
<tr>
<td></td>
<td>5. CENTER</td>
<td></td>
<td>5. NEWPGE</td>
</tr>
<tr>
<td></td>
<td>6. PECEQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. EULMX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

II. NOMINAL Subroutines

<table>
<thead>
<tr>
<th>A. Executive</th>
<th>B. Zero iterate</th>
<th>C. Targeting</th>
<th>D. Insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. NOMINAL</td>
<td>1. ZERIT</td>
<td>1. TARGET</td>
<td>1. INSERTS</td>
</tr>
<tr>
<td>2. PRELIM</td>
<td>2. HELIO</td>
<td>2. TAROPT</td>
<td>2. COPINS</td>
</tr>
<tr>
<td>3. TRJRY</td>
<td>3. LAUNCH</td>
<td>3. TARMAX</td>
<td>3. NONINS</td>
</tr>
<tr>
<td>4. GIDANS</td>
<td>4. FLITE</td>
<td>4. DESERT</td>
<td>4. (BATCON)</td>
</tr>
<tr>
<td>5. EXECUTE</td>
<td>5. SERIE</td>
<td>5. MATIN</td>
<td>5.</td>
</tr>
<tr>
<td></td>
<td>6. LUNA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

III. ERFAN and SIMUL Subroutines

<table>
<thead>
<tr>
<th>A. Executive</th>
<th>C. Navigation</th>
<th>D. Event</th>
<th>E. Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ERFAN</td>
<td>1. NAVN</td>
<td>1. SETEVN</td>
<td>1. DATA</td>
</tr>
<tr>
<td>2. SIMUL</td>
<td>2. SCHED</td>
<td>2. DETEVs</td>
<td>2. DATA1</td>
</tr>
<tr>
<td></td>
<td>3. TRAKM</td>
<td>3. FRED</td>
<td>3. DATAS</td>
</tr>
<tr>
<td></td>
<td>4. TRAKS</td>
<td>4. PRSIT</td>
<td>4. DATA1S</td>
</tr>
<tr>
<td></td>
<td>5. TARPRL</td>
<td>5. QUASI</td>
<td>5. CONURT</td>
</tr>
<tr>
<td></td>
<td>6. STAPRL</td>
<td>6. GUID</td>
<td>6. TRANS</td>
</tr>
<tr>
<td></td>
<td>7. MENO</td>
<td>7. GUIS</td>
<td>7. CORREL</td>
</tr>
<tr>
<td></td>
<td>8. MENOS</td>
<td>8. GUID</td>
<td>8. STMPR</td>
</tr>
<tr>
<td></td>
<td>9. BIAS</td>
<td>9. GUIS</td>
<td>9. SUB1</td>
</tr>
<tr>
<td>10. MUND</td>
<td>10. RNUM</td>
<td>10. VARADA</td>
<td>10. TITLE</td>
</tr>
<tr>
<td>11. PCTM</td>
<td>11. DYNOS</td>
<td>11. VBAS</td>
<td>11. PRINT3</td>
</tr>
<tr>
<td>12. CONCZ</td>
<td>12. PARTL</td>
<td>12. PRNTS</td>
<td>12. PRINT4</td>
</tr>
<tr>
<td>13. CASCAD</td>
<td>13. BLA1M</td>
<td>13. PRNTS3</td>
<td>13. PRNTS4</td>
</tr>
<tr>
<td>14. JACOBI</td>
<td>14. POICOM</td>
<td>14. PRNTS4</td>
<td></td>
</tr>
<tr>
<td>15. HYEL</td>
<td>15. QCOMPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. LIGH</td>
<td>16. NOWLIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>17. PULCOV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>18. EXECUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19. EXECS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.1 Subroutine Hierarchy of VMP
1. **BLOCK DATA** stores constants available to all subroutines.

2. Subroutine hierarchy available in Table 4.1.

3. Subroutine hierarchy available on far left.

**Figure 4.2** Subroutine Hierarchy of NOMINAL.
Superscript notation:
1. Subroutine hierarchy continued elsewhere in Figure 4.3.
2. Subroutine hierarchy continued in Figure 4.1.
3. Subroutine hierarchy continued in Figure 4.2.

Figure 4.3a Subroutine Hierarchy of ERRAN
Figure 4.33 Subroutine Hierarchy of PERAN (cont. on)
Figure 4.4a Subroutine Hierarchy of SIMUL.
Figure 4.4b Subroutine Hierarchy of SIMUL (continued)
Table 4.2 STEAP II Subroutine Summaries

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Virtual Mass Subroutines</td>
<td></td>
</tr>
<tr>
<td>A. Conic</td>
<td></td>
</tr>
<tr>
<td>1. CAREL</td>
<td>Convert Cartesian state to conic elements</td>
</tr>
<tr>
<td>2. ELCAK</td>
<td>Convert conic elements to Cartesian state</td>
</tr>
<tr>
<td>3. IMPACT</td>
<td>Compute impact plane parameters</td>
</tr>
<tr>
<td>B. Ephemeris</td>
<td></td>
</tr>
<tr>
<td>1. TIME</td>
<td>Convert J.D. to calendar date or vice versa</td>
</tr>
<tr>
<td>2. BLOCK DATA</td>
<td>Set gravitational body ephemeris constants</td>
</tr>
<tr>
<td>3. ORB</td>
<td>Compute orbital elements of gravitational body at given time</td>
</tr>
<tr>
<td>4. EPHM</td>
<td>Compute inertial state of gravitational body at given time</td>
</tr>
<tr>
<td>5. CENTER</td>
<td>Convert state of bodies to barycentric coordinates</td>
</tr>
<tr>
<td>6. FCEQ</td>
<td>Compute transformation matrix from ecliptic to equatorial coordinates</td>
</tr>
<tr>
<td>7. EUAUX</td>
<td>Compute rotation matrix</td>
</tr>
<tr>
<td>C. Propagation</td>
<td></td>
</tr>
<tr>
<td>1. VMP</td>
<td>Executive routine for virtual mass trajectory propagation</td>
</tr>
<tr>
<td>2. ESTMT</td>
<td>Determine final position and magnitude of VM on current step</td>
</tr>
<tr>
<td>3. VECTOR</td>
<td>Compute the spacecraft final position on current step</td>
</tr>
<tr>
<td>4. VMASS</td>
<td>Determine VM data for current step</td>
</tr>
<tr>
<td>D. Input/Output</td>
<td></td>
</tr>
<tr>
<td>1. TRAPAR</td>
<td>Compute and record navigation parameter data</td>
</tr>
<tr>
<td>2. INPUTZ</td>
<td>Convert input data into VMP compatible form</td>
</tr>
<tr>
<td>3. PRINT</td>
<td>Print output of VM trajectory</td>
</tr>
<tr>
<td>4. SPACE</td>
<td>Space paper for output purposes</td>
</tr>
<tr>
<td>5. NEWPGE</td>
<td>Record headings for each new page in VM printout</td>
</tr>
<tr>
<td>II. NORMAL Subroutines</td>
<td></td>
</tr>
<tr>
<td>A. Executive</td>
<td></td>
</tr>
<tr>
<td>1. NORMAL</td>
<td>Control nominal trajectory generation (Main program)</td>
</tr>
<tr>
<td>2. PRELIM</td>
<td>Perform preliminary work for NORMAL</td>
</tr>
<tr>
<td>3. TILJXY</td>
<td>Propagate virtual mass trajectory to next event</td>
</tr>
<tr>
<td>4. GIDANS</td>
<td>Control computation of trajectory correction</td>
</tr>
<tr>
<td>5. EXECUTE</td>
<td>Control execution of trajectory correction</td>
</tr>
<tr>
<td>B. Zero Iterate</td>
<td></td>
</tr>
<tr>
<td>1. ZERIT</td>
<td>Control computation of zero iterate</td>
</tr>
<tr>
<td>2. HELIO</td>
<td>Compute heliocentric phase of interplanetary zero iterate</td>
</tr>
<tr>
<td>3. LAUNCH</td>
<td>Compute launch phase of interplanetary zero iterate</td>
</tr>
<tr>
<td>4. FLITE</td>
<td>Lambert time of flight equation solver</td>
</tr>
<tr>
<td>5. SREK</td>
<td>Batten generalized function solver</td>
</tr>
<tr>
<td>6. LUNA</td>
<td>Control lunar zero iterate generation</td>
</tr>
<tr>
<td>7. LUNCON</td>
<td>Generate patched conic lunar trajectory</td>
</tr>
<tr>
<td>8. LUNAR</td>
<td>Control patched conic targeting</td>
</tr>
<tr>
<td>9. KULCON</td>
<td>Generate lunar multi-conic trajectory</td>
</tr>
</tbody>
</table>
Table 4.2 STEAP II Subroutine Summaries (Cont'd)

10. MULTAR  Control multi-conic targeting
11. BATCON  Propagate conic trajectory

C. Targeting
1. TARGET  Control n-body targeting
2. TAROPT  Set up target parameter arrays
3. TARKIN  Compute Newton-Raphson targeting matrix
4. DESENT  Compute steepest descent-conjugate gradient corrections

5. MATIN  Compute matrix inverse

D. Insertion
1. INSERS  Control orbit insertion computations
2. COPINS  Compute coplanar orbit insertion
3. NONINS  Compute nonplanar orbit insertion

E. Pulsing Arc
1. PREPUL  Perform preliminary work for multiple pulses
2. PULSEX  Execute pulsing arc
3. PERHEL  Propagate perturbed heliocentric conic
4. (BATCON) Propagate conic trajectory

III. ERRAN and SIMUL Subroutines
A. Executive
1. ERRAN  Control error analysis program (Main program)
2. SIMUL  Control simulation program (Main program)

B. Dynamic Model
1. NTM  Control generation of trajectory data for ERRAN
2. NTNS  Control generation of trajectory data for SIMUL
3. PSIM  Control computation of state transition matrix (STM)

4. NDTM  Compute unaugmented partition of STM by numerical differencing
5. PLED  Compute STM partition associated with ephemeris biases
6. MUND  Compute STM partition associated with gravitational constants
7. FCTM  Compute unaugmented partition of STM by patched conic technique
8. CONC2  Compute unaugmented partition of STM by virtual mass technique
9. CASCAD  Compute unaugmented partition of STM by cascaded Darby matrices

C. Navigation
1. NAVM  Propagate covariance matrices between measurements and between events
2. SCHED  Select next measurement time from measurement schedule
3. TRACK  Compute observation matrices
4. TRAKS  Compute observation matrices and actual measurements
5. TARPRL  Compute target planet position partials
6. STAPRL  Compute station location position and velocity partials
7. MEKO  Compute assumed measurement noise covariance matrix
### Table 4.2 STRAP II Subroutine Summary (Cont'd)

<table>
<thead>
<tr>
<th>8.</th>
<th>KEROB</th>
<th>Compute assumed and actual measurement noise covariance matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>BIAS</td>
<td>Compute actual measurement bias</td>
</tr>
<tr>
<td>10.</td>
<td>ENUM</td>
<td>Generate random numbers</td>
</tr>
<tr>
<td>11.</td>
<td>DYN0</td>
<td>Compute dynamic noise covariance matrix</td>
</tr>
<tr>
<td>12.</td>
<td>DYN0B</td>
<td>Compute dynamic noise covariance matrix and actual dynamic noise</td>
</tr>
<tr>
<td>13.</td>
<td>GHA</td>
<td>Compute Greenwich hour angle</td>
</tr>
<tr>
<td>14.</td>
<td>JACOB</td>
<td>Compute eigenvalues and eigenvectors of a matrix</td>
</tr>
<tr>
<td>15.</td>
<td>HYELS</td>
<td>Compute hyperellipsoids</td>
</tr>
<tr>
<td>16.</td>
<td>EIGHY</td>
<td>Control computation of eigenvalues, eigenvectors, and hyperellipsoids</td>
</tr>
</tbody>
</table>

### D. Event

| 1.  | SETEVA | Perform computations common to most events in ERRAN            |
| 2.  | SETEVS | Perform computations common to most events in SIMUL            |
| 3.  | PRED  | Perform prediction event in ERRAN                              |
| 4.  | PREDM | Perform prediction event in SIMUL                              |
| 5.  | QUASI | Perform quasi-linear filtering event in SIMUL                  |
| 6.  | GUID  | Perform guidance event in ERRAN                               |
| 7.  | GUIDM | Perform guidance event in SIMUL                                |
| 8.  | GUID  | Compute guidance and variation matrices in ERRAN               |
| 9.  | GUIDS | Compute guidance and variation matrices in SIMUL               |
| 10. | VARADA| Compute 3VBP variation matrix in ERRAN                         |
| 11. | VARSIM| Compute 3VBP variation matrix in SIMUL                         |
| 12. | PARTL | Compute partials of $B^T$, $B^R$ wrt state                    |
| 13. | BLA1M | Perform biased aimpoint guidance                               |
| 14. | POICOM| Compute probability of impact                                  |
| 15. | QCOMP | Compute execution error covariance matrix                      |
| 16. | NONLIN| Control execution of nonlinear guidance events                 |
| 17. | PULCOV| Propagate covariance matrix across a series of pulses          |
| 18. | EXECUT| Control execution of pulsing arc in ERRAN                     |
| 19. | EXECUT| Control execution of pulsing arc in SIMUL                     |

### E. Input/Output

| 1.  | DATA  | Perform preliminary computations and read data in ERRAN        |
| 2.  | DATA1 | Continuation of DATA                                           |
| 3.  | DATAS | Perform preliminary computations and read data in SIMUL        |
| 4.  | DATA2 | Continuation of DATAS                                          |
| 5.  | CONVET| Convert JPL injection conditions to Cartesian components       |
| 6.  | TRANS | Compute coordinate transformations                              |
| 7.  | COREL | Compute and print correlation matrix partitions and standard deviations |
| 8.  | SIMPR | Print STIM partitions                                         |
| 9.  | SUB1 | Compute position and velocity magnitudes                       |
| 10. | TITLE | Print titles                                                   |
| 11. | PRINT3| Print basic cycle data in ERRAN                               |
| 12. | PRINT3| Print ERRAN summary                                           |
| 13. | PRINT4| Print basic cycle data in SIMUL                               |
| 14. | PRINT4| Print SIMUL summary                                           |
5. COMMON VARIABLE DEFINITIONS

THE BULK OF THE VARIABLES USED IN THE STEAP PROGRAMS ARE COMMON VARIABLES. THESE VARIABLES ARE DEFINED IN DETAIL IN THIS CHAPTER. THE FIRST SECTION LISTS THE COMMON BLOCKS IN ALPHABETICAL ORDER. THE PROGRAMS (NORMAL, ERRAN, SIMU) USING EACH COMMON BLOCK ARE NOTED. THE VARIABLES OF EACH COMMON BLOCK ARE DEFINED IN THE ORDER THAT THEY APPEAR IN THE COMMON BLOCK.

THE SECOND SECTION LISTS ALPHABETICALLY ALL VARIABLES APPEARING ANYWHERE IN COMMON. THE COMMON BLOCK TO WHICH THE VARIABLE BELONGS IS REFERENCED. THE DEFINITION OF THE VARIABLE IS THEN GIVEN.

THE THIRD SECTION SUPPLIES THE DEFINITIONS OF SEVERAL LARGE FREQUENTLY REFERENCED ARRAYS. THE ELEMENT APPEARING IN EACH COMPONENT OF EACH ARRAY IS NOTED.
### 5.1 Common Variables by Blocks

In this section, common blocks appearing in STEAP are listed in alphabetical order. Variables within these blocks are listed and defined in the order they appear in the program.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATRANS(6)</td>
<td>Closest approach state</td>
</tr>
<tr>
<td>TMPR(3)</td>
<td>Most recent target state</td>
</tr>
<tr>
<td>TNONC(7)</td>
<td>Nominal closest approach target state, incl. time</td>
</tr>
<tr>
<td>TNONMB(3)</td>
<td>Nominal b-plane target state</td>
</tr>
<tr>
<td>PHIZ(3,3)</td>
<td>Inverse of variation matrix partition</td>
</tr>
<tr>
<td>VINF</td>
<td>Hyperbolic excess velocity</td>
</tr>
<tr>
<td>TINJ</td>
<td>Injection time</td>
</tr>
<tr>
<td>PROBI</td>
<td>Allowable probability of impact</td>
</tr>
<tr>
<td>ADA(3,6)</td>
<td>Variation matrix</td>
</tr>
<tr>
<td>T3(10)</td>
<td>Array of guidance event times</td>
</tr>
<tr>
<td>IBAG</td>
<td>Not used</td>
</tr>
<tr>
<td>IPQ</td>
<td>Not used</td>
</tr>
<tr>
<td>IGUID(5,10)</td>
<td>Array of guidance event codes</td>
</tr>
<tr>
<td>II</td>
<td>Guidance event counter</td>
</tr>
</tbody>
</table>

/BAIM / MODE ERRAN, SIMUL
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Trajectory time in days</td>
</tr>
<tr>
<td>PMASS(11)</td>
<td>Gravitational constants of planets in A.U.<strong>3/DAY</strong>2</td>
</tr>
<tr>
<td>CN(60)</td>
<td>Constants used to calculate the orbital elements of the first five planets</td>
</tr>
<tr>
<td>ST(50)</td>
<td>Constants used to calculate the orbital elements of the last four planets</td>
</tr>
<tr>
<td>EMN(15)</td>
<td>The constants used to calculate the orbital elements of the moon</td>
</tr>
<tr>
<td>SMJR(18)</td>
<td>Constants used to calculate the semi-major axes of the planets</td>
</tr>
<tr>
<td>RADIUS(11)</td>
<td>The radius of a given planet in A.U.</td>
</tr>
<tr>
<td>RMASS(11)</td>
<td>The relative gravitational constant of a stated planet with respect to the sun</td>
</tr>
<tr>
<td>ELMNT(80)</td>
<td>Contains the orbital elements of the planets</td>
</tr>
<tr>
<td>SPHERE(11)</td>
<td>The spheres of influence of the planets in A.U.</td>
</tr>
<tr>
<td>XP(6)</td>
<td>The position and velocity of a planet in inertial ecliptic coordinates</td>
</tr>
<tr>
<td>NO(11)</td>
<td>An array of planet codes being used to generate the virtual mass trajectory</td>
</tr>
</tbody>
</table>
### /CENTRIC/ MODE: NOMINAL, ERRAN, SIMUL

**IBARY**
- REFERENCE COORDINATE SYSTEM CODE
  - 0 HELIOCENTRIC COORDINATES
  - 1 BARYCENTRIC COORDINATES

**ICOORD**
- NON-FUNCTIONAL IN ERROR ANALYSIS MODE

**INITIAL**
- NON-FUNCTIONAL IN ERROR ANALYSIS MODE

### /COM / MODE: NOMINAL, ERRAN, SIMUL

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V'[16,7]</td>
<td>AN ARRAY WHICH STORES PERTINENT VECTORS USED IN THE CALCULATION OF THE VIRTUAL MASS TRAJECTORY (SEE LARGE ARRAY DEFNS IN SECT 5.3)</td>
</tr>
<tr>
<td>F[44,4]</td>
<td>CONTAINS THE POSITIONS AND VELOCITIES OF THE PLANETS AT A SPECIFIED TIME PLUS THE POSITIONS AND VELOCITIES OF THE SPACECRAFT RELATIVE TO THE PLANETS (SEE LARGE ARRAY DEFNS IN SECT 5.3)</td>
</tr>
<tr>
<td>PI</td>
<td>THE VALUE OF THE MATHEMATICAL CONSTANT PI</td>
</tr>
<tr>
<td>RAD</td>
<td>THE NUMBER OF DEGREES PER RADIANS</td>
</tr>
<tr>
<td>ITRAT</td>
<td>IN INTERNAL CODE USED TO DETERMINE HOW MANY ITERATIONS HAVE BEEN ACCOMPLISHED IN THE VIRTUAL MASS PROCEDURE</td>
</tr>
<tr>
<td>KOUNT</td>
<td>A CODE WHICH SPECIFIES WHETHER PRINT-OUT IS TO OCCUR AFTER THIS TIME INCREMENT</td>
</tr>
<tr>
<td>INCNNT</td>
<td>NUMBER OF INCREMENTS USED</td>
</tr>
<tr>
<td>INCPR</td>
<td>SPECIFIES AFTER HOW MANY TIME INCREMENTS PRINT-OUT IS TO OCCUR</td>
</tr>
<tr>
<td>INC</td>
<td>DETERMINE WHETHER THE ABOVE OPTION IS TO BE USED</td>
</tr>
<tr>
<td>IPR</td>
<td>A CODE WHICH DETERMINES IF PRINT-OUT IS TO OCCUR AFTER A SPECIFIED NUMBER OF DAYS</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>NBODYI</td>
<td>NUMBER OF BODIES CONSIDERED IN VIRTUAL MASS TRAJECTORY</td>
</tr>
<tr>
<td>NBODY</td>
<td>BASED ON ABOVE VALUE--EQUAL TO 4*NBODYI-3</td>
</tr>
<tr>
<td>IPRT(4)</td>
<td>SPECIFIES PRINT OPTIONS (IN STEAP TRAJECTORY THIS OPTION IS OMITTED. WHEN PRINT-OUT OCCURS ALL SECTIONS ARE AUTOMATICALLY PRINTED)</td>
</tr>
<tr>
<td>KL</td>
<td>PROBLEM NUMBER (NORMAL ONLY)</td>
</tr>
<tr>
<td>IPG</td>
<td>PAGE NUMBER (NORMAL ONLY)</td>
</tr>
<tr>
<td>LINCT</td>
<td>LINE COUNT (NORMAL ONLY)</td>
</tr>
<tr>
<td>LIMPGE</td>
<td>LINES PER PAGE (NORMAL ONLY)</td>
</tr>
</tbody>
</table>

---------------------------------------------
/CONST / MODE: ERRAN, SIMUL
---------------------------------------------

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMEGA</td>
<td>ROTATION RATE OF EARTH</td>
</tr>
<tr>
<td>EPS</td>
<td>OBLIQUITY OF EARTH</td>
</tr>
<tr>
<td>SAL(3)</td>
<td>ALTITUDES OF STATIONS</td>
</tr>
<tr>
<td>SLAT(3)</td>
<td>LATITUDES OF STATIONS</td>
</tr>
<tr>
<td>SLMON(3)</td>
<td>LONGITUDES OF STATIONS</td>
</tr>
<tr>
<td>DMCW(3)</td>
<td>CONSTANTS FROM WHICH DYNAMIC NOISE IS COMPUTED</td>
</tr>
<tr>
<td>MNCH(12)</td>
<td>MEASUREMENT NOISE CONSTANTS</td>
</tr>
<tr>
<td>NST</td>
<td>NUMBER OF STATIONS TO BE USED (MAXIMUM 3)</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UST(3)</td>
<td>DIRECTION COSINE ARRAYS OF THREE REFERENCE STARS</td>
</tr>
<tr>
<td>VST(3)</td>
<td>DIRECTION COSINE ARRAYS OF THREE REFERENCE STARS</td>
</tr>
<tr>
<td>WST(3)</td>
<td>DIRECTION COSINE ARRAYS OF THREE REFERENCE STARS</td>
</tr>
<tr>
<td>FOP</td>
<td>OFF-DIAGONAL ANNihilation VALUE FOR POSITION EIGENVALUES</td>
</tr>
<tr>
<td>FOV</td>
<td>OFF-DIAGONAL ANNihilation VALUE FOR VELOCITY EIGENVALUES</td>
</tr>
<tr>
<td>DELAXS</td>
<td>TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELECC</td>
<td>TARGET PLANET ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELICL</td>
<td>TARGET PLANET INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMOD</td>
<td>TARGET PLANET LONGITUDE OF THE ASCENDING NODE FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELW</td>
<td>TARGET PLANET ARGUMENT OF PERIAPSIS FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMA</td>
<td>TARGET PLANET MEAN ANOMALY FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMUS</td>
<td>SUN GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMUP</td>
<td>TARGET PLANET GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
</tbody>
</table>
ZERO THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES
ONE THE NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES
TWO THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES
THREE THE NUMBER THREE (3) TO NINE SIGNIFICANT FIGURES
FOUR THE NUMBER FOUR (4) TO NINE SIGNIFICANT FIGURES
FIVE THE NUMBER FIVE (5) TO NINE SIGNIFICANT FIGURES
EIGHT THE NUMBER EIGHT (8) TO NINE SIGNIFICANT FigURES
TEN THE NUMBER TEN (10) TO NINE SIGNIFICANT FIGURES
NINETY THE NUMBER NINETY (90) TO NINE SIGNIFICANT FIGURES
HALF THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES
THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES
THE NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES
THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES
THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES
THE NUMBER THREE (3) TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-1 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-2 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-3 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-4 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-5 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-6 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-7 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-8 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-9 TO NINE SIGNIFICANT FIGURES
THE NUMBER 1.E-50 TO NINE SIGNIFICANT FIGURES
THE MATHEMATICAL CONSTANT 2.0*PI
THE NUMBER 1.E-13 TO NINE SIGNIFICANT FIGURES
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEV(50)</td>
<td>TIMES OF EVENTS</td>
</tr>
<tr>
<td>TPT2(20)</td>
<td>PREDICTION TIMES</td>
</tr>
<tr>
<td>SIGRES</td>
<td>VARIANCE OF RESOLUTION ERROR</td>
</tr>
<tr>
<td>SIGPRO</td>
<td>VARIANCE OF PROPORTIONALITY ERROR</td>
</tr>
<tr>
<td>SIGALP</td>
<td>VARIANCE OF ERROR IN POINTING ANGLE 1</td>
</tr>
<tr>
<td>SIGBET</td>
<td>VARIANCE OF ERROR IN POINTING ANGLE 2</td>
</tr>
<tr>
<td>MP7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>P7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>TAU7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>AINC7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>ANODE7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>PERP7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>ECC7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>DV8(3)</td>
<td>ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV</td>
<td>NUMBER OF EVENTS</td>
</tr>
<tr>
<td>IEVENT(50)</td>
<td>CODES OF EVENTS</td>
</tr>
<tr>
<td>IMYP1</td>
<td>HYPERELLIPSOID CODE USED TO DETERMINE IF K=1, K=3, OR BOTH</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IEIG</td>
<td>CODE USED TO DECIDE IF BOTH POSITION AND VELOCITY EIGENVECTORS ARE REQUESTED</td>
</tr>
<tr>
<td>ICT3(20)</td>
<td>CODES WHICH DETERMINE WHICH GUIDANCE POLICIES ARE BEING USED</td>
</tr>
<tr>
<td>NPE</td>
<td>NUMBER OF PREDICTION EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>NGE</td>
<td>NUMBER OF GUIDANCE EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>IPOL</td>
<td>CODE WHICH DETERMINES IF FIXED-TIME-OF-ARRIVAL GUIDANCE EVENT HAS OCCURRED</td>
</tr>
<tr>
<td>IIPO1</td>
<td>CODE WHICH DETERMINES IF EITHER TWO-VARIABLE OR THREE-VARIABLE B-PLANE GUIDANCE POLICY HAS OCCURRED</td>
</tr>
<tr>
<td>ICOQ3(20)</td>
<td>ARRAY OF CODES WHICH DETERMINE WHICH EXECUTION POLICIES ARE TO BE USED IN GUIDANCE EVENTS</td>
</tr>
<tr>
<td>NEV1</td>
<td>TOTAL NUMBER OF EIGENVECTOR EVENTS</td>
</tr>
<tr>
<td>NEV2</td>
<td>TOTAL NUMBER OF PREDICTION EVENTS</td>
</tr>
<tr>
<td>NEV3</td>
<td>TOTAL NUMBER OF GUIDANCE EVENTS</td>
</tr>
<tr>
<td>NEV4</td>
<td>TOTAL NUMBER OF -COMCON- EVENTS</td>
</tr>
<tr>
<td>NQE</td>
<td>QUASI-LINEAR FILTERING EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>NEV5</td>
<td>TOTAL NUMBER OF QUASI-LINEAR FILTERING EVENTS</td>
</tr>
<tr>
<td>NEV6</td>
<td>TOTAL NUMBER OF ADAPTIVE FILTERING EVENTS. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NAE</td>
<td>ADAPTIVE FILTERING EVENTS HAVING OCCURRED. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NAF6(20)</td>
<td>ARRAY OF ADAPTIVE FILTERING EVENT CODES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
</tbody>
</table>
IOPT7  ORBIT INSERTION VARIABLE, NON-FUNCTIONAL IN EXISTING PROGRAM

NEV0  ORBIT INSERTION VARIABLE, NON-FUNCTIONAL IN EXISTING PROGRAM

NEV9  ORBIT INSERTION VARIABLES, NON-FUNCTIONAL IN EXISTING PROGRAM

NEV10 ORBIT INSERTION VARIABLES, NON-FUNCTIONAL IN EXISTING PROGRAM

NEV11 ORBIT INSERTION VARIABLES, NON-FUNCTIONAL IN EXISTING PROGRAM

-------------------------------------------------------------------------
/EXE / MODE ERRAN, SIMUL
-------------------------------------------------------------------------

XXIN(6) STATE VECTOR TRANSFERRED TO EXECUT OR EXCUTS

DIPX JULIAN DATE TRANSFERRED TO EXECUT OR EXCUTS

DELPX(3) VELOCITY CORRECTION TO BE MODELED AS AN IMPULSE SERIES

QK(6,6) EFFECTIVE EXECUTION COVARIANCE MATRIX

DUMMYQ(4) ARRAY OF EXECUTION ERROR VARIANCES

INPX IMPULSE SERIES CODE
<table>
<thead>
<tr>
<th>LIN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG(6,6)</td>
<td>POSITION/VELOCITY CONTROL COVARIANCE</td>
</tr>
<tr>
<td>CXSG(6,24)</td>
<td>CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>CXUG(6,8)</td>
<td>CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS</td>
</tr>
<tr>
<td>CXVG(6,15)</td>
<td>CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS</td>
</tr>
<tr>
<td>PSG(24,24)</td>
<td>SOLVE-FOR PARAMETER CONTROL COVARIANCE</td>
</tr>
<tr>
<td>CXSUG(24,8)</td>
<td>CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS</td>
</tr>
<tr>
<td>CXSVG(24,15)</td>
<td>CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS</td>
</tr>
<tr>
<td>XG(6)</td>
<td>POSITION/VELOCITY STATE AT MOST RECENT GUIDANCE EVENT</td>
</tr>
<tr>
<td>TG</td>
<td>TRAJECTORY TIME AT MOST RECENT GUIDANCE EVENT</td>
</tr>
<tr>
<td>EM(2,6)</td>
<td>VARIATION MATRIX RELATING POSITION/VELOCITY DEVIATIONS TO B.T AND B.R DEVIATIONS</td>
</tr>
</tbody>
</table>

**/IMPTAR/ MODE NOMNAL, ERRAN, SIMUL**

**ANG**
TARGET INCLINATION CONVERTED FROM INPUT FORMAT TO VALUE BETWEEN 0 AND 180 DEGREES AND SATISFYING APPROACH ASYMPTOTE CONSTRAINT
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTAR(3)</td>
<td>Target values of SMA, B.T, and B.R in Lunar targeting</td>
</tr>
<tr>
<td>PCON(3)</td>
<td>Perturbations in controls (Alpha, Delta, Theta)</td>
</tr>
<tr>
<td>TTOL(3)</td>
<td>Allowable tolerances in SMA, B.T, B.R</td>
</tr>
<tr>
<td>BCON(3)</td>
<td>Maximum step sizes of controls</td>
</tr>
<tr>
<td>RI(6)</td>
<td>Geocentric state of S/C at Lunar SOI</td>
</tr>
<tr>
<td>RMQ(6)</td>
<td>Geocentric state of Center of Moon at TSI in equatorial coordinates</td>
</tr>
<tr>
<td>RSI(6)</td>
<td>Selenocentric state of S/C at Lunar SOI</td>
</tr>
<tr>
<td>RME(6)</td>
<td>Geocentric state of Center of Moon in ecliptic coordinates at TSI</td>
</tr>
<tr>
<td>DECLIN</td>
<td>Declination of approach asymptote with respect to Lunar equator</td>
</tr>
<tr>
<td>OTAR(3)</td>
<td>Desired values of SMA, RCA, and INC</td>
</tr>
<tr>
<td>TCA</td>
<td>J.D. of time at Lunar closest approach (desired)</td>
</tr>
<tr>
<td>RCA</td>
<td>Radius of closest approach to Moon (desired)</td>
</tr>
<tr>
<td>SMA</td>
<td>Semi-major axis of Lunar hyperbola (desired)</td>
</tr>
<tr>
<td>CAI</td>
<td>Desired closest approach equatorial inclination</td>
</tr>
<tr>
<td>RPE</td>
<td>Radius of Earth parking orbit</td>
</tr>
<tr>
<td>TSI</td>
<td>Projected J.D. at SOI intersection</td>
</tr>
<tr>
<td>EMU</td>
<td>Gravitational constant of Earth (km^3/sec^2)</td>
</tr>
</tbody>
</table>
TSPH

RADIUS OF LUNAR SOI (KM)

EQLQ(3,3)

TRANSFORMATION MATRIX FROM EARTH-EQUATORIAL TO LUNAR EQUATORIAL COORDINATES

ITAG

FLAG SPECIFYING STAGE OF TARGETING
=1 IN SMA TARGETING
=0 IN SMA, INC, RCA TARGETING

-----------------------------------------------

/MEAS / MODE: ERRAN, SIMUL

TMN(1000)

TIMES OF MEASUREMENTS

MCODE(1000)

ARRAY OF MEASUREMENT CODES

WNN

TOTAL NUMBER OF MEASUREMENTS

MCHTR

NUMBER OF MEASUREMENTS HAVING OCCURRED

-----------------------------------------------

/MISC / MODE: ERRAN, SIMUL

ACC

ACCURACY FIGURE USED IN VIRTUAL MASS PROGRAM

FACP

POSITION FACTOR USED IN NUMERICAL DIFFERENCING

FACV

VELOCITY FACTOR USED IN NUMERICAL DIFFERENCING

BIA(12)

MEASUREMENT BIASES

IDNF

DYNAMIC NOISE FLAG

ICOOR

STATE VECTOR CODE WHICH DETERMINES IN WHICH COORDINATE SYSTEM THE VECTOR IS READ IN

ITR

MODE FLAG

IMNF

MEASUREMENT NOISE FLAG

ISP2

SPHERE OF INFLUENCE FLAG
<table>
<thead>
<tr>
<th>Event Name</th>
<th>Measurement Name</th>
<th>Component Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVNM(11)</td>
<td>MNNAME(12,3)</td>
<td>CMPNM(30)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF(6)</td>
<td>Final Targeted Nominal State Vector</td>
</tr>
<tr>
<td>RF1(6)</td>
<td>Final Most Recent Nominal State Vector</td>
</tr>
<tr>
<td>RI(6)</td>
<td>Initial Targeted Nominal State Vector</td>
</tr>
<tr>
<td>TENV</td>
<td>Time of Current Event</td>
</tr>
<tr>
<td>RI1(6)</td>
<td>Initial Most Recent Nominal State Vector</td>
</tr>
<tr>
<td>ICODE</td>
<td>Event Code</td>
</tr>
<tr>
<td>NAFC</td>
<td>Non-Functional Adaptive Filter Code</td>
</tr>
<tr>
<td>NR</td>
<td>Number of Rows in the Observation Matrix</td>
</tr>
</tbody>
</table>
/OVERL / MODE ERRAN, SIMUL

DTIME TIME INTERVAL BETWEEN ORBITAL INSERTION DECISION AND EXECUTION

/OVERR / MODE ERRAN

RI(6) STATE VECTOR AT EVENT TIME
TEVN EVENT TIME

/OVERX / MODE ERRAN, SIMUL

IX NONLINEAR GUIDANCE CODE
JX GUIDANCE EVENT COUNTER
XIM(6) STATE VECTOR TRANSFERRED TO NONLIN

/OVERZ / MODE ERRAN, SIMUL

RF(6) FINAL TARGETED STATE VECTOR
RF1(6) FINAL MOST RECENT NOMINAL STATE VECTOR
IGP MIDCOURSE GUIDANCE POLICY CODE
GA(3,6) GUIDANCE MATRIX
A(2,3) FTA IMPACT PLANE TRANSFORMATION MATRIX
XMUS(2) NOMINAL IMPACT PLANE TARGET STATE
EXEC(3,3) EXECUTION ERROR COVARIANCE MATRIX
CR CAPTURE RADIUS OF TARGET PLANET
POI PROBABILITY OF IMPACT
XLAM(2,2) PROJECTION OF TARGET CONDITION COVARIANCE MATRIX INTO THE IMPACT PLANE
XLAMI(2,2) INVERSE OF XLAM(2,2)
DVRB(3) VELOCITY CORRECTION REQUIRED TO REMOVE AIMPOINT BIAS
DVUP(3) UPDATE VELOCITY CORRECTION
PSTAR NOMINAL PROBABILITY DENSITY FUNCTION EVALUATED AT TARGET PLANET CENTER
DVN(3) COMMANDED VELOCITY CORRECTION TRANSFERRED TO IBAIM
DELV(3,10) ARRAY OF EXTERNALLY-SUPPLIED VELOCITY CHANGES
IIGP MIDCOURSE GUIDANCE POLICY CODE
IBIAS BIASED AIMPOINT GUIDANCE EVENT FLAG
  = 0 AIMPOINT NOT BIASED
  = 1 AIMPOINT BIASED
IDENS PROBABILITY DENSITY FUNCTION CODE. NON-FUNCTIONAL
/PRT / MODE: NOMNAL

MONTH(12) NAMES OF MONTHS
PLANET(11) NAMES OF GRAVITATIONAL BODIES

/PRT / MODE: ERRAN, SIMUL

PLANET(11) NAMES OF PLANETS
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PULMAG</td>
<td>Thrust magnitude of pulsing engine</td>
</tr>
<tr>
<td>PULMAS</td>
<td>Nominal mass of spacecraft during pulsing arc</td>
</tr>
<tr>
<td>DUR</td>
<td>Duration of single pulse</td>
</tr>
<tr>
<td>DTI</td>
<td>Time interval (days) between successive pulses</td>
</tr>
<tr>
<td>DVI(3)</td>
<td>Velocity increment added on typical pulse</td>
</tr>
<tr>
<td>DVF(3)</td>
<td>Velocity increment added on final pulse</td>
</tr>
<tr>
<td>PULT</td>
<td>Total time interval of pulsing arc</td>
</tr>
<tr>
<td>RK(2,3)</td>
<td>Position vectors of launch and target planets at impulsive time (midpoint of pulsing arc)</td>
</tr>
<tr>
<td>VK(2,3)</td>
<td>Velocity vectors of launch and target planets at impulsive time (midpoint of pulsing arc)</td>
</tr>
<tr>
<td>FS(2,5)</td>
<td>F-series coefficients of launch and target bodies</td>
</tr>
<tr>
<td>GS(2,4)</td>
<td>G-series coefficients of launch and target bodies</td>
</tr>
<tr>
<td>GG(3)</td>
<td>Gravitational constants of sun, launch, and target bodies</td>
</tr>
<tr>
<td>NPUL</td>
<td>Number of pulses in pulsing arc</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>XBDT</td>
<td>ORIGINAL VALUE OF B.T IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XBDR</td>
<td>ORIGINAL VALUE OF B.R IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XDSI</td>
<td>ORIGINAL VALUE OF TSI IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XRSI(3)</td>
<td>ORIGINAL VALUE OF RSI IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XVSI(3)</td>
<td>ORIGINAL VALUE OF VSI IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XRC(6)</td>
<td>ORIGINAL VALUE OF RC IN NONLINEAR GUIDANCE</td>
</tr>
<tr>
<td>XDC</td>
<td>ORIGINAL VALUE OF DC IN NONLINEAR GUIDANCE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMUSB</td>
<td>BIAS IN GRAVITATIONAL CONSTANT OF SUN</td>
</tr>
<tr>
<td>DMUPB</td>
<td>BIAS IN GRAVITATIONAL CONSTANT OF TARGET PLANET</td>
</tr>
<tr>
<td>DAB</td>
<td>BIAS IN SEMI-MAJOR AXIS OF TARGET PLANET</td>
</tr>
<tr>
<td>DEB</td>
<td>BIAS IN ECCENTRICITY OF TARGET PLANET</td>
</tr>
<tr>
<td>DIB</td>
<td>BIAS IN INCLINATION OF TARGET PLANET</td>
</tr>
<tr>
<td>DMOCB</td>
<td>BIAS IN LONGITUDE OF ASCENDING NODE</td>
</tr>
<tr>
<td>DWB</td>
<td>BIAS IN ARGUMENT OF PERIAPSIS</td>
</tr>
<tr>
<td>DMAB</td>
<td>BIAS IN MEAN ANOMALY</td>
</tr>
<tr>
<td>TTIM1</td>
<td>FIRST TIME USED FOR UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>TTIM2</td>
<td>SECOND TIME USED FOR UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>UNMAC(3,3)</td>
<td>UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>SLB(9)</td>
<td>BIASES IN STATION LOCATION CONSTANTS</td>
</tr>
<tr>
<td>AVM(12)</td>
<td>VARIANCE OF ACTUAL MEASUREMENT NOISE</td>
</tr>
<tr>
<td>ARE(20)</td>
<td>ACTUAL RESOLUTION ERROR</td>
</tr>
<tr>
<td>APRO(20)</td>
<td>ACTUAL PROPORTIONALITY ERROR</td>
</tr>
<tr>
<td>AALP(20)</td>
<td>ACTUAL ERROR IN POINTING ANGLE 1</td>
</tr>
<tr>
<td>ABET(20)</td>
<td>ACTUAL ERROR IN POINTING ANGLE 2</td>
</tr>
<tr>
<td>IAMNF</td>
<td>ACTUAL MEASUREMENT NOISE FLAG</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>XI1(6)</td>
<td>Initial state vector of most recent nominal trajectory</td>
</tr>
<tr>
<td>XF1(6)</td>
<td>Final state vector of most recent nominal trajectory</td>
</tr>
<tr>
<td>ADEVX(6)</td>
<td>Actual deviation in the state vector</td>
</tr>
<tr>
<td>ADEVXS(24)</td>
<td>Actual deviation in solve-for parameters</td>
</tr>
<tr>
<td>EDEVX(6)</td>
<td>Estimated deviation in the state vector</td>
</tr>
<tr>
<td>EDEVXS(24)</td>
<td>Estimated deviation in solve-for parameters</td>
</tr>
<tr>
<td>W(6)</td>
<td>Actual dynamic noise</td>
</tr>
<tr>
<td>ZI(6)</td>
<td>Initial actual state vector</td>
</tr>
<tr>
<td>ZF(6)</td>
<td>Final actual state vector after adding effect of unmodeled acceleration</td>
</tr>
<tr>
<td>ANOIS(4)</td>
<td>Actual white noise</td>
</tr>
<tr>
<td>RES(4)</td>
<td>Residual</td>
</tr>
<tr>
<td>EY(4)</td>
<td>Estimated measurement</td>
</tr>
<tr>
<td>AY(4)</td>
<td>Actual measurement</td>
</tr>
<tr>
<td>AR(4,4)</td>
<td>Actual measurement noise</td>
</tr>
<tr>
<td>ADEVXB(6)</td>
<td>Actual deviation in state vector at beginning of trajectory</td>
</tr>
<tr>
<td>ADEVSX(24)</td>
<td>Actual deviation in solve-for parameters at beginning of trajectory</td>
</tr>
<tr>
<td>AYMEY(4)</td>
<td>Actual measurement minus estimated measurement</td>
</tr>
<tr>
<td>EDEVXM(6)</td>
<td>Estimated deviation in the state vector</td>
</tr>
<tr>
<td>EDEVSM(24)</td>
<td>Estimated deviation in solve-for parameters</td>
</tr>
</tbody>
</table>

(for adaptive filtering)
ARRAY OF PLANET CODES IN ACTUAL TRAJECTORY
ACC1  ACCURACY USED IN ACTUAL TRAJECTORY
NBODI  NUMBER OF BODIES IN ACTUAL TRAJECTORY

POSITION/VELOCITY COVARIANCE
CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS
CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
SOLVE-FOR PARAMETER COVARIANCE
CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS
DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0(15,15)$</td>
<td>Measurement consider parameter covariance matrix</td>
</tr>
<tr>
<td>Note</td>
<td>If the entire covariance matrix were assembled from the given partitions the resultant matrix would be $P(53,53)$ with the symmetric structure:</td>
</tr>
</tbody>
</table>

$$
P(5,5) \quad CXS(6,24), \quad CXU(6,8) \quad CXV(6,15) \quad P(53,53) = \\
\begin{array}{ccc}
PS(24,24) & CXSU(24,8) & CXSV(24,15) \\
U_0(8,8) & CU_V(8,15) & V_0(15,15)
\end{array}
$$

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi(6,6)$</td>
<td>Position/velocity state transition matrix</td>
</tr>
<tr>
<td>$TXXS(6,24)$</td>
<td>State transition matrix partition associated with solve-for parameters</td>
</tr>
<tr>
<td>$TXU(6,8)$</td>
<td>State transition matrix partition associated with dynamic consider parameters</td>
</tr>
<tr>
<td>$Q(6,6)$</td>
<td>Dynamic noise covariance matrix</td>
</tr>
<tr>
<td>$R(4,4)$</td>
<td>Measurement noise covariance matrix</td>
</tr>
<tr>
<td>$AK(6,4)$</td>
<td>Kalman gain constant for position/velocity state</td>
</tr>
<tr>
<td>$S(24,4)$</td>
<td>Kalman gain constant for solve-for parameters</td>
</tr>
<tr>
<td>$H(4,6)$</td>
<td>Observation matrix relating observables to position/velocity state</td>
</tr>
<tr>
<td>$AH(4,24)$</td>
<td>Observation matrix relating observables to solve-for parameter state</td>
</tr>
<tr>
<td>$G(4,8)$</td>
<td>Observation matrix relating observables to dynamic consider parameter state</td>
</tr>
</tbody>
</table>
AL(4,15) OBSERVATION MATRIX RELATING OBSERVABLES TO MEASUREMENT CONSIDER PARAMETER STATE
NFR(4,4) NON-FUNCTIONAL IN PRESENT ERROR ANALYSIS PROGRAM
PB(6,6) POSITION/VELOCITY COVARIANCE AT INITIAL TIME
CXXSB(6,24) CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS AT INITIAL TIME
CXXSB(6,6) CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS AT INITIAL TIME
CXXSB(6,15) CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS AT INITIAL TIME
PSB(24,24) SOLVE-FOR PARAMETER COVARIANCE AT INITIAL TIME
CXXSB(24,6) CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS AT INITIAL TIME
CXXSB(24,15) CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS AT INITIAL TIME
PP(6,6) POSITION/VELOCITY COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
CXXSP(6,24) CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXXSP(6,6) CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
CXXSP(6,15) CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
PSP(24,24) SOLVE-FOR PARAMETER COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT
CXSUP(24,8) CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

CXSVP(24,15) CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

---

/stvec / model erran, simul

XI(6) INITIAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XF(6) FINAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL
XB(6) BEGINNING ORIGINAL NOMINAL VEHICLE STATE VECTOR
NDIM1 DIMENSION OF SOLVE-FOR PARAMETER STATE
NDIM2 DIMENSION OF DYNAMIC CONSIDER STATE
NDIM3 DIMENSION OF MEASUREMENT CONSIDER PARAMETER STATE
IAUGIN(24) INPUT AUGMENTATION VECTOR OF ONE#S AND ZERO#S
IAUG(24) AUGMENTATION VECTOR
IAUGDC(8) DYNAMIC CONSIDER AUGMENTATION VECTOR
IAUGMC(15) MEASUREMENT CONSIDER AUGMENTATION VECTOR
/STAREAL/ MODE: NOMINAL, ERRAN, SIMUL

AC(5,10)  ACCURACY LEVELS (UP TO 5) USED IN EACH GUIDANCE EVENT
PHI(3,3)  TARGETING MATRIX
TIMG(10)  TIMES OF EACH GUIDANCE EVENT REFERENCED TO EPOCH-
          -INITIAL TIME, SOI TIME, OR CA TIME
TAR(6,10)  DESIRED VALUES OF TARGET PARAMETERS (UP TO 6
          AVAILABLE) FOR EACH GUIDANCE EVENT
DAUX(3)  DESIRED AUXILIARY PARAMETER VALUES OF ITERATE
AAUX(3)  ACTUAL AUXILIARY VALUES OF ITERATE
DTAR(3)  DESIRED TARGET VALUES OF ITERATE
ATAR(3)  ACTUAL TARGET VALUES OF ITERATE
TOL(6,10)  ALLOWABLE TOLERANCES OF TARGET PARAMETERS FOR
          EACH GUIDANCE EVENT
TOLR(6)  NOT USED IN CURRENT TARGET VERSION
CTOL(6)  TOLERANCES FOR CURRENT EVENT
FAC(3)  SCALING FACTORS USED IN BAD STEP CHECK
TMPR   DAYS BETWEEN PRINTOUTS OF NOMINAL TRAJECTORY
PERV(10)  PERTURBATION SIZE FOR VELOCITY COMPONENTS IN
          CONSTRUCTING SENSITIVITY MATRICES IN TARGETING
          EVENTS
DINTG(10)  NOT USED IN CURRENT TARGET VERSION
DT(10)   JULIAN DATES OF TARGET TIMES
DELV(3,10)  EXTERNALLY SUPPLIED VELOCITY CORRECTION OR
            VELOCITY INCREMENT COMPUTED BY INSERTION DECISION
<table>
<thead>
<tr>
<th>TRTH</th>
<th>TRAJECTORY TIME (DAYS) REF. TO INJECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIN(6)</td>
<td>CURRENT STATE VECTOR AT I-TH EVENT</td>
</tr>
<tr>
<td>TIN</td>
<td>JULIAN DATE AT INJECTION</td>
</tr>
<tr>
<td>D1</td>
<td>JULIAN DATE ASSOCIATED WITH RIN ARRAY</td>
</tr>
<tr>
<td>DG(10)</td>
<td>JULIAN DATES OF EVENT TIMES</td>
</tr>
<tr>
<td>DELTAT</td>
<td>NUMBER OF DAYS INTEGRATION IS TO CONTINUE IF NO OTHER STOPPING CONDITION OCCURS</td>
</tr>
<tr>
<td>THU</td>
<td>GRAVITATIONAL CONSTANT OF TARGET PLANET</td>
</tr>
<tr>
<td>RRF(3)</td>
<td>SPACECRAFT POSITION AT END OF INTEGRATION</td>
</tr>
<tr>
<td>DELTAV(3)</td>
<td>CORRECTIONS TO BE ADDED TO VELOCITY COMPONENTS FOR NEXT ITERATION</td>
</tr>
<tr>
<td>DVMAX(10)</td>
<td>MAXIMUM ALLOWABLE CHANGE IN ANY VELOCITY COMPONENT FOR EACH EVENT</td>
</tr>
<tr>
<td>ACKT</td>
<td>TRAJECTORY INTEGRATION ACCURACY</td>
</tr>
<tr>
<td>EQECP(3,3)</td>
<td>TRANSFORMATION FROM ECLIPTIC TO EQUATORIAL SYSTEM FOR TARGET PLANE</td>
</tr>
<tr>
<td>TINS</td>
<td>INTERNAL CLOCK TIME AT START OF COMPUTER RUN</td>
</tr>
<tr>
<td>SPHFAC(10)</td>
<td>REDUCTION FACTORS FOR TARGET PLANET SPHERE OF INFLUENCE FOR EACH EVENT</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOGYD</td>
<td>Total number of guidance events</td>
</tr>
<tr>
<td>KTIM(10)</td>
<td>Epoch to which guidance event times are referenced</td>
</tr>
<tr>
<td>KTYP(10)</td>
<td>Type of guidance event for each event</td>
</tr>
<tr>
<td>KMXQ(10)</td>
<td>Compute/execute modes for each guidance event</td>
</tr>
<tr>
<td>MOL(10)</td>
<td>Execution models for each guidance event</td>
</tr>
<tr>
<td>NPAR(10)</td>
<td>Number of target parameters in each targeting event</td>
</tr>
<tr>
<td>KTAR(6,10)</td>
<td>Codes of target parameters (up to 6) for each targeting event or orbit insertion option for each insertion event</td>
</tr>
<tr>
<td>KEYTAR(3)</td>
<td>Key defining desired target parameters for current event</td>
</tr>
<tr>
<td>MAT(10)</td>
<td>Targeting matrix computation code for each targeting event</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>IBADS(10)</td>
<td>Bad step flags for each targeting event</td>
</tr>
<tr>
<td></td>
<td>-1 Never use bad step check</td>
</tr>
<tr>
<td></td>
<td>-2 Use bad step check at final level only</td>
</tr>
<tr>
<td></td>
<td>-3 Use bad step check at all levels</td>
</tr>
<tr>
<td>N0IT(10)</td>
<td>The number of total iterations allowed at the first and last levels of targeting events for each guidance event</td>
</tr>
<tr>
<td>MAXB(10)</td>
<td>The number of bad steps allowed during any targeting event</td>
</tr>
<tr>
<td>LEVELS</td>
<td>Number of accuracy levels for current event</td>
</tr>
<tr>
<td>LEV</td>
<td>Current level in current targeting event</td>
</tr>
<tr>
<td>NITS</td>
<td>Allowable number of iterations for current event</td>
</tr>
<tr>
<td>MAXBAD</td>
<td>Maximum number of bad iterations for current event</td>
</tr>
<tr>
<td>IBASE</td>
<td>Bad step check indicator for current event</td>
</tr>
<tr>
<td>MATX</td>
<td>Matrix computation code for current targeting event (see defn of mat)</td>
</tr>
<tr>
<td>ISTART</td>
<td>Stage of initial targeting</td>
</tr>
<tr>
<td></td>
<td>-0 No targeting started</td>
</tr>
<tr>
<td></td>
<td>-1 First phase started and have targeting matrix</td>
</tr>
<tr>
<td></td>
<td>-2 Second phase started and have matrix</td>
</tr>
<tr>
<td>IFHASE</td>
<td>Phase counter for current targeting event</td>
</tr>
<tr>
<td>NOPHAS</td>
<td>Number of targeting phases for current event</td>
</tr>
<tr>
<td>ITARM</td>
<td>Flag to control construction of targeting matrix</td>
</tr>
<tr>
<td></td>
<td>-0 Do not compute targeting matrix</td>
</tr>
<tr>
<td></td>
<td>-1 Compute targeting matrix on current iteration</td>
</tr>
<tr>
<td>IBAD</td>
<td>Bad step flag for current accuracy level</td>
</tr>
<tr>
<td></td>
<td>-1 Do not check for bad step</td>
</tr>
<tr>
<td></td>
<td>-2 Check for bad step</td>
</tr>
</tbody>
</table>
ISTOP = STOPPING CONDITION INDICATOR IN SUBROUTINE TARGET
=1 STOP ON TIME
=2 STOP AT SPHERE OF INFLUENCE
=3 STOP AT CLOSEST APPROACH

NOPAR = NUMBER OF TARGET PARAMETERS FOR CURRENT EVENT

KWIT = TERMINATION FLAG
=0 CONTINUE RUN
=1 TERMINATE RUN

IPRE = CASE FLAG
=0 FIRST CASE
=1 STACKED CASE

NCPR = NUMBER OF INTEGRATION INCREMENTS BETWEEN PRINTOUTS OF NOMINAL TRAJECTORY

IFINT(10) = NOT USED IN THIS TRAJECTORY VERSION

KGYD(10) = INDICES OF EVENTS TO BE PROCESSED

KSICA = FLAG INDICATING STAGE OF NOMINAL TRAJECTORY
=1 SOI NOT YET INTERSECTED
=2 SOI INTERSECTED BUT NO CLOSEST APPROACH
=3 CLOSEST APPROACH ALREADY ENCOUNTERED

KUR = INDEX OF CURRENT EVENT

KAXTAR(3) = KEY DEFINING AUXILIARY PARAMETERS FOR CURRENT EVENT

LVLS(10) = NUMBER OF ACCURACY LEVELS TO BE USED ON EACH TARGETING EVENT

NOSOI = OUTER TARGETING FLAG
=0 NORMAL TARGETING
=1 OUTER TARGETING
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTAR(6,10)</td>
<td>Desired Target Values</td>
</tr>
<tr>
<td>XTOL(6,10)</td>
<td>Tolerances on Target Parameters</td>
</tr>
<tr>
<td>XAC(5,10)</td>
<td>Accuracy Levels Employed in Targeting</td>
</tr>
<tr>
<td>XPERV(10)</td>
<td>Velocity Perturbation Used to Compute Targeting Matrix</td>
</tr>
<tr>
<td>XOVMAX(10)</td>
<td>Maximum Allowable Velocity Correction</td>
</tr>
<tr>
<td>XFAC(10)</td>
<td>Sphere of Influence Factors</td>
</tr>
<tr>
<td>XDELV(3,10)</td>
<td>Nonlinear Velocity Correction</td>
</tr>
<tr>
<td>TGT3(10)</td>
<td>Desired Target Times Referenced to Initial Trajectory Time</td>
</tr>
<tr>
<td>LKTAR(6,10)</td>
<td>Array Defining Target Parameters</td>
</tr>
<tr>
<td>LKTP(10)</td>
<td>Array of Target Planets</td>
</tr>
<tr>
<td>LKLP(10)</td>
<td>Array of Launch Planets</td>
</tr>
<tr>
<td>LMPAR(10)</td>
<td>Number of Target Parameters Desired</td>
</tr>
<tr>
<td>LLVLS(10)</td>
<td>Number of Integration Accuracy Levels Used</td>
</tr>
</tbody>
</table>
/TIM / MODE: ERRAN, SIMUL

DATEJ  JULIAN DATE OF INITIAL TRAJECTORY TIME
       (REFERENCED TO 1950)
TRTM1  INITIAL TRAJECTORY TIME
DELTM  TIME INCREMENT
FNTM   FINAL TRAJECTORY TIME
UNIVT  UNIVERSAL TIME
TRTMB  TRAJECTORY TIME AT BEGINNING OF TRAJECTORY

/TIM2 / MODE: SIMUL

T1     EIGENVECTOR EVENT TIMES
T2     PREDICTION EVENT STARTING TIMES
T4     CONIC COMPUTATION EVENT TIMES
T5     QUASI-LINEAR EVENT TIMES
T6     NOT USED
T7     NOT USED
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTMAX</td>
<td>Maximum time increment for which ISTMC is valid</td>
</tr>
<tr>
<td>ACCNO</td>
<td>Accuracy used in numerical differencing if NDACC indicates</td>
</tr>
<tr>
<td>DTSUN</td>
<td>State transition integration interval when the sun is central body and ISTM1 = 1</td>
</tr>
<tr>
<td>DTPLAN</td>
<td>State transition integration interval when target planet is central body and ISTM1 = 1</td>
</tr>
<tr>
<td>NTMC</td>
<td>Nominal trajectory code</td>
</tr>
<tr>
<td>ISTMC</td>
<td>State transition matrix code</td>
</tr>
<tr>
<td>ISTM1</td>
<td>Alternate state transition matrix code</td>
</tr>
<tr>
<td>NDACC</td>
<td>Numerical differencing accuracy code</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCA1(6)</td>
<td>State at closest approach on original nominal</td>
</tr>
<tr>
<td>RCA2(6)</td>
<td>State at closest approach on most recent nominal</td>
</tr>
<tr>
<td>RCA3(6)</td>
<td>State at closest approach on actual trajectory</td>
</tr>
<tr>
<td>RSOI1(3)</td>
<td>Position at sphere of influence on original nominal</td>
</tr>
<tr>
<td>RSOI2(3)</td>
<td>Position at sphere of influence on most recent nominal</td>
</tr>
<tr>
<td>RSOI3(3)</td>
<td>Position at sphere of influence on actual trajectory</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VSOI1(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>VSOI2(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>VSOI3(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>TCA1</td>
<td>TIME AT CLOSEST APPROACH OF ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>TCA2</td>
<td>TIME AT CLOSEST APPROACH OF MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>TCA3</td>
<td>TIME AT CLOSEST APPROACH OF ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>TSOI1</td>
<td>TIME AT SPHERE OF INFLUENCE OF ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>TSOI2</td>
<td>TIME AT SPHERE OF INFLUENCE OF MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>TSOI3</td>
<td>TIME AT SPHERE OF INFLUENCE OF ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>BSI1</td>
<td>B ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>BSI2</td>
<td>B ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>BSI3</td>
<td>B ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>BDTSI1</td>
<td>B DOT T ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>BDTSI2</td>
<td>B DOT T ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>BDTSI3</td>
<td>B DOT T ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>BDRSI1</td>
<td>B DOT R ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>BDRSI2</td>
<td>B DOT R ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>BDRSI3</td>
<td>B DOT R ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>ISOI1</td>
<td>SPHERE OF INFLUENCE CODE FOR ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>ISOI2</td>
<td>SPHERE OF INFLUENCE CODE FOR MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>ISO13</td>
<td>SPHERE OF INFLUENCE CODE FOR ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>ICA1</td>
<td>CLOSEST APPROACH CODE FOR ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>ICA2</td>
<td>CLOSEST APPROACH CODE FOR MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>ICA3</td>
<td>CLOSEST APPROACH CODE FOR ACTUAL TRAJECTORY</td>
</tr>
</tbody>
</table>

---

/IMTRIX/ MODE: NOMINAL, ERRAN, SIMUL

---

CHI(3,3) SENSITIVITY MATRIX (TRANSFERRED FOR OUTPUT)
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALNGTH</td>
<td>LENGTH UNITS PER A.U.</td>
</tr>
<tr>
<td>TM</td>
<td>TIME UNITS PER DAY</td>
</tr>
<tr>
<td>DELTP</td>
<td>PRINT INCREMENTS (IN DAYS)</td>
</tr>
<tr>
<td>RC(6)</td>
<td>STATE AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>OC</td>
<td>JULIAN DATE, EPOCH 1900, AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>RSI(3)</td>
<td>POSITION AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>VS(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>OSI</td>
<td>JULIAN DATE, EPOCH 1900, AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>RVS(6)</td>
<td>POSITION OF VEHICLE RELATIVE TO VIRTUAL MASS</td>
</tr>
<tr>
<td>VMU</td>
<td>GRAVITATIONAL CONSTANT OF VIRTUAL MASS</td>
</tr>
<tr>
<td>B</td>
<td>B AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>BDT</td>
<td>B DOT T</td>
</tr>
<tr>
<td>BDR</td>
<td>B DOT R</td>
</tr>
<tr>
<td>DELTH</td>
<td>INCREMENT IN TRUE ANOMALY USED</td>
</tr>
<tr>
<td>TIMINT</td>
<td>TOTAL TIME USED</td>
</tr>
<tr>
<td>RE(6)</td>
<td>POSITION AND VELOCITY OF EARTH</td>
</tr>
<tr>
<td>RTP(6)</td>
<td>POSITION AND VELOCITY OF TARGET PLANET</td>
</tr>
<tr>
<td>CAINC</td>
<td>INCLINATION AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>RCA</td>
<td>MAGNITUDE OF CLOSEST APPROACH POSITION VECTOR</td>
</tr>
<tr>
<td>TACA</td>
<td>TRAJECTORY SEMIMAJOR AXIS WITH RESPECT TO TARGET BODY AT CLOSEST APPROACH TO TARGET BODY</td>
</tr>
</tbody>
</table>
SSS(3)  DIRECTION COSINE VECTOR OF SPACECRAFT SPIN AXIS
NLP    CODE OF LAUNCH PLANET
NBOD   NUMBER OF BODIES USED IN VIRTUAL MASS PROGRAM
NB(11) CODES OF PLANETS
NTP    CODE OF TARGET PLANET
INPR   PRINT INCREMENTS (IN INCREMENTS)
IPROB  PROBLEM NUMBER
ISPH   SPHERE OF INFLUENCE CODE
       =0  SPHERE OF INFLUENCE NOT INTERSECTED
       =1  SPHERE OF INFLUENCE ALREADY ENCOUNTERED
INCMT  TOTAL INCREMENTS USED
IEPHEM EPHEMERIS CODE
ICL    CLOSEST APPROACH CODE
       =0  CLOSEST APPROACH NOT ENCOUNTERED
       =1  CLOSEST APPROACH ALREADY ENCOUNTERED
IPRINT PRINT CODE
       =0  OUTPUT INITIAL AND FINAL DATA
       =1  DO NOT OUTPUT INITIAL AND FINAL DATA
ICL2   CLOSEST APPROACH TERMINATION CODE
       =0  DO NOT STOP AT CLOSEST APPROACH
       =1  STOP AT CLOSEST APPROACH
### SOLVE-FOR PARAMETER LABELS
- XSL(24)
- XU(8)
- XV(15)

### DYNAMIC CONSIDER PARAMETER LABELS
- X'AB(6)

### MEASUREMENT CONSIDER PARAMETER LABELS
- XNM(24)

### VEHICLE POSITION/ VELOCITY VECTOR Component NAMES
- KPRINT

---

### ZERO ITERATE VECTOR
- ZDAT(6)
- RP
- FI
- PSI1
- PSI2
- TIM1
- TIM2
- THELS
- PHILS
- TI
- TF
- THEDOT
- RPRAT

### PARKING ORBIT RADIUS
### INJECTION TRUE ANOMALY
### ANGLE OF FIRST BURN
### ANGLE OF SECOND BURN
### TIME INTERVAL OF FIRST BURN
### TIME INTERVAL OF SECOND BURN
### LONGITUDE OF LAUNCH SITE
### LATITUDE OF LAUNCH SITE
### NOT USED
### NOT USED
### ROTATION RATE OF LAUNCH PLANET
### PARKING ORBIT INVERSE RATE
<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>NOMINAL LAUNCH AZIMUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>IZERO</td>
<td>ZERO ITERATION FLAG</td>
</tr>
<tr>
<td></td>
<td>=0 INITIAL STATE READ IN</td>
</tr>
<tr>
<td></td>
<td>=1 PLANET-TO-PLANET</td>
</tr>
<tr>
<td></td>
<td>=2 PLANET-TO-POINT</td>
</tr>
<tr>
<td></td>
<td>=3 POINT-TO-PLANET</td>
</tr>
<tr>
<td></td>
<td>=4 POINT-TO-POINT</td>
</tr>
<tr>
<td></td>
<td>=10 LUNAR TARGETING</td>
</tr>
<tr>
<td>KOAST</td>
<td>PARKING ORBIT INDICATOR</td>
</tr>
<tr>
<td></td>
<td>=-1 SHORT COAST</td>
</tr>
<tr>
<td></td>
<td>=1 LONG COAST</td>
</tr>
<tr>
<td>LTARG</td>
<td>TYPE OF MISSION FOR TARGETING</td>
</tr>
<tr>
<td></td>
<td>=0 INTERPLANETARY MISSION</td>
</tr>
<tr>
<td></td>
<td>=1 LUNAR MISSION</td>
</tr>
</tbody>
</table>

| VHPM            | MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY AT TARGET BODY (VHP VECTOR) |
|                 |                                                                   |
| DPA             | DECLINATION OF VHP                                                |
| RAP             | RIGHT ASCENSION OF VHP                                            |
5.2 COMMON VARIABLES IN ALPHABETICAL ORDER

In this section all variables appearing in common are listed and defined in alphabetical order. The second field serves to identify the block in which the variable appears.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(2,3) PBLK</td>
<td>FTA IMPACT PLANE TRANSFORMATION MATRIX</td>
</tr>
<tr>
<td>AALP(20) SIMCNT</td>
<td>ACTUAL ERROR IN POINTING ANGLE 1</td>
</tr>
<tr>
<td>AAUX(3) TAREAL</td>
<td>ACTUAL AUXILIARY VALUES OF ITERATE</td>
</tr>
<tr>
<td>ABET(20) SIMCNT</td>
<td>ACTUAL ERROR IN POINTING ANGLE 2</td>
</tr>
<tr>
<td>AC(5,10) TAREAL</td>
<td>ACCURACY LEVELS (UP TO 5) USED IN EACH GUIDANCE EVENT</td>
</tr>
<tr>
<td>ACC</td>
<td>MISC</td>
</tr>
<tr>
<td>ACC1</td>
<td>SIM2</td>
</tr>
<tr>
<td>ACCND</td>
<td>TRAJCD</td>
</tr>
<tr>
<td>ACKT</td>
<td>TAREAL</td>
</tr>
<tr>
<td>ADA(3,6) BAIM</td>
<td>VARIATION MATRIX</td>
</tr>
<tr>
<td>ADEVSB(24) SIM1</td>
<td>ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS AT TRAJECTORY BEGINNING</td>
</tr>
<tr>
<td>ADEVX(6) SIM1</td>
<td>ACTUAL DEVIATION IN THE STATE VECTOR</td>
</tr>
<tr>
<td>ADEVXB(6) SIM1</td>
<td>ACTUAL DEVIATION IN STATE VECTOR AT BEGINNING OF TRAJECTORY</td>
</tr>
<tr>
<td>ADEVXS(24) SIM1</td>
<td>ACTUAL DEVIATION IN SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>AINC7 EVENT</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>AK(6,4) STM</td>
<td>KALMAN GAIN CONSTANT FOR POSITION VELOCITY STATE</td>
</tr>
<tr>
<td>AL((4,15))</td>
<td>STM</td>
</tr>
<tr>
<td>ALNGLTH</td>
<td>VM</td>
</tr>
<tr>
<td>AM((4,24))</td>
<td>STM</td>
</tr>
<tr>
<td>ANODE7</td>
<td>EVENT</td>
</tr>
<tr>
<td>ANG</td>
<td>IMPTAR</td>
</tr>
<tr>
<td>ANOIS((4))</td>
<td>SIM1</td>
</tr>
<tr>
<td>APRO((20))</td>
<td>SIMCNT</td>
</tr>
<tr>
<td>AR((4,4))</td>
<td>SIM1</td>
</tr>
<tr>
<td>ARES((20))</td>
<td>SIMCNT</td>
</tr>
<tr>
<td>ATAR((3))</td>
<td>TAREAL</td>
</tr>
<tr>
<td>ATRANS((6))</td>
<td>BAIM</td>
</tr>
<tr>
<td>AVARM((12))</td>
<td>SIMCNT</td>
</tr>
<tr>
<td>AY((4))</td>
<td>SIM1</td>
</tr>
<tr>
<td>AYHEV((4))</td>
<td>SIM1</td>
</tr>
<tr>
<td>B</td>
<td>VM</td>
</tr>
<tr>
<td>BCON((3))</td>
<td>LUNARY</td>
</tr>
<tr>
<td>BDR</td>
<td>VM</td>
</tr>
<tr>
<td>BDT</td>
<td>VM</td>
</tr>
<tr>
<td>BIA((12))</td>
<td>MISC</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BSI1</td>
<td>TRJ</td>
</tr>
<tr>
<td>BSI2</td>
<td>TRJ</td>
</tr>
<tr>
<td>BSI3</td>
<td>TRJ</td>
</tr>
<tr>
<td>B0TS1</td>
<td>TRJ</td>
</tr>
<tr>
<td>BDTS1</td>
<td>TRJ</td>
</tr>
<tr>
<td>BDTS2</td>
<td>TRJ</td>
</tr>
<tr>
<td>B0RS1</td>
<td>TRJ</td>
</tr>
<tr>
<td>B0RS2</td>
<td>TRJ</td>
</tr>
<tr>
<td>B0RS3</td>
<td>TRJ</td>
</tr>
<tr>
<td>C1</td>
<td>LUNART</td>
</tr>
<tr>
<td>CAINC</td>
<td>VM</td>
</tr>
<tr>
<td>CH(3,3)</td>
<td>TMTRX</td>
</tr>
<tr>
<td>CN(30)</td>
<td>NAME</td>
</tr>
<tr>
<td>CN(80)</td>
<td>BLK</td>
</tr>
<tr>
<td>CR</td>
<td>PBLK</td>
</tr>
<tr>
<td>CTOL</td>
<td>TAREAL</td>
</tr>
<tr>
<td>CXSU</td>
<td>STM</td>
</tr>
<tr>
<td>CSUB</td>
<td>STM</td>
</tr>
<tr>
<td>CXUG</td>
<td>GUI</td>
</tr>
<tr>
<td>CSUP</td>
<td>GUI</td>
</tr>
</tbody>
</table>
CXSV(24,15)  STM  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS

CXSVB(24,15)  STM  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS AT INITIAL TIME

CXSVG(24,15)  GUI  CONTROL CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS

CXSVP(24,15)  STM  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

CXU(6,8)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS

CXUB(6,8)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS AT INITIAL TIME

CXUG(6,8)  GUI  CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS

CXUP(6,8)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT

CXV(6,15)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS

CXVB(6,15)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS AT INITIAL TIME

CXVG(6,15)  GUI  CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS

CXVP(6,15)  STM  CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS PRIOR TO PROCESSING A MEASUREMENT
<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXXS(6,24)</td>
<td>STM</td>
<td>CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>CXXSB(6,24)</td>
<td>STM</td>
<td>CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS AT INITIAL TIME</td>
</tr>
<tr>
<td>CXXSG(6,24)</td>
<td>GUI</td>
<td>CONTROL CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>CXXSP(6,24)</td>
<td>STM</td>
<td>CORRELATION BETWEEN POSITION/VELOCITY STATE AND SOLVE-FOR PARAMETERS PRIOR TO PROCESSING A MEASUREMENT</td>
</tr>
<tr>
<td>D1</td>
<td>TAREAL</td>
<td>JULIAN DATE ASSOCIATED WITH RIN ARRAY</td>
</tr>
<tr>
<td>DAB</td>
<td>SIMCNT</td>
<td>BIAS IN SEMI-MAJOR AXIS OF TARGET PLANET</td>
</tr>
<tr>
<td>DATEJ</td>
<td>TIME</td>
<td>JULIAN DATE OF INITIAL TRAJECTORY TIME (REFERENCED TO 1950)</td>
</tr>
<tr>
<td>DAUX(3)</td>
<td>TAREAL</td>
<td>DESIRED AUXILIARY PARAMETER VALUES OF ITERATE</td>
</tr>
<tr>
<td>DC</td>
<td>VM</td>
<td>JULIAN DATE, EPOCH 1900, AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>DEB</td>
<td>SIMCNT</td>
<td>BIAS IN ECCENTRICITY OF TARGET PLANET</td>
</tr>
<tr>
<td>DECLIN</td>
<td>LUNART</td>
<td>DECLINATION OF APPROACH ASYMPTOTE WITH RESPECT TO LUNAR EQUATOR</td>
</tr>
<tr>
<td>DELAXS</td>
<td>CONST3</td>
<td>TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELECC</td>
<td>CONST3</td>
<td>TARGET PLANET ECCENTRICITY FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELICL</td>
<td>CONST3</td>
<td>TARGET PLANET INCLINATION FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMA</td>
<td>CONST3</td>
<td>TARGET PLANET MEAN ANOMALY FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMUP</td>
<td>CONST3</td>
<td>TARGET PLANET GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
<tr>
<td>DELMUS</td>
<td>CONST3</td>
<td>SUN GRAVITATIONAL CONSTANT FACTOR USED IN NUMERICAL DIFFERENCING</td>
</tr>
</tbody>
</table>
DELNOOD  CONST3  TARGET PLANET LONGITUDE OF THE ASCENDING
        NODE FACTOR USED IN NUMERICAL DIFFERENCING.
DELPX(3)  EXE   VELOCITY CORRECTION TO BE MODELED AS AN
            IMPULSE SERIES.
DELTAT   TAREAL  NUMBER OF DAYS INTEGRATION IS TO CONTINUE
            IF NO OTHER STOPPING CONDITION OCCURS.
DELTAV(3) TAREAL  CORRECTIONS TO BE ADDED TO VELOCITY
            COMPONENTS FOR NEXT ITERATION.
DELTH    VM     INCREMENT IN TRUE ANOMALY USED.
DELTM    TIME   TIME INCREMENT.
DELP     VM     PRINT INCREMENTS (IN DAYS).
DELV(3,10) TAREAL  EXTERNALLY SUPPLIED VELOCITY CORRECTION
            OR VELOCITY INCREMENT COMPUTED BY
            INSERTION.
DELV(3,10) PBLK   ARRAY OF EXTERNALLY-SUPPLIED VELOCITY
            CHANGES.
DELW     CONST3  TARGET PLANET ARGUMENT OF PERIAPSIS FACTOR
            USED IN NUMERICAL DIFFERENCING.
DG(I0)   TAREAL  JULIAN DATES OF EVENT TIMES.
DIB      SIMCNT  BIAS IN INCLINATION OF TARGET PLANET.
DINTG(I0)  TAREAL  NOT USED IN CURRENT TARGET VERSION.
DIPX     EXE    JULIAN DATE TRANSFERRED TO EXCUT OR EXCUTS.
DMAB     SIMCNT  BIAS IN MEAN ANOMALY.
DMUPB    SIMCNT  BIAS IN GRAVITATIONAL CONSTANT OF TARGET
            PLANET.
DMUSB     SIMCNT  BIAS IN GRAVITATIONAL CONSTANT OF SUN.
DNCN(I3)  CONST  CONSTANTS FROM WHICH DYNAMIC NOISE IS
            COMPUTED.
DNOB     SIMCNT  BIAS IN LONGITUDE OF ASCENDING NODE.
DPA      ZOUT   DECLINATION OF VHP.

<table>
<thead>
<tr>
<th>DSI</th>
<th>VM</th>
<th>JULIAN DATE, EPOCH 1900, AT SPHERE OF INFLUENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT(10)</td>
<td>TAREAL</td>
<td>JULIAN DATES OF TARGET TIMES</td>
</tr>
<tr>
<td>DTAR(3)</td>
<td>TAREAL</td>
<td>DESIRED TARGET VALUES OF ITERATE</td>
</tr>
<tr>
<td>DTAR(3)</td>
<td>LUNART</td>
<td>TARGET VALUES OF SMA,B.T, AND B.R IN LUNAR TARGETING</td>
</tr>
<tr>
<td>DTI</td>
<td>PULS</td>
<td>TIME INTERVAL (DAYS) BETWEEN SUCCESSIVE PULSES</td>
</tr>
<tr>
<td>DTIME</td>
<td>OVERL</td>
<td>TIME INTERVAL BETWEEN ORBITAL INSERTION DECISION AND EXECUTION</td>
</tr>
<tr>
<td>DTHAX</td>
<td>TRAJCO</td>
<td>MAXIMUM TIME INCREMENT FOR WHICH ISTMC IS VALID</td>
</tr>
<tr>
<td>DTPLAN</td>
<td>TRAJCO</td>
<td>STATE TRANSITION INTEGRATION INTERVAL WHEN TARGET PLANET IS CENTRAL BODY AND ISTM1=1</td>
</tr>
<tr>
<td>DTSUN</td>
<td>TRAJCO</td>
<td>STATE TRANSITION INTEGRATION INTERVAL WHEN THE SUN IS CENTRAL BODY AND ISTM1=1</td>
</tr>
<tr>
<td>DUMMYQ(4)</td>
<td>EXE</td>
<td>ARRAY OF EXECUTION ERROR VARIANCES</td>
</tr>
<tr>
<td>DUR</td>
<td>PULS</td>
<td>DURATION OF SINGLE PULSE</td>
</tr>
<tr>
<td>DV8(3)</td>
<td>EVENT</td>
<td>ORBIT INSERTION VARIABLE, NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>DVF(3)</td>
<td>PULS</td>
<td>VELOCITY INCREMENT ADDED ON FINAL PULSE</td>
</tr>
<tr>
<td>DVI(3)</td>
<td>PULS</td>
<td>VELOCITY INCREMENT ADDED ON TYPICAL PULSE</td>
</tr>
<tr>
<td>DVMAX(10)</td>
<td>TAREAL</td>
<td>MAXIMUM ALLOWABLE CHANGE IN ANY VELOCITY COMPONENT FOR EACH EVENT</td>
</tr>
<tr>
<td>DVN(3)</td>
<td>PBLK</td>
<td>COMMANDED VELOCITY CORRECTION TRANSFERRED TO BIAM</td>
</tr>
<tr>
<td>DVRB(3)</td>
<td>PBLK</td>
<td>VELOCITY CORRECTION REQUIRED TO REMOVE AIMPOINT BIAS</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>OVUP(3)</td>
<td>PBLK</td>
<td>UPDATE VELOCITY CORRECTION</td>
</tr>
<tr>
<td>DWB</td>
<td>SIMC</td>
<td>BIAS IN ARGUMENT OF PERIAPSIS</td>
</tr>
<tr>
<td>ECC7</td>
<td>EVENT</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>EDEVSM(24)</td>
<td>SIM1</td>
<td>ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS (FOR ADAPTIVE FILTERING)</td>
</tr>
<tr>
<td>EDEVX(6)</td>
<td>SIM1</td>
<td>ESTIMATED DEVIATION IN THE STATE VECTOR</td>
</tr>
<tr>
<td>EDEVXM(6)</td>
<td>SIM1</td>
<td>ESTIMATED DEVIATION IN THE STATE VECTOR (FOR ADAPTIVE FILTERING)</td>
</tr>
<tr>
<td>EDEVXS(24)</td>
<td>SIM1</td>
<td>ESTIMATED DEVIATION IN SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>EIGHT</td>
<td>DPNUM</td>
<td>THE NUMBER EIGHT (8) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY</td>
</tr>
<tr>
<td>ELMNT(80)</td>
<td>BLK</td>
<td>CONTAINS THE ORBITAL ELEMENTS OF THE PLANETS (SEE LARGE ARRAY DEFNS IN SECTION 5.3)</td>
</tr>
<tr>
<td>EM(2,6)</td>
<td>GUI</td>
<td>VARIATION MATRIX RELATING POSITION /VELOCITY DEVIATIONS TO B.T AND B.R DEVIATIONS</td>
</tr>
<tr>
<td>EM1</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-1 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>EM2</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-2 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>EM3</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-3 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>EM4</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-4 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>EM5</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-5 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>EM6</td>
<td>DPNUM</td>
<td>THE NUMBER 1.E-6 TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>EM7</td>
<td>DPNUM THE NUMBER 1.E-7 TO NINE SIGNIFICANT FIGURES</td>
<td></td>
</tr>
<tr>
<td>EM8</td>
<td>DPNUM THE NUMBER 1.E-8 TO NINE SIGNIFICANT FIGURES</td>
<td></td>
</tr>
<tr>
<td>EM9</td>
<td>DPNUM THE NUMBER 1.E-9 TO NINE SIGNIFICANT FIGURES</td>
<td></td>
</tr>
<tr>
<td>EM13</td>
<td>DPNUM THE NUMBER 1.E-13 TO NINE SIGNIFICANT FIGURES</td>
<td></td>
</tr>
<tr>
<td>EM50</td>
<td>DPNUM THE NUMBER 1.E-50 TO NINE SIGNIFICANT FIGURES</td>
<td></td>
</tr>
<tr>
<td>EMN(15)</td>
<td>BLK THE CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE MOON</td>
<td></td>
</tr>
<tr>
<td>EMU</td>
<td>LUNART GRAVITATIONAL CONSTANT OF EARTH (KM3/SEC2)</td>
<td></td>
</tr>
<tr>
<td>EPS</td>
<td>CONST OBLIQUITY OF EARTH</td>
<td></td>
</tr>
<tr>
<td>EQLQ(3,3)</td>
<td>TAREAL TRANSFORMATION MATRIX FROM EARTH-EQUATORIAL TO LUNAR EQUATORIAL COORDINATES</td>
<td></td>
</tr>
<tr>
<td>EVNM(11)</td>
<td>NAME EVENT NAME</td>
<td></td>
</tr>
<tr>
<td>EXEC(3,3)</td>
<td>PBLK EXECUTION ERROR COVARIANCE MATRIX</td>
<td></td>
</tr>
<tr>
<td>EY(4)</td>
<td>SIM1 ESTIMATED MEASUREMENT</td>
<td></td>
</tr>
<tr>
<td>F(44,4)</td>
<td>COM CONTAINS THE POSITIONS AND VELOCITIES OF THE PLANETS AT A SPECIFIED TIME PLUS THE POSITIONS AND VELOCITIES OF THE SPACECRAFT RELATIVE TO THE PLANETS</td>
<td></td>
</tr>
<tr>
<td>FAC(3)</td>
<td>TAREAL SCALING FACTORS USED IN BAD STEP CHECK</td>
<td></td>
</tr>
<tr>
<td>FACP</td>
<td>MISC POSITION FACTOR USED IN NUMERICAL DIFFERENCING</td>
<td></td>
</tr>
<tr>
<td>FACV</td>
<td>MISC VELOCITY FACTOR USED IN NUMERICAL DIFFERENCING</td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>ZERDAT INJECTION TRUE ANOMALY</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>FIVE</td>
<td>DPNUM</td>
<td></td>
</tr>
<tr>
<td>FNTH</td>
<td>TIME</td>
<td></td>
</tr>
<tr>
<td>FOP</td>
<td>CONST2</td>
<td></td>
</tr>
<tr>
<td>FOUR</td>
<td>DPNUM</td>
<td></td>
</tr>
<tr>
<td>FOV</td>
<td>CONST2</td>
<td></td>
</tr>
<tr>
<td>FS(2,5)</td>
<td>PULS</td>
<td></td>
</tr>
<tr>
<td>G(4,8)</td>
<td>STM</td>
<td></td>
</tr>
<tr>
<td>GA(3,6)</td>
<td>OVERZ</td>
<td></td>
</tr>
<tr>
<td>GG(3)</td>
<td>PULS</td>
<td></td>
</tr>
<tr>
<td>GS(2,4)</td>
<td>PULS</td>
<td></td>
</tr>
<tr>
<td>H(4,6)</td>
<td>STM</td>
<td></td>
</tr>
<tr>
<td>HALF</td>
<td>DPNUM</td>
<td></td>
</tr>
<tr>
<td>HP7</td>
<td>EVENT</td>
<td></td>
</tr>
<tr>
<td>HPHR(4,4)</td>
<td>STM</td>
<td></td>
</tr>
<tr>
<td>IAMNF</td>
<td>SIMCNT</td>
<td></td>
</tr>
<tr>
<td>IAUGIN(24)</td>
<td>STVEC</td>
<td></td>
</tr>
<tr>
<td>Iaug(24)</td>
<td>STVEC</td>
<td></td>
</tr>
<tr>
<td>IAUGDC(8)</td>
<td>STVEC</td>
<td></td>
</tr>
</tbody>
</table>

THE NUMBER FIVE (5) TO NINE SIGNIFICANT FIGURES, TARGETING MODE ONLY

FINAL TRAJECTORY TIME

OFF-DIAGONAL ANNIHILATION VALUE FOR POSITION EIGENVALUES

OFF-DIAGONAL ANNIHILATION VALUE FOR VELOCITY EIGENVALUES

F-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES

OBSERVATION MATRIX RELATING OBSERVABLES TO DYNAMIC CONSIDER PARAMETER STATE

GUIDANCE MATRIX

GRAVITATIONAL CONSTANTS OF SUN, LAUNCH, AND TARGET BODIES

G-SERIES COEFFICIENTS OF LAUNCH AND TARGET BODIES

OBSERVATION MATRIX RELATING OBSERVABLES TO POSITION/VELOCITY STATE

THE NUMBER ONE-HALF (1/2) TO NINE SIGNIFICANT FIGURES

ORBIT INSERTION VARIABLES, NON-FUNCTIONAL IN EXISTING PROGRAM

NON-FUNCTIONAL IN PRESENT ERROR ANALYSIS PROGRAM

ACTUAL MEASUREMENT NOISE FLAG

INPUT AUGMENTATION VECTOR OF ONE'S AND ZERO'S

AUGMENTATION VECTOR

DYNAMIC CONSIDER AUGMENTATION VECTOR
<table>
<thead>
<tr>
<th>IAUGMC(15)</th>
<th>STVEC</th>
<th>MEASUREMENT CONSIDER AUGMENTATION VECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBAD</td>
<td>TARINT</td>
<td>BAD STEP FLAG FOR CURRENT ACCURACY LEVEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=1, DO NOT CHECK FOR BAD STEP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=2, CHECK FOR BAD STEP</td>
</tr>
<tr>
<td>IBAOS(10)</td>
<td>TARINT</td>
<td>BAD STEP FLAGS FOR EACH TARGETING EVENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=1 NEVER USE BAD STEP CHEQUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=2 USE BAD STEP CHECK AT FINAL LEVEL ONLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=3 USE BAD STEP CHECK AT ALL LEVELS</td>
</tr>
<tr>
<td>IBAG</td>
<td>BAIH</td>
<td>NOT USED</td>
</tr>
<tr>
<td>IBARY</td>
<td>CNTRIC</td>
<td>REFERENCE COORDINATE SYSTEM CODE</td>
</tr>
<tr>
<td>IBAST</td>
<td>TARINT</td>
<td>BAD STEP CHECK INDICATOR FOR CURRENT EVENT</td>
</tr>
<tr>
<td>IBIAS</td>
<td>PBLK</td>
<td>BIASED AIMPOINT GUIDANCE EVENT FLAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 0 AIMPOINT NOT BIASED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=1 AIMPOINT BIASED</td>
</tr>
<tr>
<td>ICA1</td>
<td>TRJ</td>
<td>CLOSEST APPROACH CODE FOR ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>ICA2</td>
<td>TRJ</td>
<td>CLOSEST APPROACH CODE FOR MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>ICA3</td>
<td>TRJ</td>
<td>CLOSEST APPROACH CODE FOR ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>ICDQ3(20)</td>
<td>EVENT</td>
<td>ARRAY OF CODES WHICH DETERMINE WHICH EXECUTION POLICIES ARE TO BE USED IN GUIDANCE EVENTS</td>
</tr>
<tr>
<td>ICOT3(20)</td>
<td>EVENT</td>
<td>CODES WHICH DETERMINE WHICH GUIDANCE POLICIES ARE BEING USED</td>
</tr>
<tr>
<td>ICL</td>
<td>VM</td>
<td>CLOSEST APPROACH CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=0 CLOSEST APPROACH NOT ENCOUNTERED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=1 CLOSEST APPROACH ENCOUNTERED</td>
</tr>
<tr>
<td>ICL2</td>
<td>VM</td>
<td>CLOSEST APPROACH TERMINATION CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=0 DO NOT STOP AT CLOSEST APPROACH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=1 STOP AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>ICODE</td>
<td>OVER1</td>
<td>MEASUREMENT CODE</td>
</tr>
<tr>
<td>ICOOR</td>
<td>MISC</td>
<td>CODE TO DETERMINE WHICH COORDINATE SYSTEM THE INITIAL STATE VECTOR IS INPUT</td>
</tr>
<tr>
<td>CODE</td>
<td>MEANING</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ICOORD</td>
<td>NON-FUNCTIONAL IN ERROR ANALYSIS MODE</td>
<td></td>
</tr>
<tr>
<td>IDENS</td>
<td>PROBABILITY DENSITY FUNCTION CODE, NON-FUNCTIONAL</td>
<td></td>
</tr>
<tr>
<td>IDNF</td>
<td>DYNAMIC NOISE FLAG</td>
<td></td>
</tr>
<tr>
<td>IEIG</td>
<td>CODE USED TO DECIDE IF BOTH POSITION AND VELOCITY EIGENVECTORS ARE REQUESTED</td>
<td></td>
</tr>
<tr>
<td>IEPHEM</td>
<td>EPHMERIS CODE</td>
<td></td>
</tr>
<tr>
<td>IEVNT(50)</td>
<td>EVENT CODES OF EVENTS</td>
<td></td>
</tr>
<tr>
<td>IFINT(10)</td>
<td>NOT USED IN THIS TRAJECTORY VERSION</td>
<td></td>
</tr>
<tr>
<td>IGP</td>
<td>MIDCOURSE GUIDANCE POLICY CODE</td>
<td></td>
</tr>
<tr>
<td>IGUID(5,10)</td>
<td>ARRAY OF GUIDANCE EVENT CODES</td>
<td></td>
</tr>
<tr>
<td>IMYP1</td>
<td>HYPERELLIPSOID CODE USED TO DETERMINE IF K=1, K=3, OR BOTH</td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>GUIDANCE EVENT COUNTER</td>
<td></td>
</tr>
<tr>
<td>IIIGP</td>
<td>MIDCOURSE GUIDANCE POLICY CODE</td>
<td></td>
</tr>
<tr>
<td>IIIPOL</td>
<td>CODE WHICH DETERMINES IF EITHER TWO -VARIBLE OR THREE-VARIABLE I-PLANE GUIDANCE POLICY HAS OCCURRED</td>
<td></td>
</tr>
<tr>
<td>IMNF</td>
<td>MEASUREMENT NOISE FLAG</td>
<td></td>
</tr>
<tr>
<td>INC</td>
<td>DETERMINE WHETHER THE ABOVE OPTION IS TO BE USED</td>
<td></td>
</tr>
<tr>
<td>INCMNT</td>
<td>NUMBER OF INCREMENTS USED</td>
<td></td>
</tr>
<tr>
<td>INCMNT</td>
<td>TOTAL INCREMENTS USED</td>
<td></td>
</tr>
<tr>
<td>INCPR</td>
<td>SPECIFIES AFTER HOW MANY TIME INCREMENTS PRINT-OUT IS TO OCCUR</td>
<td></td>
</tr>
<tr>
<td>INITIAL</td>
<td>NON-FUNCTIONAL IN ERROR ANALYSIS MODE</td>
<td></td>
</tr>
<tr>
<td>INPR</td>
<td>PRINT INCREMENTS (IN INCREMENTS)</td>
<td></td>
</tr>
<tr>
<td>INPX</td>
<td>IMP - SERIES CODE</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IOP7</td>
<td>EVENT</td>
<td>ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAMS</td>
</tr>
<tr>
<td>I1G</td>
<td>COM</td>
<td>PAGE NUMBER</td>
</tr>
<tr>
<td>I1HASE</td>
<td>TARINT</td>
<td>PHASE COUNTER FOR CURRENT TARGETING EVENT</td>
</tr>
<tr>
<td>I1POL</td>
<td>EVENT</td>
<td>CODE WHICH DETERMINES IF FIXED-TIME-OF-ARRIVAL GUIDANCE EVENT HAS OCCURRED</td>
</tr>
<tr>
<td>I1PQ</td>
<td>BAIR</td>
<td>NOT USED</td>
</tr>
<tr>
<td>I1PR</td>
<td>COM</td>
<td>A CODE WHICH DETERMINES IF PRINT-OUT IS TO OCCUR AFTER A SPECIFIED NUMBER OF DAYS</td>
</tr>
<tr>
<td>I1PRE</td>
<td>PRINT</td>
<td>CONTROLS INITIALIZATION OF PROGRAM CONSTANTS IN SUBROUTINE -PRELIM-.</td>
</tr>
<tr>
<td>I1PRINT</td>
<td>VM</td>
<td>PRINT CODE</td>
</tr>
<tr>
<td>I1PROB</td>
<td>VM</td>
<td>PROBLEM NUMBER</td>
</tr>
<tr>
<td>I1PR(4)</td>
<td>COM</td>
<td>SPECIFIES PRINT OPTIONS (NOT APPLICABLE TO STEEP TRAJECTORY)</td>
</tr>
<tr>
<td>I15011</td>
<td>TRJ</td>
<td>SPHERE OF INFLUENCE CODE FOR ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>I15012</td>
<td>TRJ</td>
<td>SPHERE OF INFLUENCE CODE FOR MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>I15013</td>
<td>TRJ</td>
<td>SPHERE OF INFLUENCE CODE FOR ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>I1SP2</td>
<td>MISC</td>
<td>SPHERE OF INFLUENCE FLAG</td>
</tr>
<tr>
<td>I1SPH</td>
<td>VM</td>
<td>SPHERE OF INFLUENCE CODE</td>
</tr>
<tr>
<td>I1START</td>
<td>TARINT</td>
<td>STAGE OF INITIAL TARGETING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I1START Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NO TARGETING STARTED</td>
</tr>
<tr>
<td>1</td>
<td>FIRST PHASE STARTED - HAVE TARG MATRIX</td>
</tr>
<tr>
<td>2</td>
<td>SECOND PHASE STARTED - HAVE TARG MATRIX</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>ISTM1</td>
<td>TRJCD</td>
</tr>
<tr>
<td>ISTMC</td>
<td>TRJCD</td>
</tr>
<tr>
<td>I-TOP</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ITAG</td>
<td>LUNAR</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ITARM</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ITR</td>
<td>MISC</td>
</tr>
<tr>
<td>ITRAT</td>
<td>COM</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td>OVERX</td>
</tr>
<tr>
<td>IZERO</td>
<td>ZERDAY</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>JX</td>
<td>OVERX</td>
</tr>
<tr>
<td>KAXTAR(3)</td>
<td>TARINT</td>
</tr>
<tr>
<td>KEYTAR(3)</td>
<td>TARINT</td>
</tr>
<tr>
<td>KGYD(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>KL</td>
<td>COM</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>KMXQ(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>KCOAST</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>KCOUNT</td>
<td>COM</td>
</tr>
<tr>
<td>KPRINT</td>
<td>XXXL</td>
</tr>
<tr>
<td>KSICA</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>KTAR(6,10)</td>
<td>TARINT</td>
</tr>
<tr>
<td>KTIM(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>KTYPE(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>KMXQ(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>KUR</td>
<td>TARINT</td>
</tr>
<tr>
<td>KMIT</td>
<td>TARINT</td>
</tr>
<tr>
<td>LEV</td>
<td>TARINT</td>
</tr>
<tr>
<td>LEVELS</td>
<td>TARINT</td>
</tr>
<tr>
<td>LINCT</td>
<td>COM</td>
</tr>
<tr>
<td>LNPGE</td>
<td>COM</td>
</tr>
<tr>
<td>LKLP(10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>LKTAR(6,10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>LKTP(10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>LLVLS(10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>LNPTR(10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>LARG</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>LVLS</td>
<td>TARINT</td>
</tr>
<tr>
<td>MAT(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td>MATX</td>
<td>TARINT</td>
</tr>
<tr>
<td>MAXB(10)</td>
<td>TARINT</td>
</tr>
<tr>
<td>MAXBAD</td>
<td>TARINT</td>
</tr>
<tr>
<td>MCHTR</td>
<td>MEAS</td>
</tr>
<tr>
<td>MCODE(1000)</td>
<td>MEAS</td>
</tr>
</tbody>
</table>
| MDL(10) | TARINT | EXECUTION MODELS FOR EACH GUIDANCE EVENT  
|         |        | =1 IMPULSIVE  
|         |        | =2 PULSING ARC  
| MNCN(12) | CONST | MEASUREMENT NOISE CONSTANTS  
| MNNAME(12,3) | NAME | MEASUREMENT NAME  
| MONTH(12) | PRT | NAMES OF MONTHS  
| NAE | EVENT | ADAPTIVE FILTERING EVENTS HAVING OCCURRED.  
| NAF6(20) | EVENT | ARRAY OF ADAPTIVE FILTERING EVENT CODES - NON-FUNCTIONAL IN EXISTING PROGRAM  
| NAFC | OVER1 | ADAPTIVE FILTER FLAG  
| NB(11) | VM | CODES OF PLANETS  
| NB1(11) | SIM2 | ARRAY OF PLANET CODES IN ACTUAL TRAJECTORY  
| NBOD | VM | NUMBER OF BODIES USED IN VIRTUAL MASS PROGRAM  
| NBOD1 | SIM2 | NUMBER OF BODIES IN ACTUAL TRAJECTORY  
| NBODY | COM | EQUAL TO 4*NBODYI-3  
| NBODYI | COM | NUMBER OF BODIES CONSIDERED IN VIRTUAL MASS TRAJECTORY  
| NCPR | TARINT | NUMBER OF INTEGRATION INCREMENTS BETWEEN PRINT-OUTS OF NOMINAL TRAJECTORY  
| NDACC | TRAJCD | NUMERICAL DIFFERENCING ACCURACY CODE  
| NDIM1 | STVEC | DIMENSION OF SOLVE-FOR PARAMETER STATE  
| NDIM2 | STVEC | DIMENSION OF DYNAMIC CONSIDER STATE  
| NDIM3 | STVEC | DIMENSION OF MEASUREMENT CONSIDER PARAMETER STATE  
| NEV | EVENT | NUMBER OF EVENTS  
| NEV1 | EVENT | TOTAL NUMBER OF EIGENVECTOR EVENTS  
| NEV2 | EVENT | TOTAL NUMBER OF PREDICTION EVENTS  

61
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEV3</td>
<td>EVENT TOTAL NUMBER OF GUIDANCE EVENTS</td>
</tr>
<tr>
<td>NEV4</td>
<td>EVENT TOTAL NUMBER OF -COMCON- EVENTS</td>
</tr>
<tr>
<td>NEV5</td>
<td>EVENT TOTAL NUMBER OF QUASI-LINEAR FILTERING EVENTS</td>
</tr>
<tr>
<td>NEV6</td>
<td>EVENT TOTAL NUMBER OF ADAPTIVE FILTERING EVENTS. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV7</td>
<td>EVENT ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV8</td>
<td>EVENT ORBIT INSERTION VARIABLE. NON-FUNCTIONAL IN EXISTING PROGRAMS</td>
</tr>
<tr>
<td>NEV9</td>
<td>EVENT ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV10</td>
<td>EVENT ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NEV11</td>
<td>EVENT ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>NGE</td>
<td>EVENT NUMBER OF GUIDANCE EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>NINETY</td>
<td>OPNUM THE NUMBER NINETY (90) TO NINE SIGNIFICANT FIGURES. TARGETING MODE ONLY</td>
</tr>
<tr>
<td>NITS TARINT</td>
<td>ALLOWABLE NUMBER OF ITERATIONS FOR CURRENT EVENT</td>
</tr>
<tr>
<td>NLP VM</td>
<td>CODE OF LAUNCH PLANET</td>
</tr>
<tr>
<td>NMM MEAS</td>
<td>TOTAL NUMBER OF MEASUREMENTS</td>
</tr>
<tr>
<td>NO(11) BLK</td>
<td>AN ARRAY OF PLANET CODES BEING USED TO GENERATE THE VIRTUAL MASS TRAJECTORY</td>
</tr>
<tr>
<td>NOGD TARINT</td>
<td>TOTAL NUMBER OF GUIDANCE EVENTS</td>
</tr>
<tr>
<td>NOIT(10) TARINT</td>
<td>THE NUMBER OF TOTAL ITERATIONS ALLOWED AT THE FIRST AND LAST LEVELS OF TARGETING EVENTS FOR EACH GUIDANCE EVENT</td>
</tr>
<tr>
<td>NOPAR TARINT</td>
<td>NUMBER OF TARGET PARAMETERS FOR CURRENT EVENT</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NPHAS</td>
<td>NUMBER OF TARGETING PHASES FOR CURRENT EVENT</td>
</tr>
<tr>
<td>NOSO1</td>
<td>OUTER TARGETING FLAG =0 NORMAL TARGETING =1 OUTER TARGETING</td>
</tr>
<tr>
<td>NPE</td>
<td>NUMBER OF PREDICTION EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>NPUL</td>
<td>NUMBER OF PULSES IN PULSING ARC</td>
</tr>
<tr>
<td>NQEI</td>
<td>QUASI-LINEAR FILTERING EVENTS HAVING OCCURRED</td>
</tr>
<tr>
<td>NR</td>
<td>NUMBER OF ROWS IN THE OBSERVATION MATRIX</td>
</tr>
<tr>
<td>NST</td>
<td>NUMBER OF STATIONS TO BE USED (MAXIMUM 3)</td>
</tr>
<tr>
<td>NTMC</td>
<td>NOMINAL TRAJECTORY CODE</td>
</tr>
<tr>
<td>NTP</td>
<td>CODE OF TARGET PLANET</td>
</tr>
<tr>
<td>OMEGA</td>
<td>EARTH'S ROTATION RATE</td>
</tr>
<tr>
<td>ONE</td>
<td>1.5: NUMBER ONE (1) TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>QTAR(3)</td>
<td>DESIRED VALUES OF SMA, R, A, AND INC</td>
</tr>
<tr>
<td>P(6,6)</td>
<td>POSITION/VELOCITY COVARIANCE</td>
</tr>
<tr>
<td>PB(6,5)</td>
<td>POSITION/VELOCITY COVARIANCE AT INITIAL TIME</td>
</tr>
<tr>
<td>P7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>PCON(3)</td>
<td>PERTURBATIONS IN CONTROLS (ALPHA, DELTA, THETA)</td>
</tr>
<tr>
<td>PERP7</td>
<td>ORBIT INSERTION VARIABLES. NON-FUNCTIONAL IN EXISTING PROGRAM</td>
</tr>
<tr>
<td>PERV(10)</td>
<td>PERTURBATIONS SIZE FOR VELOCITY COMPONENTS IN CONSTRUCTING SENSITIVITY MATRICES IN TARGETING EVENTS</td>
</tr>
<tr>
<td>PG(6,6)</td>
<td>POSITION/VELOCITY CONTROL COVARIANCE</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>PHI(3,3)</td>
<td>TAREAL TARGETING MATRIX. REQ-D ONLY IF ISTART=1,2</td>
</tr>
<tr>
<td>PHI(6,6)</td>
<td>STM POSITION/VELOCITY STATE TRANSITION MATRIX</td>
</tr>
<tr>
<td>PHI2(3,3)</td>
<td>BAIM INVERSE OF VARIATION MATRIX PARTITION</td>
</tr>
<tr>
<td>PHILS</td>
<td>ZERDAT LATITUDE OF LAUNCH SITE</td>
</tr>
<tr>
<td>PI</td>
<td>COM THE VALUE OF THE MATHEMATICAL CONSTANT PI</td>
</tr>
<tr>
<td>PLANET(II)</td>
<td>PRT NAMES OF PLANETS</td>
</tr>
<tr>
<td>PMASS(II)</td>
<td>BLK GRAVITATIONAL CONSTANTS OF PLANETS IN ( \text{A.U.}^{-3}/\text{DAY}^{-2} )</td>
</tr>
<tr>
<td>POI</td>
<td>PBLK PROBABILITY OF IMPACT</td>
</tr>
<tr>
<td>PP(6,6)</td>
<td>STM POSITION/VELOCITY COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT</td>
</tr>
<tr>
<td>PROBI</td>
<td>BAIM ALLOWABLE PROBABILITY OF IMPACT</td>
</tr>
<tr>
<td>PS(24,24)</td>
<td>STM SOLVE-FOR PARAMETER COVARIANCE</td>
</tr>
<tr>
<td>PSB(24,24)</td>
<td>STM SOLVE-FOR PARAMETER COVARIANCE AT INITIAL TIME</td>
</tr>
<tr>
<td>PSG(24,24)</td>
<td>GJI SOLVE-FOR PARAMETER CONTROL COVARIANCE</td>
</tr>
<tr>
<td>PSI1</td>
<td>ZERDAT ANGLE OF FIRST BURN</td>
</tr>
<tr>
<td>PSI2</td>
<td>ZERDAT ANGLE OF SECOND BURN</td>
</tr>
<tr>
<td>PSP(24,24)</td>
<td>STM SOLVE-FOR PARAMETER COVARIANCE MATRIX PRIOR TO PROCESSING A MEASUREMENT</td>
</tr>
<tr>
<td>PSTAR</td>
<td>PBLK NOMINAL PROBABILITY DENSITY FUNCTION EVALUATED AT TARGET PLANET CENTER</td>
</tr>
<tr>
<td>PULHAG</td>
<td>PULS THRUST MAGNITUDE OF PULSING ENGINE</td>
</tr>
<tr>
<td>PULMAS</td>
<td>PULS NOMINAL MASS OF SPACECRAFT DURING PULSING ARC</td>
</tr>
<tr>
<td>PULT</td>
<td>PULS TOTAL TIME INTERVAL OF PULSING ARC</td>
</tr>
<tr>
<td>Q(6,6)</td>
<td>STM</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>QK(6,6)</td>
<td>EEX</td>
</tr>
<tr>
<td>R(4,4)</td>
<td>STM</td>
</tr>
<tr>
<td>RAP</td>
<td>ZOUT</td>
</tr>
<tr>
<td>RAD</td>
<td>CM</td>
</tr>
<tr>
<td>RADIUS(11)</td>
<td>BLK</td>
</tr>
<tr>
<td>RC(6)</td>
<td>VM</td>
</tr>
<tr>
<td>RCA</td>
<td>VM</td>
</tr>
<tr>
<td>RCA</td>
<td>LUNART</td>
</tr>
<tr>
<td>RCA1(6)</td>
<td>TRJ</td>
</tr>
<tr>
<td>RCA2(6)</td>
<td>TRJ</td>
</tr>
<tr>
<td>RCA3(6)</td>
<td>TRJ</td>
</tr>
<tr>
<td>RE(6)</td>
<td>VM</td>
</tr>
<tr>
<td>RES(4)</td>
<td>SIM1</td>
</tr>
<tr>
<td>RF(6)</td>
<td>OVER</td>
</tr>
<tr>
<td>RF(6)</td>
<td>OVERZ</td>
</tr>
<tr>
<td>RF1(6)</td>
<td>OVER</td>
</tr>
<tr>
<td>RF1(6)</td>
<td>OVERZ</td>
</tr>
<tr>
<td>RI(6)</td>
<td>OVER1</td>
</tr>
<tr>
<td>RI(6)</td>
<td>OVERR</td>
</tr>
<tr>
<td>RI(6)</td>
<td>LUNART</td>
</tr>
<tr>
<td>R11</td>
<td>OVER1</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RIN(6)</td>
<td>CURRENT STATE VECTOR AT I-TH EVENT</td>
</tr>
<tr>
<td>RK(2,3)</td>
<td>POSITION VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME (MIDPOINT OF PULSING ARC)</td>
</tr>
<tr>
<td>RMASS(11)</td>
<td>THE RELATIVE GRAVITATIONAL CONSTANT OF A STATED PLANET WITH RESPECT TO THE SUN</td>
</tr>
<tr>
<td>RHE(6)</td>
<td>GEOCENTRIC STATE OF CENTER OF MOON IN ECLIPTIC COORDINATES AT TSI</td>
</tr>
<tr>
<td>RMQ(6)</td>
<td>GEOCENTRIC STATE OF CENTER OF MOON AT TSI IN EQUATORIAL COORDINATES</td>
</tr>
<tr>
<td>RP</td>
<td>PARKING ORBIT RADIUS</td>
</tr>
<tr>
<td>RPE</td>
<td>RADIUS OF EARTH PARKING ORBIT</td>
</tr>
<tr>
<td>RPRAT</td>
<td>PARKING ORBIT INVERSE RATE</td>
</tr>
<tr>
<td>RRF(3)</td>
<td>SPACECRAFT POSITION AT END OF INTEGRATION, USED IN BROKEN PLANE TARGETING</td>
</tr>
<tr>
<td>RSI(3)</td>
<td>POSITION AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>RSI(6)</td>
<td>SELENOCENTRIC STATE OF S/C AT LUNAR SOI</td>
</tr>
<tr>
<td>RSOI1(3)</td>
<td>POSITION AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>RSOI2(3)</td>
<td>POSITION AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>RSOI3(3)</td>
<td>POSITION AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>RTP(6)</td>
<td>POSITION AND VELOCITY OF TARGET PLANET</td>
</tr>
<tr>
<td>RVS(6)</td>
<td>POSITION OF VEHICLE RELATIVE TO VIRTUAL MASS</td>
</tr>
<tr>
<td>S(24,4)</td>
<td>KALMAN GAIN CONSTANT FOR SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>SAL(3)</td>
<td>ALTITUDES OF STATIONS</td>
</tr>
<tr>
<td>SIGALP</td>
<td>VARIANCE OF ERROR IN POINTING ANGLE 1</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SIGBET</td>
<td>EVENT VARIANCE OF ERROR IN POINTING ANGLE 2</td>
</tr>
<tr>
<td>SIGNAL</td>
<td>ZEROAT NOMINAL LAUNCH AZIMUTH</td>
</tr>
<tr>
<td>SIGPRO</td>
<td>EVENT VARIANCE OF PROPORTIONALITY ERROR</td>
</tr>
<tr>
<td>SIGRES</td>
<td>EVENT VARIANCE OF RESOLUTION ERROR</td>
</tr>
<tr>
<td>SLAT(3)</td>
<td>CONST LATITUDES OF STATIONS</td>
</tr>
<tr>
<td>SLB(9)</td>
<td>SIMCNT BIASES IN STATION LOCATION CONSTANTS</td>
</tr>
<tr>
<td>SLON(3)</td>
<td>CONST LONGITUDES OF STATIONS</td>
</tr>
<tr>
<td>SHA</td>
<td>LUNART SIMI-MAJOR AXIS OF LUNAR HYPERBOLA (DESIRED)</td>
</tr>
<tr>
<td>SMJR(18)</td>
<td>BLK CONSTANTS USED TO CALCULATE THE SEMI-MAJOR AXES OF THE PLANETS</td>
</tr>
<tr>
<td>SPHERE(11)</td>
<td>BLK THE SPHERES OF INFLUENCE OF THE PLANETS IN A.U.</td>
</tr>
<tr>
<td>SPHFAC(10)</td>
<td>TAREAL REDUCTION FACTORS FOR TARGET PLANET SPHERE OF INFLUENCE FOR EACH EVENT</td>
</tr>
<tr>
<td>SSS(3)</td>
<td>VM DIRECTION COSINES VECTOR OF SPACECRAFT SPIN AXIS</td>
</tr>
<tr>
<td>ST(50)</td>
<td>BLK CONSTANTS USED TO CALCULATE THE ORBITAL ELEMENTS OF THE LAST FOUR PLANETS</td>
</tr>
<tr>
<td>T</td>
<td>BLK TRAJECTORY TIME IN DAYS</td>
</tr>
<tr>
<td>T1(20)</td>
<td>TMW2 EIGENVECTOR EVENT TIMES</td>
</tr>
<tr>
<td>T2(20)</td>
<td>TMW2 PREDICTION EVENT STARTING TIMES</td>
</tr>
<tr>
<td>T3(10)</td>
<td>BIM ARRAY OF GUIDANCE EVENT TIMES</td>
</tr>
<tr>
<td>T4(20)</td>
<td>TMW2 CONIC COMPUTATION EVENT TIMES</td>
</tr>
<tr>
<td>T5(20)</td>
<td>TMW2 QUASI-LINEAR EVENT TIMES</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>T6(20)</td>
<td>TMW2</td>
</tr>
<tr>
<td>T7</td>
<td>TMW2</td>
</tr>
<tr>
<td>TACA</td>
<td>VM</td>
</tr>
<tr>
<td>TAR(6,10)</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TAU1</td>
<td>EVENT</td>
</tr>
<tr>
<td>TCA</td>
<td>LUNAR</td>
</tr>
<tr>
<td>TCA1</td>
<td>TRJ</td>
</tr>
<tr>
<td>TCA2</td>
<td>TRJ</td>
</tr>
<tr>
<td>TCA3</td>
<td>TRJ</td>
</tr>
<tr>
<td>TEN</td>
<td>DPNUM</td>
</tr>
<tr>
<td>TEV(50)</td>
<td>EVENT</td>
</tr>
<tr>
<td>TEVN</td>
<td>OVER1</td>
</tr>
<tr>
<td>TEVN</td>
<td>OVERR</td>
</tr>
<tr>
<td>TF</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>TG</td>
<td>GUI</td>
</tr>
<tr>
<td>TGT3(10)</td>
<td>TARVAR</td>
</tr>
<tr>
<td>THT10</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>THEL0</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>THREE</td>
<td>OPNUM</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>TI</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>TIMG(10)</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TIM1</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>TIM2</td>
<td>ZERDAT</td>
</tr>
<tr>
<td>TIMINT</td>
<td>VM</td>
</tr>
<tr>
<td>TIMS</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TIN</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TINJ</td>
<td>BAIIH</td>
</tr>
<tr>
<td>TM</td>
<td>VM</td>
</tr>
<tr>
<td>TMN(1000)</td>
<td>MEAS</td>
</tr>
<tr>
<td>TMPR(3)</td>
<td>BAIIH</td>
</tr>
<tr>
<td>TNOHB(3)</td>
<td>BAIIH</td>
</tr>
<tr>
<td>TNOHC(7)</td>
<td>BAIIH</td>
</tr>
<tr>
<td>THU</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TMU</td>
<td>LUNART</td>
</tr>
<tr>
<td>TOL(6,10)</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TRTM</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TOLR(6)</td>
<td>TAREAL</td>
</tr>
<tr>
<td>TOLR(6)</td>
<td>TAREAL</td>
</tr>
<tr>
<td>Identifier</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TPT2(20)</td>
<td>EVENT ARRAY OF TIMES TO WHICH A PREDICTION IS MADE</td>
</tr>
<tr>
<td>TRTH</td>
<td>TAREAL TRAJECTORY TIME ON NOMINAL TRAJECTORY</td>
</tr>
<tr>
<td>TRTH1</td>
<td>TIME INITIAL TRAJECTORY TIME</td>
</tr>
<tr>
<td>TRTHB</td>
<td>TIME TRAJECTORY TIME AT BEGINNING OF TRAJECTORY</td>
</tr>
<tr>
<td>TSL</td>
<td>LUNART PROJECTED J.D. AT SOI INERTSECTION</td>
</tr>
<tr>
<td>TSO11</td>
<td>TRJ TIME AT SPHERE OF INFLUENCE OF ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>TS012</td>
<td>TKJ TIME AT SPHERE OF INFLUENCE OF MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>TS013</td>
<td>TRJ TIME AT SPHERE OF INFLUENCE OF ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>TSPH</td>
<td>LUNART RADIUS OF LUNAR SOI (KM)</td>
</tr>
<tr>
<td>TTH1</td>
<td>SIMCNT FIRST TIME USED FOR UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>TTH2</td>
<td>SIMCNT SECOND TIME USED FOR UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>TTOL(3)</td>
<td>LUNART ALLOWABLE TOLERANCES IN SHA, T, B, R</td>
</tr>
<tr>
<td>TWO</td>
<td>DPNUM THE NUMBER TWO (2) TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>TWOP1</td>
<td>DPNUM THE MATHEMATICAL CONSTANT 2. *PI</td>
</tr>
<tr>
<td>TXU(6,8)</td>
<td>STM STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH DYNAMIC CONSIDER PARAMETERS</td>
</tr>
<tr>
<td>TXXS(6,24)</td>
<td>STM STATE TRANSITION MATRIX PARTITION ASSOCIATED WITH SOLVE-FOR PARAMETERS</td>
</tr>
<tr>
<td>UO(3,8)</td>
<td>STM DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX</td>
</tr>
<tr>
<td>UNIVT</td>
<td>TIME UNIVERSAL TIME</td>
</tr>
<tr>
<td>UNMAC(3,3)</td>
<td>SIMCNT UNMODELED ACCELERATION</td>
</tr>
<tr>
<td>UST(2)</td>
<td>CONST2 DIRECTION COSINES OF 3 REFERENCE STARS</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>V(16,7)</td>
<td>AN ARRAY WHICH STORES PERTINANT VECTORS USED IN THE CALCULATION OF THE VIRTUAL MASS TRAJECTORY (SEE LARGE ARRAY DEFNS IN SECTION 5.3)</td>
</tr>
<tr>
<td>V0(15,1)</td>
<td>MEASUREMENT CONSIDER PARAMETER COVARIANCE MATRIX</td>
</tr>
<tr>
<td>VHPM</td>
<td>MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY AT TARGET BODY (VHP VECTOR)</td>
</tr>
<tr>
<td>VINF</td>
<td>HYPERBOLIC EXCESS VELOCITY</td>
</tr>
<tr>
<td>VK(2,3)</td>
<td>VELOCITY VECTORS OF LAUNCH AND TARGET PLANETS AT IMPULSIVE TIME (MIDPOINT OF PULSING ARC)</td>
</tr>
<tr>
<td>VMU</td>
<td>GRAVITATIONAL CONSTANT OF VIRTUAL MASS</td>
</tr>
<tr>
<td>VS(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>VSOI1(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>VSOI2(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>VSOI3(3)</td>
<td>VELOCITY AT SPHERE OF INFLUENCE ON ACTUAL TRAJECTORY</td>
</tr>
<tr>
<td>VST(3)</td>
<td>DIRECTION COSINES OF 3 REFERENCE STARS</td>
</tr>
<tr>
<td>W(17)</td>
<td>ACTUAL DYNAMIC NOISE</td>
</tr>
<tr>
<td>WST(3)</td>
<td>DIRECTION COSINES OF 3 REFERENCE STARS</td>
</tr>
<tr>
<td>XAC(5,10)</td>
<td>ACCURACY LEVELS EMPLOYED IN TARGETING</td>
</tr>
<tr>
<td>XB(6)</td>
<td>BEGINNING ORIGINAL NOMINAL VEHICLE STATE VECTOR</td>
</tr>
<tr>
<td>XBDS</td>
<td>ORIGINAL VALUE OF B.T IN N-L GUIDANCE</td>
</tr>
<tr>
<td>XBDR</td>
<td>ORIGINAL VALUE OF B.R IN N-L GUIDANCE</td>
</tr>
<tr>
<td>XDC</td>
<td>ORIGINAL VALUE OF DC IN N-L GUIDANCE</td>
</tr>
<tr>
<td>XDELV(3,10)</td>
<td>NONLINEAR VELOCITY CORRECTION</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>XDSI</td>
<td>SAVVAL ORIGINAL VALUE OF TSI IN N-L GUIDANCE</td>
</tr>
<tr>
<td>XDVMAX(10)</td>
<td>TARVAR MAXIMUM ALLOWABLE VELOCITY CORRECTION</td>
</tr>
<tr>
<td>XF(6)</td>
<td>STVEC FINAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>XF1(6)</td>
<td>SIM1 FINAL STATE VECTOR OF MOST RECENT NOMINAL TRAJECTORY</td>
</tr>
<tr>
<td>XFC(10)</td>
<td>TARVAR SPHERE OF INFLUENCE FACTORS</td>
</tr>
<tr>
<td>XG(6)</td>
<td>GUI STATE VECTOR AT TIME OF LAST GUIDANCE EVENT</td>
</tr>
<tr>
<td>XI(6)</td>
<td>STVEC INITIAL VEHICLE STATE VECTOR OF ORIGINAL NOMINAL</td>
</tr>
<tr>
<td>XI1(6)</td>
<td>SIM1 INITIAL STATE VECTOR OF MOST RECENT NOMINAL</td>
</tr>
<tr>
<td>XIN(6)</td>
<td>OVERX STATE VECTOR TRANSFERRED TO NONLIN</td>
</tr>
<tr>
<td>XLAB(6)</td>
<td>XXXL VEHICLE POSITION/VELOCITY VECTOR COMPONENT NAMES</td>
</tr>
<tr>
<td>XLM(2,2)</td>
<td>PBLK PROJECTION OF TARGET CONDITION COVARIANCE MATRIX INTO THE IMPACT PLANE</td>
</tr>
<tr>
<td>XLM(2,2)</td>
<td>PBLK INVERSE OF XLM(2,2)</td>
</tr>
<tr>
<td>XMUS(2)</td>
<td>PBLK NOMINAL IMPACT PLANE TARGET STATE</td>
</tr>
<tr>
<td>XNM(24)</td>
<td>XXXL AUGMENTATION PARAMETER LABELS</td>
</tr>
<tr>
<td>XP(6)</td>
<td>BLK THE POSITION AND VELOCITY OF A PLANET IN HELIOCENTRIC ECLIPTIC COORDINATES</td>
</tr>
<tr>
<td>XPERV(10)</td>
<td>TARVAR VELOCITY PERTURBATION USED TO COMPUTE TARGETING MATRIX</td>
</tr>
<tr>
<td>XRC(6)</td>
<td>SAVVAL ORIGINAL VALUE OF RC IN N-L GUIDANCE</td>
</tr>
<tr>
<td>XRSI(3)</td>
<td>SAVVAL ORIGINAL VALUE OF RSI IN N-L GUIDANCE</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>XSL(24)</td>
<td>SOLVE-FOR PARAMETER LABELS</td>
</tr>
<tr>
<td>XTAR(6,10)</td>
<td>DESIRED TARGET VALUES</td>
</tr>
<tr>
<td>XTOL(6,10)</td>
<td>TOLERANCES ON TARGET PARAMETERS</td>
</tr>
<tr>
<td>XU(8)</td>
<td>DYNAMIC CONSIDER PARAMETER LABELS</td>
</tr>
<tr>
<td>XV(15)</td>
<td>MEASUREMENT CONSIDER PARAMETER LABELS</td>
</tr>
<tr>
<td>XVSI(3)</td>
<td>ORIGINAL VALUE OF VSI IN N- GUIDANCE</td>
</tr>
<tr>
<td>XXIN(6)</td>
<td>STATE VECTOR TRANSFERRED TO EXECUT OR EXECUTS</td>
</tr>
<tr>
<td>Z(17)</td>
<td>ACTUAL STATE VECTOR</td>
</tr>
<tr>
<td>ZDAT(6)</td>
<td>ZERO ITERATE VECTOR</td>
</tr>
<tr>
<td>ZERDAT</td>
<td>IF IZERO=0 ZDAT(1-6) = INITIAL STATE</td>
</tr>
<tr>
<td></td>
<td>=2,3,4, ZDAT(1-3) = INITIAL</td>
</tr>
<tr>
<td></td>
<td>POSITION ZDAT(4-6) = FINAL POSITION</td>
</tr>
<tr>
<td>ZERO</td>
<td>THE NUMBER ZERO (0) TO NINE SIGNIFICANT FIGURES</td>
</tr>
<tr>
<td>ZI(17)</td>
<td>INITIAL ACTUAL STATE VECTOR</td>
</tr>
<tr>
<td>ZF(6)</td>
<td>FINAL ACTUAL STATE VECTOR AFTER ADDING THE EFFECT OF UNMODELED ACCELERATION</td>
</tr>
</tbody>
</table>
5.3 Large Array Definitions

In this section large arrays appearing in COMMON will be displayed. The arrays depicted are frequently referenced in trajectory propagation subroutines in STEAP; hence the programmer studying such subroutines will find the following tables extremely useful.

Tables 5.1 to 5.5 describe arrays containing planetary ephemeris constants. The values actually stored in these arrays may be found in the documentation for BLOCK DATA. Tables 5.6 through 5.8 contain variables used in the virtual mass propagation procedure. Discussions of these variables may be found in VMP, ZPHM, ORB, and similar routines.

<table>
<thead>
<tr>
<th>Constant</th>
<th>i</th>
<th>( \Omega )</th>
<th>( \omega )</th>
<th>e</th>
<th>M</th>
<th>a</th>
<th>( \omega )</th>
<th>E</th>
<th>( a_0 )</th>
<th>( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>.7</td>
<td>.8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Venus</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>.5</td>
<td>.6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Earth</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>.2</td>
<td>.3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Mars</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>.3</td>
<td>.4</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Jupiter</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>.3</td>
<td>.4</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Saturn</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>.4</td>
<td>.5</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Uranus</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>.5</td>
<td>.6</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Neptune</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>.6</td>
<td>.7</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Pluto</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>.7</td>
<td>.8</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Moon</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>.8</td>
<td>.9</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5.1 ELMAT Array -- Conic Elements

<table>
<thead>
<tr>
<th>Constant</th>
<th>i</th>
<th>( \Omega )</th>
<th>( \omega )</th>
<th>e</th>
<th>M</th>
<th>a</th>
<th>( \omega )</th>
<th>E</th>
<th>( a_0 )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( a_4 )</th>
<th>( a_5 )</th>
<th>( a_6 )</th>
<th>( a_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>.7</td>
<td>.8</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>.7</td>
<td>.8</td>
</tr>
<tr>
<td>Venus</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>.5</td>
<td>.6</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>.7</td>
<td>.8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Earth</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>.2</td>
<td>.3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>.9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Mars</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>.3</td>
<td>.4</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Jupiter</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>.3</td>
<td>.4</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Saturn</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>.4</td>
<td>.5</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Uranus</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>.5</td>
<td>.6</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Neptune</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>.6</td>
<td>.7</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>Pluto</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>.7</td>
<td>.8</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>Moon</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>.8</td>
<td>.9</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 5.2 SMAR Array

<table>
<thead>
<tr>
<th>Constant</th>
<th>i</th>
<th>( \Omega_0 )</th>
<th>( \Omega_1 )</th>
<th>( \omega_0 )</th>
<th>( \omega_1 )</th>
<th>e_0</th>
<th>e_1</th>
<th>M_0</th>
<th>M_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Venus</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Earth</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>Mars</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>Jupiter</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>Saturn</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>Uranus</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
</tr>
<tr>
<td>Neptune</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>Pluto</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td>Moon</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
</tr>
</tbody>
</table>

Table 5.3 CN Array -- Inner Planet Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>i</th>
<th>( \Omega_0 )</th>
<th>( \Omega_1 )</th>
<th>( \omega_0 )</th>
<th>( \omega_1 )</th>
<th>e_0</th>
<th>e_1</th>
<th>M_0</th>
<th>M_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jupiter</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Saturn</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>Uranus</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>Neptune</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>Pluto</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>Moon</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
</tr>
</tbody>
</table>

Table 5.4 ST Array -- Outer Planet Constants
Table 5.5 EMN Array -- Lunar Constants

\[ \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\
L_0 & L_1 & L_2 & L_3 & I & e & a \\
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15
\end{array} \]

Table 5.5 EMN Array -- Lunar Constants

| \( x_1 \) | \( y_1 \) | \( z_1 \) | \( r_1 \) |
| \( \dot{x}_1 \) | \( \dot{y}_1 \) | \( \dot{z}_1 \) | \( \dot{r}_1 \) |
| \( \ddot{x}_{s1} \) | \( \ddot{y}_{s1} \) | \( \ddot{z}_{s1} \) | \( \ddot{r}_{s1} \) |
| \( \dddot{x}_{s1} \) | \( \dddot{y}_{s1} \) | \( \dddot{z}_{s1} \) | \( \dddot{r}_{s1} \) |

Note:
Subscript \( i \) indicates component is \( i \)-th body referenced to inertial coordinate system.
Subscript \( si \) indicates component is spacecraft referenced to \( i \)-th body.

Table 5.6 F-Array -- Ephemeral Data

| TRG(1) \( \cos E \) | TRG(5) \( \cos i \) | TRG(9) \( \cos \omega \) | TRG(13) \( \cos(\omega + f) \) |
| TRG(2) \( \sin E \) | TRG(6) \( \sin i \) | TRG(10) \( \sin \omega \) | TRG(14) \( \sin(\omega + f) \) |
| TRG(3) \( \cos f \) | TRG(7) \( \cos \Omega \) | TRG(11) \( \cos \omega \) |
| TRG(4) \( \sin f \) | TRG(8) \( \sin \Omega \) | TRG(12) \( \sin \omega \) |

Table 5.7 TRG Array -- Trigonometric Functions
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$u_1$</td>
<td>$e_1$</td>
<td>$z_{e_1}$</td>
<td>$e_{x_1}$</td>
<td>$z_{x_1}$</td>
<td>$w$ (deg/l), $w$ (rad/l)</td>
<td>$D$</td>
</tr>
<tr>
<td>3</td>
<td>$e_3$</td>
<td>$e_3$</td>
<td>$e_3$</td>
<td>$e_3$</td>
<td>$(r_{x_3})$, $r_{y_3}$</td>
<td>$(r_{x_3})$, $r_{y_3}$</td>
<td>$C_3$</td>
</tr>
<tr>
<td>4</td>
<td>$l_4$</td>
<td>$e_4$</td>
<td>$e_4$</td>
<td>$e_4$</td>
<td>$C_4$</td>
<td>$D$ (velocity)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$(r_{x_5})$, $r_{y_5}$</td>
<td>$(r_{x_5})$, $r_{y_5}$</td>
<td>$(r_{x_5})$, $r_{y_5}$</td>
<td>$(r_{x_5})$, $r_{y_5}$</td>
<td>$D^2$ (area rate)</td>
<td>$D^3$ (mass rate)</td>
<td>$C_5$</td>
</tr>
<tr>
<td>6</td>
<td>$r_6$</td>
<td>$r_6$</td>
<td>$r_6$</td>
<td>$r_6$</td>
<td>$r_6$</td>
<td>$r_6$</td>
<td>$r_6$</td>
</tr>
<tr>
<td>7</td>
<td>$(r_{x_7})$, $r_{y_7}$</td>
<td>$(r_{x_7})$, $r_{y_7}$</td>
<td>$(r_{x_7})$, $r_{y_7}$</td>
<td>$(r_{x_7})$, $r_{y_7}$</td>
<td>$D^2$ (area rate)</td>
<td>$D^3$ (mass rate)</td>
<td>$C_7$</td>
</tr>
<tr>
<td>8</td>
<td>$r_8$</td>
<td>$r_8$</td>
<td>$r_8$</td>
<td>$r_8$</td>
<td>$r_8$</td>
<td>$r_8$</td>
<td>$r_8$</td>
</tr>
<tr>
<td>9</td>
<td>$r_9$</td>
<td>$r_9$</td>
<td>$r_9$</td>
<td>$r_9$</td>
<td>$r_9$</td>
<td>$r_9$</td>
<td>$r_9$</td>
</tr>
<tr>
<td>10</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
<td>$r_{10}$</td>
</tr>
<tr>
<td>11</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
<td>$r_{11}$</td>
</tr>
<tr>
<td>12</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
<td>$r_{12}$</td>
</tr>
<tr>
<td>13</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
<td>$r_{13}$</td>
</tr>
<tr>
<td>14</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
<td>$r_{14}$</td>
</tr>
<tr>
<td>15</td>
<td>$(r_{x_{15}})$, $r_{y_{15}}$</td>
<td>$(r_{x_{15}})$, $r_{y_{15}}$</td>
<td>$(r_{x_{15}})$, $r_{y_{15}}$</td>
<td>$(r_{x_{15}})$, $r_{y_{15}}$</td>
<td>$D^2$ (area rate)</td>
<td>$D^3$ (mass rate)</td>
<td>$C_{15}$</td>
</tr>
<tr>
<td>16</td>
<td>$b_1$</td>
<td>$b_1$</td>
<td>$b_1$</td>
<td>$b_1$</td>
<td>$b_1$</td>
<td>$b_1$</td>
<td>$b_1$</td>
</tr>
</tbody>
</table>

Table 5.8 W-Array -- Virtual Mass Propagation Variables
6. INDIVIDUAL SUBROUTINE DOCUMENTATION

This chapter contains the individual documentation for all the subroutines in the STEAP II series. The following information is given for each subroutine.

1. Purpose: The tasks performed by the subroutine.
2. Calling Sequence: The statement by which the subroutine is called.
3. Arguments: The arguments in the calling sequence, their definition, and identification as input, output, or both.
4. Subroutines Supported: A list of subroutines calling the subroutine being documented.
5. Subroutines Required: A list of subroutines called by the subroutines being documented.
6. Local Symbols: The internal (non-common) variables used in the subroutine and their definitions.
7. Common Computed/Used: A list of variables appearing in common blocks which are both computed and used (see Chapter 3 for definitions).
8. Common Computed: A list of common variables which are set in the program.
9. Common Used: A list of common variables only used by the subroutine.
10. Analysis: The detailed mathematical analysis on which the subroutine is based (if applicable).
11. Flowchart: A flowchart of the operation of the program (if required).

The reader is referred to Chapter 4 for an index of all subroutines of STEAP II (Tables 4.1 and 4.2) and for the calling hierarchies of the basic subprograms of STEAP II (Figures 4.1 to 4.4).
SUBROUTINE BATCON

PURPOSE: BATCON IS A CONIC PROPAGATOR USING THE BATTIN UNIVERSAL VARIABLE FORMULATION.

CALLING SEQUENCE: CALL BATCON(GM,RV,VV,DT,DV,SV)

ARGUMENTS:
GM       I  GRAVITATIONAL CONSTANT
RV(3)    I  INITIAL POSITION VECTOR
VV(3)    I  INITIAL VELOCITY VECTOR
DT       I  TIME INTERVAL OF PROPAGATION
DV(3)    O  FINAL POSITION VECTOR
SV(3)    O  FINAL VELOCITY VECTOR

SUBROUTINES SUPPORTED: PERHEL

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:
ALP      BATTIN ALPHA VARIABLE
ASQ      SQRT OF ALP
CONT     INTERMEDIATE VARIABLE
CON      NORMALIZED TIME
DELX     CORRECTION TO CURRENT VALUE OF X
EX       E RAISED TO THE ASQ*X POWER
FT       F SERIES FOR VELOCITY EVALUATED AT EPOCH
F        F SERIES FOR POSITION EVALUATED AT EPOCH
GMSQ     SQRT OF GRAVITATIONAL CONSTANT
GT       G SERIES FOR VELOCITY EVALUATED AT EPOCH
G        G-SERIES COEFFICIENT FOR POSITION
I        KEPLER-LIKE EQUATION ITERATION COUNTER
RI       NAME FOR COMPONENTS OF INITIAL POSITION
RO       MAGNITUDE OF INITIAL POSITION
R2       SQUARE OF RO
R  ITERATE VALUE OF RADIUS
SIG  INTERMEDIATE VARIABLE
U0  BATTIN TRANSCENDENTAL FUNCTION
U1  BATTIN TRANSCENDENTAL FUNCTION
U2  BATTIN TRANSCENDENTAL FUNCTION
U3  BATTIN TRANSCENDENTAL FUNCTION
V1  NAME FOR COMPONENTS OF INITIAL VELOCITY
V2  SQUARE OF INITIAL SPEED
X  BATTIN ITERATION VARIABLE
BATCON Analysis

BATCON is a conic propagator using the Battin universal variable formulation. A total derivation is too involved to be given here; rather the results of Battin's work will be given here.

Let the initial state of a point mass moving under the influence of a gravitational force $\mu$ be given by $\vec{r}_0$, $\vec{v}_0$. It is required to determine the state $\vec{r}$, $\vec{v}$ at a time $T$ units later. It is useful to introduce the parameters

$$
\sigma_0 = \frac{\vec{r}_0 \cdot \vec{v}_0}{\sqrt{\mu}}
$$

$$
J = \frac{2}{\vec{r}_0} - \frac{\vec{v}_0^2}{\mu}
$$

Battin's approach is to introduce a new independent variable $x(t)$ in place of time by the relation

$$
\frac{dx}{dt} = \frac{\sqrt{\mu}}{r(t)}
$$

This parametrization greatly simplifies the conic propagation problem. For suppose that the value of $x$ corresponding to $t = T$ is given by $X$, i.e. $x(T) = X$. Then the final state is given by

$$
\vec{r} = R_2(X) \vec{r}_0 + R_2(X) \vec{v}_0
$$

$$
\vec{v} = V_1(X) \vec{r}_0 + V_2(X) \vec{v}_0
$$

where

$$
R_2(X) = 1 - \frac{1}{\vec{r}_0} U_2(X)
$$

$$
R_2(X) = \sqrt{\mu} \left[ \frac{r_0 U_1(X) + \sigma_0 U_2(X)}{r_0} \right]
$$

$$
V_1(X) = -\frac{\sqrt{\mu}}{\vec{r}_0} \frac{U_1(X)}{\vec{r}_0}
$$

$$
V_2(X) = 1 - \frac{1}{\vec{r}_0} U_2(X)
$$
and where

\[ \begin{align*}
U_0(x) &= \cos \alpha x & \alpha > 0 \\
U_1(x) &= \frac{\sin \sqrt{-\alpha} x}{\sqrt{-\alpha}} & \alpha < 0 \\
U_2(x) &= \frac{1 - U_0(x)}{\alpha} \\
U_3(x) &= \frac{x - U_2(x)}{\alpha}
\end{align*} \]

The problem is thus reduced to the determination of \( X \). \( X \) is generated iteratively by the recursive formulae

\[ \begin{align*}
x_{n+1} &= x_n - \frac{\sqrt{\mu} t_n - \sqrt{\mu} t}{r_n} = x_n - \Delta x
\end{align*} \]

where

\[ \begin{align*}
\sqrt{\mu} t_n &= r_0 U_1(x_n) + \sigma U_2(x_n) + U_3(x_n) \\
r_n &= r_0 U_0(x_n) + \sigma U_1(x_n) + U_2(x_n)
\end{align*} \]

To start the process the initial guess is set to

\[ x_0 = \frac{\sqrt{\mu} T}{r_0} \left( 1 - \frac{\sigma r_0^2 \sqrt{\mu} T}{2r_0^2} \right) \left[ 3 \sigma^2 - r_0 (1 - \alpha r_0^2) \right] \left[ \frac{1}{6r_0} \right] - \frac{1}{6r_0} \left[ 3 \sigma^2 - r_0 (1 - \alpha r_0^2) \right] \left[ \frac{1}{6r_0} \right] \]

The program sets \( X = x_n \) when the correction \( \Delta x \) is less than \( 10^{-8} \).

It terminates if the number of iterations exceeds 10.

References:


BATICON Flowchart

ENTER

Compute \( \sigma_c, a, \) etc.

Compute initial guess \( x \)

Compute \( U_0(x), U_1(x), U_2(x), U_3(x) \sqrt{\mu_1 r_n, r_n} \)

Iterations : 10

\( \geq \) STOP

\( < 0 \)

Compute correction \( \Delta x \) to \( x \)

\( |\Delta x| : \epsilon \)

\( \geq \)

\( < \)

Compute \( R_1, R_2, V_1, V_2, T, \nabla \)

RETURN
SUBROUTINE BIAIM

PURPOSE: TO PERFORM BIADED AIMPOINT GUIDANCE.

CALLING SEQUENCE: CALL BIAIM(RI, TEVN)

ARGUMENTS: RI I NOMINAL SPACECRAFT STATE AT TIME OF BIADED AIMPOINT GUIDANCE EVENT
TEVN I TIME OF BIADED AIMPOINT GUIDANCE EVENT

SUBROUTINES SUPPORTED: GU1SIM GU1ION

SUBROUTINES REQUIRED: MATIN POLCON PSIM QCOHP

LOCAL SYMBOLS: AOA1 VARIATION MATRIX AT TIME T(J+1)
BB RIGHT HALF PARTITION OF AOA1 MATRIX
CSQ CONSTANT DEFINING CONSTRAINT ELLIPSE
C1 A COEFFICIENT IN THE NECESSARY CONDITION
C2 A COEFFICIENT IN THE NECESSARY CONDITION
C3 A COEFFICIENT IN THE NECESSARY CONDITION
C4 A COEFFICIENT IN THE NECESSARY CONDITION
C5 A COEFFICIENT IN THE NECESSARY CONDITION
C SQUARE ROOT OF CSQ
DELMU AIMPOINT BIAS IN IMPACT PLANE
DENOM INTERMEDIATE VARIABLE
DET DETERMINANT OF PROJECTED TARGET CONDITION COVARIANCE MATRIX
DVBIAS BIAS VELOCITY CORRECTION
DVIT TOTAL VELOCITY CORRECTION IF BIAS IS REMOVED
DVT TOTAL VELOCITY CORRECTION IF BIAS IS APPLIED
DVUPP UPDATE VELOCITY ITERATE
D1 PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE
02 PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE

03 PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE

04 PARTIAL DERIVATIVE USED IN NEWTON ITERATION TECHNIQUE

EE MATRIX DEFINING FUNCTION TO BE MINIMIZED

IKNT COUNTER ON NEWTON ITERATION LOOP

IS INDEX OF NEXT GUIDANCE EVENT

ITRN COUNTER ON OUTER ITERATION LOOP

NDIM1S STORAGE FOR NDIM1

NDIM2S STORAGE FOR NDIM2

PHII INVERSE OF STATE TRANSITION MATRIX

PHI1 INTERMEDIATE ARRAY

PSIJ1 GUIDANCE MATRIX PSI AT T(J+1)

.PSIJ Guidance MATRIX PSI AT EVENT TIME T(j)

QUOT INTERMEDIATE VARIABLE

RF DUMMY VECTOR

SAVET STORAGE FOR TRTM1

SUM1 INTERMEDIATE VARIABLE

SUM INTERMEDIATE VARIABLE

TWOE CONSTANT DEFINING CONSTRAINT ELLIPSE

VCA SPACECRAFT CLOSEST APPROACH VELOCITY RELATIVE TO TARGET PLANET

.XK4 INTERMEDIATE VARIABLE

XMU HOST RECENT IMPACT PLANE AIMPOINT

XM1 AIMPOINT ITERATE

XM2 AIMPOINT ITERATE

XM STORAGE FOR MOST RECENT AIMPOINT ITERATE
XN1  NEGATIVE OF CONSTRAINT EQUATION EVALUATED
     AT MOST RECENT AIMPOINT ITERATE

XN2  NEGATIVE OF NECESSARY CONDITION EVALUATED
     AT MOST RECENT AIMPOINT ITERATE

YY   INTERMEDIATE VARIABLE

ZH   AIMPOINT INCREMENT FOR MOST RECENT
     ITERATION

ZK   AIMPOINT INCREMENT FOR MOST RECENT
     ITERATION

COMMON COMPUTED/USED:

A  CR  DVM  DVRB  DVUP
EXEC  IBIAS  IIGP  PHI2  RCA
TMPR  TRTH1  XMUS

COMMON COMPUTED:

DELTN

COMMON USED:

ADA  ALNYM  ATANS  DUMMYQ  EM3
IDENS  IEMC  IGUID  II  ISTMC
ITR  NTP  ONE  PHI  PMASS
POI  PROBI  RADIUS  TINJ  TM
TNORB  TNOHC  TWO  T3  VINF
XLANH  XLAM  ZERO
BIADM Analysis

Subroutine BIADM performs biased aimpoint guidance computations. If planetary quarantine constraints are in effect at injection or at a midcourse correction, and if the nominal aimpoint does not satisfy these constraints, subroutine BIADM will compute a biased aimpoint and the required bias velocity correction such that the constraints are satisfied and some performance functional is minimized.

Aimpoint biasing is performed in the impact plane and as such permits only two degrees of freedom in the selection of the biased aimpoint. The general aimpoint in the impact plane will be denoted by the 2-dimensional vector $\hat{\mu}_j$, where the $j$-subscript indicates that the biased aimpoint guidance event is occurring at time $t_j$. Three midcourse guidance policies are available in STEAP, and it will be necessary to relate $\hat{\mu}_j$ to the specific aimpoint for each of these three policies. These relationships are summarized below:

(a) Two-variable B-plane (2VBP):

$$\hat{\mu}_j = \begin{bmatrix} \mathbf{B} \cdot \mathbf{T} \\ \mathbf{B} \cdot \mathbf{R} \end{bmatrix} \quad (1)$$

(b) Three-variable B-plane (3VBP):

$$\hat{\mu}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{B} \cdot \mathbf{T} \\ \mathbf{B} \cdot \mathbf{R} \end{bmatrix} \quad (2)$$

(c) Fixed-time-of-arrival (FTA):

$$\hat{\mu}_j = A \hat{\mathbf{v}}_{CA} \quad (3)$$

where $\hat{\mathbf{v}}_{CA}$ is the nominal closest approach position of the spacecraft relative to the target planet. Coordinate transformation $A$ projects the 3-dimensional vector $\hat{\mathbf{v}}_{CA}$ (referred to ecliptic coordinates) into an equivalent FTA impact plane which is defined to be the plane containing $\hat{\mathbf{v}}_{CA}$ and perpendicular to the spacecraft closest approach velocity $\hat{\mathbf{v}}_{CA}$ relative to the target planet. If the ecliptic coordinates of $\hat{\mathbf{v}}_{CA}$ and $\hat{\mathbf{v}}_{CA}$ are denoted by $r_x', r_y', r_z'$ and $v_{x}'$, $v_{y}'$, $v_{z}'$, respectively, then the transformation $A$ is given by
Spacecraft state variations at \( t_j \) are related to aimpoint variations (target condition variations) by the variation matrix \( \eta_j \), which is always available prior to calling BLAM. Thus, the statistical state dispersions about the nominal following the guidance correction at \( t_j \) and represented by the control covariance \( P_{c_j}^+ \), can be related to the dispersions about the nominal aimpoint represented by \( W_j^+ \) according to the equation

\[
W_j^+ = \eta_j \cdot P_{c_j}^+ \cdot \eta_j^T
\]

The control covariance \( P_{c_j}^+ \) is computed from

\[
P_{c_j}^+ = P_{k_j}^- + \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & \tilde{Q}_j
\end{bmatrix}
\]

where \( P_{k_j}^- \) is the knowledge covariance prior to the guidance event and \( \tilde{Q}_j \) is the execution error covariance.

Transformations employed in equations (1) through (3) can also be employed to project \( W_j^+ \) into the impact plane. The resulting projection is denoted by the covariance \( \Lambda_j \), and is obtained from \( W_j^+ \) according to the following equations:

(a) 2VBP : \( \Lambda_j = W_j^+ \)

(b) 3VBP : \( \Lambda_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot W_j^+ \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \)

(c) PTA : \( \Lambda_j = A \cdot W_j^+ \cdot A^T \)
With covariance $\Lambda_j$ available, it is now possible to compute the probability of impact $P_{\text{II}}$. Assuming the probability density function associated with $\Lambda_j$ is gaussian and nearly constant over the target planet capture area permits us to compute $P_{\text{II}}$ using the equation

$$P_{\text{II}} = \pi R_c^2 p$$

(10)

where $R_c$ is the target planet capture radius and $p$ represents the gaussian density function evaluated at the target planet center and is given by

$$p = \frac{1}{2\pi |\Lambda_j|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} \hat{\mu}^T \Lambda_j^{-1} \hat{\mu} \right]$$

(11)

The nominal impact plane aimpoint is denoted by $\hat{\mu}^*$. Subroutine BIADM calls subroutine PFLCWM to perform the computations involved in equations (7) through (11).

Capture radius $R_c$ is simply the physical radius $R_p$ of the target planet if the FIA guidance policy is employed, while for the two S-plane policies the capture radius is given by

$$R_c = R_p \sqrt{1 + \frac{2\mu_p}{V_{\infty}^2 R_p}}$$

(11)

where $\mu_p$ is the target planet gravitational constant and $V_{\infty}$ is the hyperbolic excess velocity.

If the probability of impact $P_{\text{II}}$ does not exceed the permissible impact probability $P_{\text{I}}$, and if the nominal aimpoint has not been previously biased, we simply return to subroutine GUDEM (or GUISIM). If the nominal aimpoint has been previously biased, a velocity correction $\Delta V_{RBj}$ required to remove that bias is computed prior to returning. But if $P_{\text{II}}$ exceeds $P_{\text{I}}$, an aimpoint bias $\delta \hat{\mu}_j$ and the associated bias velocity correction $\Delta V_{ij}$ must be computed. Before describing the details of the biasing technique it is necessary to define the relationship between $\Delta V_j$ and $\delta \hat{\mu}_j$ for linear midcourse guidance policies.

Linear impulsive guidance policies have form

$$\Delta V_j = \Gamma_j \delta \hat{\mu}_j$$

(15)
where $\Gamma_j$ is the guidance matrix and $\delta \mathbf{x}_j$ is the spacecraft state deviation from the targeted nominal trajectory. (These guidance policies are discussed in more detail in the subroutine GUIA analysis section.) Such guidance policies can be readily generalized to account for changes in the target conditions from their nominal values. This generalized version of equation (13) has form

$$\Delta \mathbf{v}_j = \Gamma_j \delta \mathbf{x}_j + \Psi_j \delta \mu_j$$

(14)

where $\Psi_j$ can also be referred to as a guidance matrix. For the purposes of the BLAPM analysis, we shall assume that $\delta \mu_j$ in equation (14) is always an aimpoint change in the $i, j, k, \ell$ plane. Thus, $\Psi_j$ will be a $3 \times 2$ guidance matrix. The derivation of the $\Psi_j$ matrix is quite similar to the derivation of the $\Gamma_j$ matrix and will not be presented here. If we partition the previously discussed variation matrix $\eta_j$ as follows:

$$\eta_j = \begin{bmatrix} \eta_{1j} \\ \eta_{2j} \end{bmatrix}$$

(15)

then the $\Psi_j$ matrices for the three micaceous guidance policies are given by the following equations:

(a) 2VBP : \[ \Psi_j = \eta_j^T \left( \eta_j \eta_j^T \right)^{-1} \]

(b) 3VBP : \[ \Psi_j = \eta_j^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \]

(c) FTA : \[ \Psi_j = \eta_j^{-1} \eta_j^T \]

(16)

(17)

(18)

If an aimpoint bias were to be removed at time $t_j$, the required velocity correction would be given by

$$\Delta \mathbf{v}_{RB_j} = -\Psi_j \delta \mu_j$$

(19)

If an aimpoint bias were to be imparted at time $t_j$, the bias velocity correction would be given by

$$\Delta \mathbf{v}_{B_j} = \Psi_j \delta \mu_j$$

(20)
If an aimpoint bias \( \delta \mu_j^{(1)} \) had been previously imparted, and if a new aimpoint bias \( \delta \mu_j^{(2)} \) is to be imparted, then the total bias velocity correction would be given by

\[
\Delta \theta_{b,j} = \psi_j \left[ \delta \mu_j^{(2)} - \delta \mu_j^{(1)} \right]
\]  

(21)

The general statement of the biased aimpoint guidance problem is as follows: Find an aimpoint \( \hat{\mu}_j \) in the impact plane which satisfies the impact probability constraint

\[
P_{II} \leq P_I
\]  

(22)

and minimizes a performance functional having form

\[
J = \left( \hat{\mu}_j - \mu^* \right)^T \hat{\Lambda} \left( \hat{\mu}_j - \mu^* \right)
\]  

(23)

where \( \mu^* \) is the nominal aimpoint and \( \hat{\Lambda} \) is a constant symmetric matrix that will be defined subsequently.

The solution of this problem is detailed in the section on biased aimpoint guidance in the analytical manual. Only the results will be presented here. The assumption of constant probability density over the target planet capture area permits us to rewrite constraint equation (22) as

\[
\lambda_1 \mu_1^2 + 2\lambda_3 \mu_1 \mu_2 + \lambda_2 \mu_2^2 = c^2
\]  

(24)

where

\[
c^2 = 2 \ln \left[ \frac{R_e^2}{2 | \Lambda^{1/2} P_I |} \right]
\]  

(25)

and \( \hat{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \) and \( \Lambda^{-1} = \begin{bmatrix} \lambda_1 & \lambda_3 \\ \lambda_3 & \lambda_2 \end{bmatrix} \).

The inequality has been replaced by an equality since the solution can be shown to lie on the constraint boundary, which, from inspection of equation (24) is an ellipse centered at the target planet.

If \( t_j \) is the time of the final midcourse correction, matrix \( \hat{\Lambda} \) will be chosen as a 2x2 identity matrix. The minimization of \( J \) is then equivalent to minimization of the miss distance \( | \hat{\mu}_j - \mu^* | \). If \( t_j \) is not the final midcourse correction time, \( \hat{\Lambda} \) will be defined as follows:
\[
\mathbf{A} = \psi_j^{T} \psi_{j+1}
\]  

(26)

Here \(\psi_{j+1}\) denotes the aimpoint guidance matrix for the next midcourse correction occurring at time \(t_{j+1}\). In this case the minimization of \(J\) is equivalent to the minimization of \(\Delta V_{RB, j+1}\), i.e., the velocity required to remove bias \(\delta \mu_j\) at time \(t_{j+1}\) will be minimized. The computation of \(\psi_{j+1}\) is based on the variation matrix \(\eta_{j+1}\), just as \(\psi_j\) was based on \(\eta_j\). However, \(\eta_{j+1}\) can be computed more efficiently by using the relationship

\[
\eta_{j+1} = \eta_{j} \Phi_{j+1, j}
\]

(27)

where \(\Phi_{j+1, j}\) is the state transition matrix over \([t_j, t_{j+1}]\).

If we define

\[
\mathbf{A} = \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix}
\]

then the necessary condition for a minimum is given by

\[
\begin{align*}
&\left(a_1 \lambda_3 - a_3 \lambda_1\right) \mu_1 + \left(a_3 \lambda_2 - a_2 \lambda_3\right) \mu_2 + \left(a_1 \lambda_2 - a_2 \lambda_1\right) \mu_1 \mu_2 \\
&+ \left(-a_1 \lambda_3 \mu_1^* - a_3 \lambda_3 \mu_2^* + a_3 \lambda_1 \mu_1^* + a_2 \lambda_1 \mu_2^*\right) \mu_1 \\
&+ \left(-a_1 \lambda_2 \mu_1^* - a_3 \lambda_2 \mu_2^* + a_3 \lambda_3 \mu_1^* + a_2 \lambda_3 \mu_2^*\right) \mu_2 = 0
\end{align*}
\]

(28)

Thus, our problem is reduced to finding \(\mu_1\) and \(\mu_2\) which satisfy equations (24) and (28). Since the analytical solution of these equations proved intractable, a standard Newton iteration technique is employed in BLAIM which quickly converges to solutions for \(\mu_1\) and \(\mu_2\). The iteration process is started with an initial guess defined as the intersection of the extended \(\vec{d}^*\) vector and the constraint boundary defined by equation (24). This initial guess is given by
\[ \mu_1^0 = \left( \frac{\mu_1}{\mu_2^*} \right) \mu_2^0 \]  

(29)

\[ \mu_2^0 = \text{sgn} (\mu_2^*) \frac{c}{\sqrt{\left(\frac{\mu_1^*}{\mu_2^*}\right)^2 + 2 \lambda_3 \left(\frac{\mu_1^*}{\mu_2^*}\right) + \lambda_2}} \]

where \( c \) is defined by equation (25).

In addition to the previously described iteration process, subroutine BIAD also employs an outer iteration loop which accounts for the dependence of \( \sigma_j \) (equation (6)) on \( \delta \mu_j \). The execution error covariance \( \sigma_j \) is a function of the total velocity correction at \( t_j \), but the total velocity correction, in particular \( \Delta \dot{V}_{Bj} \), depends on \( \delta \mu_j \). This coupling is resolved by recomputing \( \sigma_j \) at the end of the previously described biasing technique and repeating the biasing cycle until the error function

\[ |P \Omega - P_i| \leq P_i \times 10^{-3} \]

is satisfied. This outer iteration process is not performed, however, if \( t_j = \) injection time since at injection equation (6) is replaced by the equation

\[ P_{c,j} = P_{k,j} \]

and \( \sigma_j \) is always zero.

BIAFM Flow Chart

ENTER

Initialize iteration counter ITRN and define guidance policy code IIIGP.

NO

IIIGP>1?

YES

Compute FTA transformation matrix A.

Compute nominal FTA aimpoint $\vec{\mu}^*$ and most recent FTA aimpoint $\vec{\mu}$.

Set capture radius $R_c$ equal to the physical radius of planet.

Compute $\psi_j$ matrix and $\delta \vec{\mu} = \vec{\mu} - \vec{\mu}^*$.

Write out $R_c$ and $\psi_j$.

YES Has a bias been previously imparted?

NO

YES

Write out bias $\delta \vec{\mu}$.

Compute velocity correction required to remove bias $\Delta V_{RB}$.

$\Delta V_{RB} = 0$.

NO

$\Delta V_N$ to zero.

YES

$t_j = $ injection time?

Set $Q_j^*$ and

63

67

113
Set $\Delta \vec{V}_{TOT} = \Delta \vec{V}_{RB} + \Delta \vec{V}_{N}$

Call QGAMP to compute execution error covariance $\bar{\sigma}_j^* = \bar{\sigma}_j(\Delta \vec{V}_{TOT})$.

Write out $\Delta \vec{V}_{RB}$ and $\bar{\sigma}_j^*$.

Simulation mode = ?

Set $\Delta \vec{V}_{UP} = \Delta \vec{V}_{RB} + \Delta \vec{V}_{N}$

Set $\Delta \vec{V}_{UP} = \Delta \vec{V}_{RB}^*$

Call QGICOM to compute and write out $\Lambda_j^*$ and $\Phi$ for the nominal aimpoint.

$\Phi I \leq P$ ?

Set $\bar{\mu} = \bar{\mu}^*$.
Set IBIAS = 0.

RETURN

Will a velocity bias be actually imparted?

NO

Set IBIAS = 1.

YES
Set $A = I$.

YES

t$_j$ = time of final midcourse correction?

NO

Set $\Delta t = t_{j+1} - t_j$ and call PSIM to compute the state transition matrix $\phi(t_{j+1}, t_j)$.

Call MATIN to compute $\phi^{-1}(t_{j+1}, t_j)$.

Compute variation matrix $\eta_{j+1}$.

Compute $\psi_{j+1}$ matrix for the appropriate guidance policy.

Compute $\tilde{A} = \psi_{j+1}^T \psi_{j+1}$.

YES

IDENS $\neq 1$?

NO

Write: IDENS OF ION NOT AVAILABLE.

EXIT

Compute constants defining the elliptical constraint boundary associated with $A_j$.

B
Write out the equation of the elliptical constraint boundary.

\( \text{ITRM} > 0 \) ?

NO

Compute and write out the initial guess \( \mu^0 \).

Initialize Newton iteration counter \( \text{IXNT} \).

Compute constants defining necessary condition for a minimum.

Use Newton iteration technique to determine the \( i \)-th iterate \( \delta \mu^i \).

Write out \( \bar{\mu}^i \) and \( \delta \mu^i \).

Set \( \bar{\mu}^{i+1} = \bar{\mu}^i + \delta \mu^i \) and increment \( \text{IXNT} \).

\(| \delta \mu_{1}^{i} | + | \delta \mu_{2}^{i} | < 1 \) ?

NO

\( \text{IXNT} > 25 \) ?

YES

Write: Newton's method did not converge in BLAIM.

NO

5000
Write out final iteration $\mu^{i+1}$.

Compute $\delta \mu = \mu^{i+1} - \mu^*$ and

$$\Delta \nu_{\text{bias}} = \psi_j \delta \mu$$

and write out.

Store $\mu^{i+1}$ in the XM array.

$t_j = \text{injection time?}$

YES

Set

$$\Delta \nu_{\text{UP}} = \Delta \nu_{\text{UP}} + \Delta \nu_{\text{bias}}$$

and

$$\Delta \nu_{\text{TOT}} = \Delta \nu_{\text{TOT}} + \Delta \nu_{\text{bias}}$$

Call QCMP to compute $\tilde{Q}_j = Q_j (\Delta \nu_{\text{TOT}})$

and write out.

Call P0ICOM to compute $\Delta j$ and $\Phi I$ for aimpoint $\mu^{i+1}$.

Update iteration counter ITRN.

Yes

$|\Phi I - P_i| \leq P_i \times 10^{-3}$?

YES

ITRN = 5?

YES

301

NO

ITRN = 5?

NO

175

YES

5000

Write out final iteration $\mu^{i+1}$.

Compute $\delta \mu = \mu^{i+1} - \mu^*$ and

$$\Delta \nu_{\text{bias}} = \psi_j \delta \mu$$

and write out.

Store $\mu^{i+1}$ in the XM array.

$t_j = \text{injection time?}$

NO

Set

$$\Delta \nu_{\text{UP}} = \Delta \nu_{\text{UP}} + \Delta \nu_{\text{bias}}$$

and

$$\Delta \nu_{\text{TOT}} = \Delta \nu_{\text{TOT}} + \Delta \nu_{\text{bias}}$$

Call QCMP to compute $\tilde{Q}_j = Q_j (\Delta \nu_{\text{TOT}})$

and write out.

Call P0ICOM to compute $\Delta j$ and $\Phi I$ for aimpoint $\mu^{i+1}$.

Update iteration counter ITRN.

$|\Phi I - P_i| \leq P_i \times 10^{-3}$?

YES

NO

ITRN = 5?

NO

301
301

Set $\Delta \vec{V}_{UP} = \Delta \vec{V}_{UP}$.

RETURN

1555

Call PMICM to validate P31 at injection.
Set $\Delta \vec{V}_{UP} = \Delta \vec{V}_{bias}$.

Set $\hat{\mu} = \hat{\mu}_d+1$. 
SUBROUTINE BIAS

PURPOSE: COMPUTE THE ACTUAL MEASUREMENT BIAS IN THE SIMULATION PROGRAM
RETURN THE ACTUAL MEASUREMENT BIAS TO BE USED IN THE SIMULATION MODE.

CALLING SEQUENCE: CALL BIAS(MCODE,BVAL)

ARGUMENTS:
BVAL  O THE ACTUAL BIAS TO BE USED IN THE MEASUREMENT
MCODE  I MEASUREMENT TYPE CODE

SUBROUTINES SUPPORTED: SIMULL

COMMON USED: BIA
BIAS Analysis

The actual measurement $I_k^{a}$ at time $t_k$ is given by

$$I_k^{a} = I_k + b_k + \nu_k$$

where $I_k$ is the ideal measurement, which would be made in the absence of instrumentation errors, $b_k$ is the actual measurement bias and $\nu_k$ represents the actual measurement noise.

The function of subroutine BIAS is to compute the measurement bias $b_k$ for the appropriate measurement type. The constant biases for all measurement devices are stored in the vector BIA. Subroutine BIAS selects the appropriate elements from this vector to construct the actual measurement bias.
BIAS Flow Chart

ENTER

MC@DE = ?

Compute the bias vector for 3 star-planet angle measurements.

RETURN

NO

MC@DE even?

YES

MC@DE even?

Compute the bias for a range-rate measurement from the appropriate station.

RETURN

Compute the bias vector for a range and range-rate measurement from the appropriate station.

A

1,2,3,4,5,6,7,8

10,11,12,13

MC@DE = ?

10

11,12,13

MC@DE = ?

1,2,3,4,5,6,7,8

Compute the bias for the appropriate star-planet angle measurement.

RETURN

Compute the bias for an apparent planet diameter measurement.

RETURN
BLOCK DATA
PURPOSE: TO LOAD CONSTANTS INTO COMMON LOCATIONS USED IN VARIOUS
OTHER PARTS OF THE PROGRAM.
CALLING SEQUENCE: NONE
ARGUMENTS: NONE
SUBROUTINES SUPPORTED: HALF THE SUBROUTINES USE THE CONSTANTS
STORED BY THIS SUBROUTINE
SUBROUTINES REQUIRED: NONE
COMMON LOADED: CN1, CN, ELMNT, EWN, EVNH, RADIUS, RAD, RMASS, SMJR, SPHERE,
F, NNNAME, PI, PLANET, PHASS, ST
BLKDAT Analysis

Subroutine BLKDAT is responsible for setting up constants used in computing ephemeris data for the gravitating bodies.

The arrays set up by BLKDAT and their definitions are as follows:

<table>
<thead>
<tr>
<th>Array</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN(80)</td>
<td>Constants defining mean elements for inner planets</td>
</tr>
<tr>
<td>ST(50)</td>
<td>Constants defining mean elements for outer planets</td>
</tr>
<tr>
<td>SMJR(18)</td>
<td>Constants defining semi-major axes for planets and moon</td>
</tr>
<tr>
<td>EMN(15)</td>
<td>Constants defining lunar elements</td>
</tr>
<tr>
<td>PMASS(11)</td>
<td>Gravitational constants of sun, planets, and moon</td>
</tr>
<tr>
<td>MASS(11)</td>
<td>Mass of bodies relative to sun</td>
</tr>
<tr>
<td>RADIUS(11)</td>
<td>Surface radii of sun, planets, and moon</td>
</tr>
<tr>
<td>SPHERE(11)</td>
<td>Sphere of influence radii of sun, planets, and moon</td>
</tr>
<tr>
<td>MOUTH(12)</td>
<td>Names of months for output purposes</td>
</tr>
<tr>
<td>PLANET(11)</td>
<td>Names of planets for output purposes</td>
</tr>
</tbody>
</table>

The definitions of the CN, ST, SMJR, and EMN arrays are provided in Tables 2 through 5 on the following page. The actual constants stored in those arrays are the ephemeris data listed on the next pages following.

The constants stored in the other arrays are given below.

<table>
<thead>
<tr>
<th>Body</th>
<th>EMASS (AU^3/day^2)</th>
<th>EMASS*</th>
<th>RADIUS (AU)</th>
<th>SPHERE (AU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>2.959120083(-4)</td>
<td>1.0</td>
<td>4.66582(-3)</td>
<td>NA</td>
</tr>
<tr>
<td>Mercury</td>
<td>4.850(-11)</td>
<td>1.639(-7)</td>
<td>1.617(-5)</td>
<td>7.46(-4)</td>
</tr>
<tr>
<td>Venus</td>
<td>7.243(-10)</td>
<td>2.448(-6)</td>
<td>4.044(-5)</td>
<td>4.12(-3)</td>
</tr>
<tr>
<td>Earth</td>
<td>8.88757(-10)</td>
<td>3.003(-6)</td>
<td>4.263(-5)</td>
<td>6.18(-3)</td>
</tr>
<tr>
<td>Mars</td>
<td>9.5497905(-11)</td>
<td>3.236(-7)</td>
<td>2.279(-5)</td>
<td>3.78(-3)</td>
</tr>
<tr>
<td>Jupiter</td>
<td>2.8252(-7)</td>
<td>9.547(-4)</td>
<td>4.7727(-4)</td>
<td>.3216</td>
</tr>
<tr>
<td>Saturn</td>
<td>8.454(-8)</td>
<td>2.857(-4)</td>
<td>4.0374(-4)</td>
<td>.3246</td>
</tr>
<tr>
<td>Uranus</td>
<td>1.290(-8)</td>
<td>4.359(-5)</td>
<td>1.5761(-4)</td>
<td>.346</td>
</tr>
<tr>
<td>Neptune</td>
<td>1.5(-8)</td>
<td>5.069(-5)</td>
<td>1.4906(-4)</td>
<td>.5805</td>
</tr>
<tr>
<td>Pluto</td>
<td>7.6(-10)</td>
<td>2.501(-6)</td>
<td>4.679(-5)</td>
<td>.2366</td>
</tr>
<tr>
<td>Moon</td>
<td>1.0921748(-11)</td>
<td>3.696(-8)</td>
<td>1.161(-5)</td>
<td>3.71394(-4)</td>
</tr>
</tbody>
</table>

* Truncated from program values
### Array Definitions

<table>
<thead>
<tr>
<th>Constant</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>( a_0 )</th>
<th>( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Venus</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Earth</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Mars</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Jupiter</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Saturn</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Uranus</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Neptune</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Pluto</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Moon</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** ELMNT Array -- Conic Elements

<table>
<thead>
<tr>
<th>Constant</th>
<th>( i_0 )</th>
<th>( i_1 )</th>
<th>( i_2 )</th>
<th>( i_3 )</th>
<th>( \Omega_0 )</th>
<th>( \Omega_1 )</th>
<th>( \Omega_2 )</th>
<th>( \Omega_3 )</th>
<th>( \varpi_0 )</th>
<th>( \varpi_1 )</th>
<th>( \varpi_2 )</th>
<th>( \varpi_3 )</th>
<th>( e_0 )</th>
<th>( e_1 )</th>
<th>( e_2 )</th>
<th>( e_3 )</th>
<th>( H_0 )</th>
<th>( H_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>Venus</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>Earth</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
</tr>
<tr>
<td>Mars</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
</tr>
</tbody>
</table>

**Table 2.** SMJR Array

**Table 3.** CM Array -- Inner Planet Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>( i_0 )</th>
<th>( i_1 )</th>
<th>( \Omega_0 )</th>
<th>( \Omega_1 )</th>
<th>( \varpi_0 )</th>
<th>( \varpi_1 )</th>
<th>( e_0 )</th>
<th>( e_1 )</th>
<th>( H_0 )</th>
<th>( H_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jupiter</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Saturn</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Uranus</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>Neptune</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>Pluto</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
</tr>
</tbody>
</table>

**Table 4.** ST Array -- Outer Planet Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>( \Omega_0 )</th>
<th>( \Omega_1 )</th>
<th>( \varpi_0 )</th>
<th>( \varpi_1 )</th>
<th>( L_0 )</th>
<th>( L_1 )</th>
<th>( L_2 )</th>
<th>( L_3 )</th>
<th>( i )</th>
<th>( a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moon</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.** EMN Array -- Lunar Constants
Planetary and Lunar Ephemerides

Mean Elements of Mercury

\[ i = 0.1222233228 + 3.24776685 \times 10^{-3} T - 3.199770295 \times 10^{-7} T^2 \]
\[ \Omega = 0.8228518959 + 2.068578774 \times 10^{-2} T + 3.034933644 \times 10^{-6} T^2 \]
\[ \tilde{\Omega} = 1.3246996178 + 2.714840259 \times 10^{-2} T + 5.143873156 \times 10^{-6} T^2 \]
\[ e = 0.20561421 + 0.00002046 T - 0.000000030 T^2 \]
\[ M = 1.785111955 + 7.142471000 \times 10^{-2} d + 8.72664626 \times 10^{-9} D^2 \]
\[ a = 0.3870986 \text{ A.U.} = 57,909,370 \text{ km} \]

Mean Elements of Venus

\[ i = 0.0592300268 + 1.7555510339 \times 10^{-5} T - 1.696847884 \times 10^{-8} T^2 \]
\[ \Omega = 1.3226043500 + 1.570534527 \times 10^{-2} T + 7.155849933 \times 10^{-6} T^2 \]
\[ \tilde{\Omega} = 2.2717874591 + 2.457486613 \times 10^{-2} T + 1.704120089 \times 10^{-3} T^2 \]
\[ e = 0.00682069 - 0.00004774 T + 0.000000091 T^2 \]
\[ M = 3.710626172 + 2.796244623 \times 10^{-2} d + 1.682497399 \times 10^{-6} D^2 \]
\[ a = 0.7233316 \text{ A.U.} = 108,209,322 \text{ km} \]

Mean Elements of Earth

\[ i = 0 \]
\[ \Omega = 0 \]
\[ \tilde{\Omega} = 1.7666368138 + 3.000526417 \times 10^{-2} T + 7.902463002 \times 10^{-6} T^2 \]
\[ + 5.817764173 \times 10^{-8} T^3 \]
\[ e = 0.01675104 - 0.00004180 T - 0.000000126 T^2 \]
\[ M = 6.256383781 + 1.720196977 \times 10^{-2} d - 1.954768762 \times 10^{-7} D^2 \]
\[ - 1.22173047 \times 10^{-9} D^3 \]
\[ a = 1.0000003 \text{ A.U.} = 149,598,530 \text{ km} \]

Mean Elements of Mars

\[ i = 0.0322944089 - 1.178097245 \times 10^{-5} T + 2.201054112 \times 10^{-7} T^2 \]
\[ Q = 0.8314840375 + 1.345634309 \times 10^{-2} T - 2.424068406 \times 10^{-6} T^2 \\
- 9.308422677 \times 10^{-8} T^3 \]

\[ \bar{Q} = 5.8332085089 + 3.212729363 \times 10^{-2} T + 2.266503959 \times 10^{-6} T^2 \\
- 2.084698829 \times 10^{-8} T^3 \]

\[ a = 0.09331290 + 0.000092064 T - 0.000000077 T^2 \]

\[ M = 5.576840523 + 9.145887726 \times 10^{-3} d + 2.365444735 \times 10^{-7} D^2 \\
+ 4.363323130 \times 10^{-10} D^3 \]

\[ a = 1.5236915 \text{ A.U.} = 227,941,963 \text{ km} \]

**Mean Elements of Jupiter**

\[ i = 0.0228410270 - 9.696273622 \times 10^{-5} T \]

\[ \bar{Q} = 1.7355180770 + 1.764479392 \times 10^{-2} T \]

\[ \bar{Q} = 0.2218561704 + 2.812302353 \times 10^{-2} T \]

\[ a = 0.0483376 + 0.00016302 T \]

\[ M = 3.93135411 + 1.450191928 \times 10^{-3} d \]

\[ a = 5.202803 \text{ A.U.} = 778,331,525 \text{ km} \]

**Mean Element of Saturn**

\[ i = 0.0435037861 - 7.757018898 \times 10^{-8} T \]

\[ \bar{Q} = 1.9684445802 + 1.523977870 \times 10^{-2} T \]

\[ \bar{Q} = 1.5897996653 + 3.419861162 \times 10^{-2} T \]

\[ a = 0.0558900 - 0.00034705 T \]

\[ M = 3.0426210430 + 5.837120844 \times 10^{-4} d \]

\[ a = 9.538843 \text{ A.U.} = 1,426,996,160 \text{ km} \]

**Mean Elements of Uranus**

\[ i = 0.0134865470 + 0.696273622 \times 10^{-6} T \]

\[ \bar{Q} = 1.2826407705 + 8.912087493 \times 10^{-3} T \]

\[ \bar{Q} = 2.9502426085 + 2.834608631 \times 10^{-2} T \]
\[ e = 0.0470463 + 0.00027204 \ T \]
\[ M = 1.2843599198 + 2.046548840 \times 10^{-4} \ d \]
\[ a = (19.182281 - 0.00037008 \ T) \text{ A.U.} = (2.869,640,310 - 88271 \ T) \text{ km} \]

**Mean Elements of Neptune**

\[ i = 0.0310537707 - 1.399885148 \times 10^{-4} \ T \]
\[ \Omega = 2.2810642235 + 1.923032859 \times 10^{-2} \ T \]
\[ \varpi = 0.7638202701 + 1.532704516 \times 10^{-2} \ T \]
\[ e = 0.00832849 + 0.00007701 \ T \]
\[ M = 0.7204851506 + 1.033089473 \times 10^{-4} \ d \]
\[ a = (30.057053 + 0.001210166 \ T) \text{ A.U.} = (4,496,490,000 + 181039 \ T) \text{ km} \]

**Mean Elements of Pluto**

\[ i = 0.2996706970859694 \]
\[ \Omega = 1.1914337550102258 \]
\[ \varpi = 3.9099919302791948 \]
\[ e = 0.2488033053623924 \]
\[ M = 3.993890007 + 0.6962635708298997 \times 10^{-4} \]
\[ a = 39.37364135300176 \text{ A.U.} = 5,890,213,786,145,730 \text{ km} \]

**Mean Elements of Moon**

\[ i = 5.1453964^\circ \]
\[ \Omega = 259.183275^\circ - 0.0529539222d + 0.0002078 \ T^{2} + 0.000002 \ T^{3} \]
\[ \varpi = 334.329556^\circ + 0.1114040803d - 0.010325 \ T^{2} - 0.000012 \ T^{3} \]
\[ L = 270.434164^\circ - 13.1763965268d - 0.001133 \ T^{2} + 0.000019 \ T^{3} \]
\[ a = 0.0256954448 \text{ A.U.} \]
\[ e = 0.054900489 \]
Note 1: The above elements are referred to the mean equinox and ecliptic of date except for Pluto.

Note 2: The elements for Pluto are oscillating values for epoch 1960 September 23.0 E.T. = J.D. 2437200.5

Note 3: The time interval from the epoch is denoted by T when measured in Julian centuries of 36,525 ephemeris days, by D = 3.6525 T when measured in units of 10,000 ephemeris days, and by d = 10,000D = 36,525 T when measured in ephemeris days. Times are measured with respect to the epoch 1900 January 0.5 E.T. = J.D. 2415020.0.

Note 4: Angular relations are expressed in radians for planets and degrees for moon.

References:
SUBROUTINE CAREL

PURPOSE: TRANSFORM CARTESIAN COORDINATES TO CONIC ELEMENTS


ARGUMENTS:
GM 1 GRAVITATIONAL CONSTANT OF THE CENTRAL BODY
R(V) 1 POSITION VECTOR RELATIVE TO CENTRAL BODY
V(V) 1 VELOCITY VECTOR RELATIVE TO CENTRAL BODY
TFP 0 TIME OF FLIGHT FROM PERIAPSIS ON THE CONIC
A 0 SEMI-MAJOR AXIS OF THE CONIC
E 0 ECCENTRICITY OF THE CONIC
M 0 ARGUMENT OF PERIAPSIS OF THE CONIC
XI 0 INCLINATION OF THE CONIC TO THE REFERENCE FRAME
XN 0 LONGITUDE OF THE ASCENDING NODE OF THE CONIC
TA 0 INSTANTANEOUS TRUE ANOMALY OF THE CONIC
PP(V) 0 UNIT VECTOR TOWARD PERIAPSIS ON CONIC
QQ(V) 0 UNIT VECTOR NORMAL TO PP IN ORBITAL PLANE
WM(V) 0 UNIT VECTOR NORMAL TO ORBITAL PLANE

SUBROUTINES SUPPORTED: TAROPT LUNCON MULTAR EXCUTE COPINS NONINS OPROP WMP GUISIM NONLIN PULSEX GUIDM

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AUXF ECCENTRIC ANOMALY (HYPERBOLIC CASE)
AVA MEAN ANOMALY (ELLIPITC CASE)
COSEA COSINE OF THE ECCENTRIC ANOMALY (ELLIPITC CASE)
CTA COSINE OF THE TRUE ANOMALY
C MAGNITUDE OF THE ANGULAR MOMENTUM
DIV INTERMEDIATE VARIABLE IN CALCULATION OF ECCENTRIC ANOMALY
EA  ECCENTRIC ANOMALY (ELLIPTIC CASE)
P  SEMI-LATUS Rectum of the Conic
RAD DEGREES TO RADIANS CONVERSION CONSTANT
RD  TIME DERIVATIVE OF RADIUS
RM  MAGNITUDE OF CARTESIAN POSITION VECTOR
SINEA SINE OF THE ECCENTRIC ANOMALY (ELLIPTIC CASE)
SINHF HYPERBOLIC SINE OF AUXF
STA SINE OF THE TRUE ANOMALY
TANG INTERMEDIATE VARIABLE USED TO CALCULATE SINHF
VM  MAGNITUDE OF THE CARTESIAN VELOCITY VECTOR
Z  INTERMEDIATE VECTOR USED TO CALCULATE PP, QQ VECTORS
CAREL Analysis

CAREL converts the cartesian state (position and velocity) of a massless point referenced to a gravitational body to the equivalent conic elements about that body.

Let the cartesian state be denoted $\vec{r}, \vec{v}$ and let the gravitational constant of the central body be $\mu$.

The angular momentum constant $c$ is

$$c = |\vec{r} \times \vec{v}|$$

(1)

The unit normal $\hat{\omega}$ to the orbital plane is

$$\hat{\omega} = \frac{\vec{r} \times \vec{v}}{c}$$

(2)

The semilatus rectum $p$ is

$$p = \frac{c^2}{\mu}$$

(3)

The semi-major axis $a$ is

$$a = \frac{r}{2 - \frac{r^2}{\mu}}$$

(4)

Thus $a > 0$ for elliptical motion, $a < 0$ for hyperbolic motion. The eccentricity $e$ is

$$e = \sqrt{1 - \frac{p}{a}}$$

(5)

Thus $e < 1$ for elliptical motion, $e > 1$ for hyperbolic motion. The inclination of the orbit $i$ is computed from

$$\cos i = \hat{\omega}_z$$

(6)

The longitude of the ascending node $\Omega$ is defined by

$$\tan \Omega = \frac{\hat{\omega}_x}{-\hat{\omega}_y}$$

(7)
The true anomaly $f$ at the given state is computed from

$$\cos f = \frac{p - \mu}{r} \quad \sin f = \frac{c \cdot f}{\mu e} \quad (8)$$

Now define an auxiliary vector $\hat{c}$ by

$$\hat{c} = S \hat{v} - \frac{1}{e} \cdot \hat{r} \quad (9)$$

Then $\hat{p}$, the unit vector to periapsis, and $\hat{q}$, the in-plane normal to $\hat{p}$, are defined by

$$\hat{p} = \hat{c} \cos f - \hat{c} \sin f \quad (10)$$

$$\hat{q} = \hat{c} \sin f + \hat{c} \cos f \quad (11)$$

where $\hat{c} = \frac{\mathbf{r}}{r}$. The argument of periapsis $\omega$ is then computed from

$$\tan \omega = \frac{\hat{p}}{\hat{q} \cdot z} \quad (12)$$

The conic time from periapsis $t_p$ is computed from different formulae depending upon the sign of the semi-major axis. For $a > 0$ (elliptical motion)

$$t_p = \sqrt{\frac{a^3}{\mu}} \cdot (E - e \sin E)$$

$$\cos E = \frac{e + \cos f}{1 + e \cos f} \quad \sin E = \frac{\sqrt{1 - e^2 \sin f}}{1 + e \cos f} \quad (13)$$

For $a < 0$ (hyperbolic motion) the time from periapsis is

$$t_p = \sqrt{\frac{a^3}{\mu}} \cdot (e \sinh H - H)$$

$$\tanh \frac{H}{2} = \sqrt{\frac{e - 1}{e + 1}} \cdot \tan \frac{f}{2} \quad (14)$$

SUBROUTINE CASCAD

PURPOSE: TO COMPUTE THE STATE TRANSITION MATRIX DEFINING STATE PERTURBATIONS OVER AN ARBITRARY TIME INTERVAL BY CAS- CADING DANBY MATRIZANTS OVER SEGMENTS OF THE INTERVAL USING EITHER PATCHED CONIC OR VIRTUAL MASS TWO BODY FORMULAE.

CALLING SEQUENCE: CALL CASCAD(RI, SMAT)

ARGUMENTS

RI I POSITION AND VELOCITY OF VEHICLE AT BEGINNING OF TIME INTERVAL

SMAT O STATE TRANSITION MATRIX OVER DESIRED INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: CONC2 VMP

LOCAL SYMBOLS

DELTAT TIME INTERVAL USED IN A SINGLE ITERATE

DELT TIME INTERVAL OF CURRENT PROPAGATION

D1 INITIAL TIME OF ITERATE

IFLAG FLAG TO DETERMINE WHETHER ITERATION IS COMPLETED

IOS FLAG INDICATING HELIOCENTRIC OR PLANETOCENTRIC PHASE

ISP3 FLAG USED AS TRAJECTORY INTEGRATION SPHERE OF INFLUENCE STOPPING CODE

PHI CUMULATIVE STATE TRANSITION MATRIX OVER INTERVAL (T0, TK)

PSI CUMULATIVE STATE TRANSITION MATRIX OVER INTERVAL (T0, TK+1)

PTP STATE OF TARGET RELATIVE TO INERTIAL Coordinate At TIME TK

RAV STATE OF SPACECRAFT RELATIVE TO DOMINANT BODY FOR MATRIZANT

RH0 STATE TRANSITION MATRIX OVER INTERVAL (TK, TK+1)

RS INERTIAL SPACECRAFT STATE AT TK

RSF INERTIAL SPACECRAFT STATE AT TK+1
R1  SPACECRAFT STATE RELATIVE TO VIRTUAL MASS AT TK
R2  SPACECRAFT STATE RELATIVE TO VIRTUAL MASS AT TK+1
SUM  INTERMEDIATE VARIABLE
TIME  CUMULATIVE TRAJECTORY TIME FROM INITIAL TIME TO TK+1
XNU  VIRTUAL MASS MAGNITUDE AT TK
YNU  VIRTUAL MASS MAGNITUDE AT TK+1

COMMON COMPUTED:
  ICL

COMMON USED:
  ACC  ALNGTH  DATEJ  DELTM  DTPLAN
  DTSUM  ISTM1  NTP  PHASS  RTP
  RVS  TM  TRTM1  VMU  V
CASCAD Analysis

CASCAD approximates the state transition matrix \( \Phi_{f,0} \) defining state perturbations over an arbitrary interval \([t_0, t_f]\) by recursively computing state transition matrices over intervals \([t_0, t_1], [t_0, t_2], \ldots \), \([t_0, t_f]\).

The recursive formula for the \( k+1 \) iteration based on the \( k \)-th iteration is given by

\[
\Phi_{k+1,0} = \Psi_{k+1,k} \Phi_{k,0}
\]

where \( \Psi_{k+1,k} \) is the state transition matrix for the \( k+1 \)-st interval \([t_k, t_{k+1}]\).

The time interval \( \Delta t_{k+1} = t_{k+1} - t_k \) is determined by the position vector \( \vec{x}_k \) of the spacecraft relative to the target planet along the nominal \( n \)-body trajectory at the time \( t_k \). Then if \( R_{SOI} \) denotes the radius of the sphere of influence of the target planet the time interval is defined by

\[
\Delta t_{k+1} = \Delta t_{\text{planet}} \quad \text{if} \quad r_k \leq R_{SOI}
\]

\[
= \Delta t_{\text{sun}} \quad \text{if} \quad r_k > R_{SOI} \quad \text{and} \quad \text{n-body nominal trajectory propagated over } \Delta t_{\text{sun}} \quad \text{does not intersect the SOI.}
\]

\[
= \Delta t_{SOI} \quad \text{if} \quad r_k > R_{SOI} \quad \text{and} \quad \text{n-body nominal trajectory intersects the SOI after the time interval } \Delta t_{SOI} \quad \text{where } \Delta t_{SOI} \Delta t_{\text{sun}}.
\]

where \( \Delta t_{\text{planet}} \) and \( \Delta t_{\text{sun}} \) are input parameters. For the last interval \( \Delta t_{n} \) partial step may be required so that \( \Delta t_n = t_f - t_{n-1} \).

The \( \Psi_{k+1,k} \) matrix may be computed by either of two models. In the patch conic model the position and velocity vectors \( \vec{r}_k, \vec{v}_k \) of the spacecraft relative to the dominant body (the sun if \( \Delta t_{k+1} = \Delta t_{\text{sun}} \) or \( \Delta t_{SOI} \), the target planet if \( \Delta t_{k+1} = \Delta t_{\text{planet}} \)) at the time \( t_k \) is used to define a
conic with respect to the dominant body and the Danby matrization over the given interval defines $\Psi_{k+1,k}$ (CASC2).

In the virtual mass model the position and velocity vectors $\mathbf{r}_k, \mathbf{v}_k$ are computed relative to the virtual mass and the gravitational constant used is that of the virtual mass magnitude at the time $t_k$. The Danby matrization corresponding to this conic then is used to compute $\Psi_{k+1,k}$ (CASC2).

The recursive process continues until the state transition matrix over the entire interval $[t_0, t_f]$ is determined.

SUBROUTINE CENTER

PURPOSE: TO CONVERT THE POSITION AND VELOCITY VECTORS OF THE GRAVITATING BODIES FROM REFERENCE BODY ECLIPTIC TO BARYCENTRIC ECLIPTIC AND STORE THEM IN THE F ARRAY.

CALLING SEQUENCE: CALL CENTER

SUBROUTINES SUPPORTED: EPHEN

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS

BARYC  POSITION AND VELOCITY OF CENTER OF MASS RELATIVE TO EARTH. (AU, AU/DAY)

F  ARRAY OF PLANET EPHEMERIS DATA IN AU, AU/DAY UNITS. DATA INDICATED BY THE FOLLOWING INDICES:

4*I-2,J  VELOCITY OF I-TH PLANET RELATIVE TO THE SUN (INPUT) AND RELATIVE TO THE BARYCENTER (OUTPUT)

4*I-3,J  POSITION OF I-TH PLANET RELATIVE TO THE SUN (INPUT) AND RELATIVE TO THE BARYCENTER (OUTPUT)

4*IM-2,J  VELOCITY OF MOON RELATIVE TO EARTH (INPUT) AND RELATIVE TO BARYCENTER (OUTPUT)

4*IM-3,J  POSITION OF MOON RELATIVE TO EARTH (INPUT) AND RELATIVE TO BARYCENTER (OUTPUT)

GEOP  POSITION AND VELOCITY OF BODIES RELATIVE TO THE EARTH. (AU, AU/DAY)

IND  INDEX USED TO EXTRACT EARTH EPHEMERIS DATA RELATIVE TO SUN FROM F-ARRAY

II  INDEX OF IJ-TH GRAVITATIONAL BODY

SUM  SUM OF GRAVITATIONAL CONSTANTS (AU**3/DAY**2)

SUN  POSITION AND VELOCITY OF SUN RELATIVE TO EARTH (AU, AU/DAY)

COMMON COMPUTED/USED: F INITAL V

COMMON USED: NBODYI NO PMASS ZERO
CENTER Analysis

Let the state vector of position and velocity of the gravitating bodies (excluding the moon) in heliocentric ecliptic coordinates be denoted \( \rho_i, \omega_i \) at some reference time. Let the index of the earth be \( i_E \). Then the coordinates of all bodies (excluding the moon) relative to the earth is

\[
\begin{align*}
\vec{r}_i &= \vec{\rho}_i - \vec{\rho}_{iE} & i=1, n, i \neq M \\
\vec{v}_i &= \vec{\omega}_i - \vec{\omega}_{iE} & i=1, n, i \neq M
\end{align*}
\]

Let the position and velocity of the moon relative to the earth be denoted \( \vec{r}_M, \vec{v}_M \).

Define the radius vector to the center of mass (in earth ecliptic coordinates) by

\[
\vec{r}_{CM} = \frac{1}{M} \sum_{i=1}^{n} \mu_i \vec{r}_i \quad M = \sum_{i=1}^{n} \mu_i
\]

Its velocity relative to the earth may then be found by differentiation.

\[
\vec{v}_{CM} = \frac{1}{M} \sum_{i=1}^{n} \mu_i \vec{v}_i
\]

The coordinates of all gravitating bodies relative to the center of mass may then be computed

\[
\begin{align*}
\vec{r}_i &= \vec{r}_i - \vec{r}_{CM} \\
\vec{v}_i &= \vec{v}_i - \vec{v}_{CM}
\end{align*}
\]
SUBROUTINE CONC2

PURPOSE: COMPUTE STATE TRANSITION MATRIX USING ANALYTICAL
        PATCHED CONIO OR ANALYTICAL VIRTUAL MASS TECHNIQUES

CALLING SEQUENCE: CALL CONC2(R,V,DELT,GMX,PSIEC)

ARGUMENTS:
DELT  I  TIME INCREMENT OVER WHICH THE STATE
       TRANSITION MATRIX IS BEING COMPUTED
GMX   I  GRAVITATIONAL CONSTANT OF GOVERNING BODY
PSIEC O  STATE TRANSITION MATRIX
R     I  POSITION OF THE VEHICLE RELATIVE TO THE
       GOVERNING BODY
V     I  VELOCITY OF THE VEHICLE RELATIVE TO THE
       GOVERNING BODY

SUBROUTINES SUPPORTED: PSIM    CASCAD   PCTM

LOCAL SYMBOLS:
   A   SEMI-MAJOR AXIS
   A1  INTERMEDIATE VARIABLE
   A2  INTERMEDIATE VARIABLE
   A3  INTERMEDIATE VARIABLE
   AM2 INTERMEDIATE VARIABLE
   C1  MAGNITUDE OF RXV
   CSE COSINE OF ECCENTRIC ANOMALY
   GTA COSINE OF TRUE ANOMALY
   GTA2 COSINE OF TRUE ANOMALY ON ELLIPSE
   DDX0 INTERMEDIATE VARIABLE
   DDY0 INTERMEDIATE VARIABLE
   DX0  INTERMEDIATE VARIABLE
   DYO  INTERMEDIATE VARIABLE
   E   ECCENTRICITY
   EA  ECCENTRIC ANOMALY
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FM1</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
<tr>
<td>FM2</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
<tr>
<td>F</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>N</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>OPEC</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
<tr>
<td>ORB</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>P</td>
<td>SEMI-LATUS RECTUM</td>
</tr>
<tr>
<td>PI</td>
<td>MATHEMATICAL CONSTANT</td>
</tr>
<tr>
<td>PSIOP</td>
<td>INTERMEDIATE STATE TRANSITION MATRIX</td>
</tr>
<tr>
<td>PV</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
<tr>
<td>Q</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
<tr>
<td>RD</td>
<td>R DOT V DIVIDED BY MAGNITUDE OF R</td>
</tr>
<tr>
<td>RM</td>
<td>MAGNITUDE OF R</td>
</tr>
<tr>
<td>RRD</td>
<td>R DOT V</td>
</tr>
<tr>
<td>RTM0</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>R2</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>R3</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>SNE</td>
<td>SINE OF ECCENTRIC ANOMALY</td>
</tr>
<tr>
<td>SNF</td>
<td>SINE OF F</td>
</tr>
<tr>
<td>STA</td>
<td>SINE OF TRUE ANOMALY</td>
</tr>
<tr>
<td>STA2</td>
<td>SINE OF TRUE ANOMALY ON ELLIPSE</td>
</tr>
<tr>
<td>TIM1</td>
<td>INTERMEDIATE TIME</td>
</tr>
<tr>
<td>TIM2</td>
<td>INTERMEDIATE TIME</td>
</tr>
<tr>
<td>VM</td>
<td>MAGNITUDE OF V</td>
</tr>
<tr>
<td>WV</td>
<td>RXV</td>
</tr>
<tr>
<td>XO</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>YO</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>COMMON USED</td>
<td>EMB</td>
</tr>
<tr>
<td>-------------</td>
<td>-----</td>
</tr>
</tbody>
</table>

2. **Intermediate Vector**
COSC2 Analysis

COSC2 is responsible for the computation of a state transition matrix about a conic trajectory using the Danby matrization analytic formula.

Danby has shown (see Reference 2) that the state transition matrix (or matrization) has a particularly simple form if written in the orbital plane coordinate system. The state transition matrix \( \Phi \) defined by

\[
\delta x_f = \Phi(t_f, t_0) \delta x_o
\]

where \( \delta x_f, \delta x_o \) refer to perturbations about a conic trajectory at time \( t_f, t_0 \) respectively may be written in the orbital plane system

\[
\Phi(t_f, t_0) = M(t_f) M^{-1}(t_0)
\]

where \( M(t), M^{-1}(t) \) may be computed from the following formulae

\[
M = \begin{bmatrix}
\dot{X} & \dot{Y}X - h & 2X - 3Y & \dot{Y}X & \dot{X} \\
\dot{Y} & -X & 2Y - 3Y & -YX & \dot{X} \\
0 & 0 & Y & 0 & 0 \\
\dot{X} & YX & 0 & -X & \dot{Y} + YF \\
Y & -X^2 - X & 0 & -Y - 3Y & -X - YF \\
0 & 0 & Y & 0 & 0 \\
\end{bmatrix}
\]

\[
M^{-1} = APN^TJ^T
\]

where \( X, Y, \dot{X}, \dot{Y}, \ddot{X}, \ddot{Y} \) are evaluated at the time \( t \)

\( h \) is the angular momentum constant

\( \tau \) is the time interval from \( t \) to some epoch (periapsis)

and \( A = \text{diag} (a/\mu, a/\mu h, 1/h, a/\mu, a/\mu h, 1/h) \)

\[
J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}
\]

Thus to use the Danby formulation one must determine the transformation from the reference frame to the orbital plane coordinates, compute the values of the quantities \( X, Y, \dot{X}, \dot{Y}, \ddot{X}, \ddot{Y} \) and \( h \) and \( \tau \) at the times \( t_0, t_f \) and then use the above equations.

Let the initial state of the conic be denoted \( \bar{y}, \bar{v} \), the gravitational force \( \mu \), and the time interval \( \Delta t \). Then the unit vectors \( \bar{y} \) in the direction of periapsis, \( \bar{v} \) in the direction of the angular momentum vector, and \( \bar{q} = \bar{v} \times \bar{y} \) defining the orbital plane coordinate system may be computed by the following conic equations
\[ h = | \mathbf{r} \times \mathbf{v} | \]  
\[ \hat{W} = \frac{\mathbf{r} \times \mathbf{v}}{h} \]  
\[ \hat{r} = \frac{\mathbf{r} \cdot \mathbf{v}}{v} \]  
\[ p = \frac{h^2}{\mu} \]  
\[ a = \frac{-r}{2 - r v^2/\mu} \]  
\[ e = \sqrt{1 - \frac{r^2}{a}} \]  
\[ \cos f = \frac{r - a}{r} \quad \sin f = \frac{\hat{r} h}{\mu e} \]  
\[ \xi = \frac{\mathbf{r} \cdot \mathbf{v}}{h} \]  
\[ \eta = \cos f \xi - \sin f \hat{r} \]  
\[ \gamma = \sin f \xi + \cos f \hat{r} \]  
\[ \hat{e} = \frac{c}{r^2} \]  

The transformation matrix from the original \( \mathbf{r}, \mathbf{v} \) system to the orbital plane system may then be written

\[ T = \begin{bmatrix} \hat{P} & \hat{Q} & \hat{W} \end{bmatrix} \]  

Let the true anomaly at the pertinent time (\( t_o \) or \( t_f \)) be denoted \( f \). Then the quantities required in (3) are written

\[ X = r \cos f \quad Y = r \sin f \]  
\[ \dot{X} = \dot{r} \cos f - r \dot{\gamma} \sin f \quad \dot{Y} = \dot{r} \sin f + r \dot{\gamma} \cos f \]  
\[ \ddot{X} = -\frac{\mu X}{r^3} \quad \ddot{Y} = -\frac{\mu Y}{r^3} \]  

Having computed the state transition matrix \( \hat{\Phi} \) corresponding to the orbital plane system by equations (2), (3), (4), it is an easy task to convert it to the normal reference system

\[ \Phi = T \hat{\Phi} T^T \]

CONC2 Flow Chart

ENTER

Compute $\hat{\theta}, \hat{\phi}, \hat{w}$, and transformation matrix $T$

$\Delta 1$

$e = 0$

$\Delta 1$

Compute $T_1, T_f$ and $f_1$ (Kepler equation) for ellipse.

Compute $X, Y, X, Y, X, Y, M(t_1), M^{-1}(t_2)$, and $\phi$.

Compute $\phi = T \hat{\phi} T^T$.

RETURN
SUBROUTINE CONVRT

PURPOSE: TO COMPUTE THE GEOCENTRIC EQUATORIAL COORDINATES OF THE VEHICLE.

CALLING SEQUENCE: CALL CONVRT(R, PHI, THETA, VEL, GAMMA, SIGMA, X, Y, Z, VX, VY, VZ)

ARGUMENTS:
- GAMMA I PATH ANGLE
- R I GEOCENTRIC RADIUS
- PHI I DECLINATION
- THETA I RIGHT ASCENSION
- VEL I VELOCITY
- SIGMA I AZIMUTH

X 0 X COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES
Y 0 Y COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES
Z 0 Z COMPONENT OF POSITION IN GEOCENTRIC EQUATORIAL COORDINATES

VX 0 X COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES
VY 0 Y COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES
VZ 0 Z COMPONENT OF VELOCITY IN GEOCENTRIC EQUATORIAL COORDINATES

SUBROUTINES SUPPORTED: DATA DATAS

LOCAL SYMBOLS: B1 INTERMEDIATE VARIABLE
B2 INTERMEDIATE VARIABLE
B3 INTERMEDIATE VARIABLE
C6 COSINE OF PATH ANGLE
CP COSINE OF DECLINATION
CT COSINE OF RIGHT ASCENSION
SG SINE OF PATH ANGLE
SP  SINE OF DECLINATION
ST  SINE OF RIGHT ASCENSION
CONVET Analysis

Geocentric equatorial position and velocity components are related to geocentric radius, declination, right ascension, velocity magnitude, flight path angle, and azimuth through the following equations:

\[
x = r \cos \phi \cos \theta \\
y = r \cos \phi \sin \theta \\
z = r \sin \phi \\
\dot{x} = v (\sin \tau \cos \phi \cos \theta - \cos \tau \sin \sigma \sin \theta - \cos \tau \cos \sigma \sin \phi \cos \theta) \\
\dot{y} = v (\sin \tau \cos \phi \sin \theta + \cos \tau \sin \sigma \cos \theta - \cos \tau \cos \sigma \sin \phi \sin \theta) \\
\dot{z} = v (\sin \tau \sin \phi + \cos \tau \cos \sigma \cos \phi)
\]

The definitions of pertinent quantities are apparent in the following figure.
SUBROUTINE COPINS

PURPOSE: TO DETERMINE THE IMPULSIVE CORRECTION AND TIME REQUIRED TO INSERT FROM AN APPROACH HYPERBOLA INTO A COPLANAR ELIPTICAL ORBIT.

CALLING SEQUENCE: CALL COPINS(GM,R,V,DA,DE,DELW,TEX,DELY,IEX)

ARGUMENTS: GM I GRAVITATIONAL CONSTANT
R(3) I POSITION VECTOR AT DECISION
V(3) I VELOCITY VECTOR AT DECISION
DE I DESIRED SEMIMAJOR AXIS
DE I DESIRED ECCENTRICITY
DELW I DESIRED PERIAPSID SHIFT
TEX O TIME FROM DECISION TO EXECUTION (SECONDS)
DELY(3) O INSERTION VELOCITY CORRECTION
IEX O EXECUTION CODE
=0 EVENT IS EXECUTABLE
=1 NO EXECUTABLE SOLUTION FOUND

SUBROUTINES SUPPORTED: INSERS

SUBROUTINES REQUIRED: CAREL ELCAR

LOCAL SYMBOLS: AA COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A
AH HYPERBOLIC SEMIMAJOR AXIS
ARC THE CONSTANT 180
A1 CANDIDATE SOLUTION FOR SEMIMAJOR AXIS
A2 CANDIDATE SOLUTION FOR SEMIMAJOR AXIS
A TARGET SEMIMAJOR AXIS
BB COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A
B TANGENTIAL SOLUTION CONSTANT
CG COEFFICIENT DEFINING TANGENTIAL SOLUTION FOR A
COE $1/E$
COSTH $\cos(\text{THETA})$
COSW $\cos(w)$
C TANGENTIAL SOLUTION CONSTANT
DELVM MAGNITUDE OF FINAL CORRECTION
DISC DISCRIMINANT OF SOLUTION FOR THETA
DISK DISCRIMINANT OF TANGENTIAL SOLUTION FOR A
DRA DESIRED APOAPSIS RADIUS
DRP DESIRED PERIAPSIS RADIUS
DVM MAGNITUDE OF VELOCITY CORRECTION FOR CANDIDATE SOLUTION
DV VELOCITY CORRECTION OF CANDIDATE SOLUTION
D TANGENTIAL SOLUTION CONSTANT
EH HYPERBOLIC ECCENTRICITY
ERRMAX SCALAR ERROR ASSOCIATED WITH IMPOSSIBLE SOLUTION
ERR SCALAR ERRORS OF CANDIDATE SOLUTIONS
ER RADIUS ON ELLIPSE AT INSERTION
ETA TRUE ANOMALY ON ELLIPSE AT INSERTION
E ECCENTRICITY OF ELLIPSE
HI INCLINATION OF HYPERBOLA
HN ASCENDING NODE OF HYPERBOLA
HRP HYPERBOLIC PERIAPSIS RADIUS
HR RADIUS OF HYPERBOLA AT INSERTION
IOPT TYPE OF SOLUTION
ISOL INDEX OF SOLUTION
MIN    INDEX OF MINIMUM LOSS FUNCTION SOLUTION
NSOL3    NUMBER OF SOLUTIONS
PH    HYPERBOLIC SEMILATUS RECTUM
PI    THE MATHEMATICAL CONSTANT PI
PP    UNIT VECTOR TOWARD PERIAPSIS
P    ELLIPTICAL SEMILATUS RECTUM
QQ    UNIT VECTOR IN ORBIT PLANE NORMAL TO PP
RAD    DEGREE TO RADIAN TRANSFORMATION
RA    APOAPSIS RADIUS
RD    RADIUS TO DECISION STATE
REM3    MAGNITUDE OF RADIUS ON ELLIPSE AFTER INSERTION
RE    POSITION VECTOR ON ELLIPSE AFTER INSERTION
RH    POSITION ON HYPERBOLA BEFORE INSERTION
RMAG    MAGNITUDE OF RADIUS ON HYPERBOLA BEFORE INSERTION
RP    PERIAPSIS RADIUS
SGN    PARAMETER IN TANGENTIAL SOLUTION
SINW    SIN(W)
STA    TRUE ANOMALY ON HYPERBOLA AT DECISION
SYGN    POSITIVE OR NEGATIVE SIGN IN QUADRATIC
S    INTERMEDIATE VARIABLE
TFPE    TIME FROM PERIAPSIS ON ELLIPSE
TFPH    HYPERBOLIC TIME FROM PERIAPSIS AT INSERTION
THA    TRUE ANOMALY OF INSERTION ON HYPERBOLA
TIMDX    TIME FROM DECISION TO EXECUTION
TIMD    TIME FROM PERIAPSIS AT DECISION
\( V_D \) \text{ SPEED OF DECISION}

\( V_{EMG} \) \text{ SPEED ON ELLIPSE AFTER INSERTION}

\( V_E \) \text{ VELOCITY VECTOR ON ELLIPSE AFTER INSERTION}

\( V_H \) \text{ VELOCITY VECTOR ON HYPERBOLA BEFORE INSERTION}

\( V_{EMH} \) \text{ VELOCITY ON HYPERBOLA BEFORE INSERTION}

\( \theta \) \text{ ANGLE OF PERIAPSIS}

\( \eta \) \text{INIALLY PERIAPSIS}

\( \eta_H \) \text{INIALLY HYPERBOLIC PERIAPSIS}

\( \alpha \) \text{ ANGLE OF HYPERBOLIC PERIAPSIS}
COPINS Analysis:

COPINS determines the impulsive correction and time required to insert from an approach hyperbola into a coplanar elliptical orbit. The approach hyperbola is specified by a planetocentric state $\mathbf{r}, \mathbf{v}$ at a decision time $t_d$. The desired elliptical orbit is prescribed by input parameters $a$, $e$, $\Delta \omega$ where $a$ and $e$ are the semi-major axis and eccentricity of the desired ellipse and $\Delta \omega$ is the angle (measured counter clockwise) from the hyperbolic periapsis to the periapsis of the desired orbit. The situation is illustrated in Figure 1.

![Figure 1. Approach Hyperbola and Desired Orbit](image)

The planetocentric ecliptic state $\mathbf{r}, \mathbf{v}$ at the time of decision $t_d$ is first converted to Keplerian elements $(a_H, e_H, i_H, \Omega_H, \omega_H)$ via subroutine CAREL where $t_{Hd}$ is the time from periapsis (negative on the approach ray). The angle $f_{\infty}$ between the hyperbolic periapsis and the approach asymptote $\hat{S}$ is computed from

$$\cos f_{\infty} = \frac{1}{e} \quad 0 < f_{\infty} < 90^\circ$$

(1)

Thus the angle $\omega$ between the hyperbolic periapsis and the desired elliptical periapsis is given by

$$\omega = \Delta \omega$$

(2)

The hyperbola and ellipse may therefore be described in the PQ plane by standard conic formula, specifically,
where \( \theta \) is measured counter-clockwise from \( \vec{P} \) and \( p_H, p_E \) are the semi-latus rectum of the hyperbola and ellipse respectively. Obviously if an angle of intersection \( \theta^* \) is known, the states on both conics \((\vec{r}^*, \vec{v}^*\)) and \((\vec{r}^*, \vec{v}^*_E)\) may be computed from conic formulae and the desired impulsive correction is given by

\[
\Delta \vec{v} = \vec{v}^*_E - \vec{v}^*_H
\]  

(4) Likwise the time from periapsis to the intersection point \( t^* \) may be computed using hyperbolic formula and therefore the time from decision to execution is given by

\[
\Delta t = t^* - t_d
\]  

(5) Thus the coplanar insertion problem reduces to the determination of the optimal angle \( \theta^* \) for the impulsive maneuver.

From (3) the values of \( \theta \) for which \( r_H = r_E \) are given by

\[
\cos \theta = \frac{-xy + z \sqrt{D}}{y^2 + z^2}
\]  

(6) where

\[
\begin{align*}
x &= p_H - p_E \\
y &= p_H e_E \cos \omega - p_E e_H \\
z &= p_H e_E \sin \omega \\
D &= y^2 + z^2 - x^2
\end{align*}
\]  

(7) If the discriminant \( D \geq 0 \) there are at most two real non-extraneous solutions \( \theta_1, \theta_2 \) such that \( r_E(\theta) = r_H(\theta) \). Note that the angle \( \theta \) may not lie in the region inside the approach and departure asymptotes. If there are two solutions, both \( \Delta v \)'s are computed by (4) and the minimum \( \Delta v \) transfer is selected.
If $D < 0$, the applied hyperbola and the desired orbit do not intersect and there is no impulsive transfer between the two conics. In such a case the desired elements $a_e$ and $e_e$ are modified to determine the "best" tangential solution possible. Three different modifications are tested:

1. Vary $r_a$ while holding $r_p$ at the desired value.
2. Vary $r_p$ while holding $r_a$ at the desired value.
3. Vary $e_e$ while holding $a_e$ at the desired value.

The three modification schemes are illustrated in Figure 2 where the original nonintersecting orbit is shown by the broken lines.

![Diagram showing three different modifications](image)

**Figure 2. Candidate Orbit Modifications**

It is desired to modify the "a" and the "e" of the desired orbit to achieve the tangential configurations. From (6) it is obvious that a necessary condition for a tangential solution is given by $D = 0$. Using (7) $D$ may be written

$$D = p_H^2 (e_e^2 - 1) + p_e^2 b + 2p_H p_e - c p_e e_e$$

where $b = e_e^2 - 1$

$$c = 2p_H e_e \cos \omega$$

where it is observed the approach hyperbola is fixed and it is desired not to vary the $\omega$ of the desired ellipse so that subsequent apsidal rotations are avoided.
Modification Option 1: Rewriting (8a) in terms of \( a \) and \( r_p \) leads to

\[
a^2 D = (4 r_p^2 b + 4 r_p D_H - 2 r_p c) a^2 \\
+ (-2 D_H^2 - 4 r_p^3 b - 2 D_H c) a \\
+ (D_H^2 r_p^2 + r_p^4 b - c r_p^2)
\]  

(9)

Now if \( D \) is set equal to 0, \( r_p \) held at its desired value, and the resulting quadratic solved for \( a \), the solution will correspond to the tangential solution which holds \( r_p \) constant. If \( a \leq 0 \) or imaginary, the solution is disregarded. The modified eccentricity is of course defined by

\[
e = 1 - \frac{r_p}{a}
\]  

(10)

Modification Option 2: Rewriting (8a) in terms of \( a \) and \( r_a \) leads to

\[
a^2 D = (4 r_a^2 b + 4 r_a D_H + 2 r_a c) a^2 \\
+ (-2 D_H^2 - 4 r_a^3 b - 2 D_H c) a \\
+ (D_H^2 r_a^2 + r_a^4 b + c r_a^2)
\]  

(11)

For computational purposes the similarity between (9) and (11) may be exploited. Again setting \( D = 0 \) and holding \( r_a \) at its desired value, the value of \( a \) may be determined which specifies the tangential solution holding \( r_a \) constant. Having determined a realistic value of \( a \), the corresponding eccentricity is given by

\[
e = \frac{r_a}{a} - 1
\]  

(12)

Modification Option 3: Rewriting (8a) in terms of \( a \) and \( e_x \) leads to

\[
D = (d^2 b) a^2 + (2 p_d - c d e_x) a - d p_H^2 \\
d = (1 - e_x^2)
\]  

(13)

Setting \( D = 0 \) and solving for \( a \) while holding \( e_x \) at its desired value then defines the option 3 solution.
To determine the "best" modified orbit from the three candidate options a rather arbitrary scheme is used. A scalar error is assigned to each option according to a weighting factor and the difference between the desired and achieved values of the periapsis and apoapsis radii:

\[ E_i = W_i \left( |\Delta r_a| + |\Delta r_p| \right) \]

With the scalar factor \( W_i \) is set to 1, 2, 3 respectively for the three enhance. The preferred strategy is the one which requires a correction only at apoapsis while the least desired scheme requires subsequent corrections at periapsis and apoapsis.

The refined orbital elements that necessarily lead to a tangential form (5) may now be used to compute the angle of intersection \( \theta \).
COPINS Flow Chart

ENTER

Set Nominal values of flags: LEX-IOPT=0, NSOLS=1

Compute elements of hyperbola at time of decision (CAREL) and record.

Compute coefficients X,Y,Z of quadratic equation defining cos θ (eqns 6,7) and discriminant DISC

DISC ≥ 0

≥ 0 → 250

< 0

Prepare for tangential solution modifications by setting IOPT = 1, NSOLS = 3, and compute constants B,C,D for tangential solutions (eqn 8).

105

ISOL = ISOL + 1

= 1

= 3,4

= 2

 Modify r_p solution. Set s = r_p, i = -1

Modify r_a solution. Set s = r_a, i = +1

ISOL = ?

= 4

500

Modify "a" solution. Compute coefficients AA, BB, CC of quadratic in "a" defining tangential solution.

Compute discriminant DISK of quadratic defining "a" of tangential solution.
Determine two candidate solutions \( a_1, a_2 \).

\[ \text{DISK} = ? \]

- If \( \text{DISK} \geq 0 \), set \( \epsilon_{\text{ISOL}} = 10^{-25} \).
  - \( a_1 - ? \)
    - If \( a_1 \leq 0 \), determine \( e \) corresponding to \( a_2 \).
    - If \( a_1 > 0 \), determine \( e \) corresponding to \( a_1 \).
      - If \( e < 0 \), set \( \epsilon_{\text{ISOL}} = 10^{-15} \).
      - If \( e > 1 \), set \( \epsilon_{\text{ISOL}} = 10^{-25} \).

Compute \( X, Y, Z, \text{DISC} \) for tangential solution.

Solve quadratic for \( \cos \theta \).

- If \( |\cos \theta| > 1 \), set \( \text{NSOLS} = \text{NSOLS} \).
- If \( |\cos \theta| < 1 \), determine principle value of \( \theta \): \( 0 < \theta < \pi \).
  - If \( \text{IOPT} = 0 \), set \( \epsilon_{\text{ISOL}} = 10^{-25} \).
    - If \( \text{IOPT} \neq 0 \), set \( \epsilon_{\text{ISOL}} = 10^{-15} \).
Determine radius at intersection on hyperbola $r_h$ and ellipse $r_e$.

\[ |r_h - r_e| = ? \geq 1 \rightarrow \begin{array}{c}
\theta = ? \rightarrow < 0 \rightarrow B \\
\theta = 0 \rightarrow C
\end{array}

Compute state $\bar{r}_h, \bar{v}_n$ and time $t_n$ on hyperbola and state on ellipse $\bar{r}_e, \bar{v}_e$ at intersection.

Compute insertion velocity and time
\[ \Delta V_{ISOL} = \bar{v}_e - \bar{v}_n \]
\[ \Delta t_{ISOL} = t_h - t_D \]

\[ \text{IOPT = ?} \rightarrow 105 \]

\[ \text{ISOL = } |\Delta V_{ISOL}|, \Delta t > 0 \]
\[ = 10^{-25}, \Delta t < 0 \]

\[ \text{ISOL = ?} \rightarrow 250 \]

\[ \text{500} \rightarrow -2 \]

Choose index MJ of minimum
Set $\Delta V = -V_{MIN}$
$\Delta t = \Delta t_{MIN}$

If $e_{MIN} = 5 \times 10^{-25}$ or $\Delta t < 0$
Set $\text{IXX} = 1$

RETURN
SUBROUTINE CORREL

PURPOSE: CONVERT COVARIANCE MATRIX PARTITIONS TO CORRELATION MATRIX PARTITIONS AND STANDARD DEVIATIONS AND WRITE THEM OUT

CALLING SEQUENCE: CALL CORREL(PP,CXXSP,PSP,CXUP,UO,CXVP,V0,CXSUP, CXSVP)

ARGUMENTS:
- PP  I  POSITION/VELOCITY COVARIANCE MATRIX
- CXXSP I  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND POSITION/VELOCITY STATE
- PSP  I  SOLVE-FOR PARAMETER COVARIANCE MATRIX
- CXUP  I  CORRELATION BETWEEN POSITION/VELOCITY STATE AND DYNAMIC CONSIDER PARAMETERS
- UO  I  DYNAMIC CONSIDER PARAMETER COVARIANCE MATRIX
- CXVP  I  CORRELATION BETWEEN POSITION/VELOCITY STATE AND MEASUREMENT CONSIDER PARAMETERS
- VO  I  MEASUREMENT CONSIDER PARAMETER COVARIANCE MATRIX
- CXSUP I  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND DYNAMIC CONSIDER PARAMETERS
- CXSVP I  CORRELATION BETWEEN SOLVE-FOR PARAMETERS AND MEASUREMENT CONSIDER PARAMETERS

SUBROUTINES SUPPORTED:
- PRINT4  SETEVS  GUISIM  PRESIM  PRNTS4
- PRINT3  SETEVN  GUIDM  PRED  PRNTS3

LOCAL SYMBOLS:
- DUM  INVERSE OF SQUARE ROOT OF DIAGONAL ELEMENTS IN DYNAMIC AND MEASUREMENT CONSIDER COVARIANCE PARTITIONS
- IEND  COUNTER INDICATING TOTAL NUMBER OF AUGMENTED STATE VARIABLES
- ROW  INTERMEDIATE COMPUTATION AND OUTPUT VECTOR
- SQP  INVERSE OF THE SQUARE ROOT OF DIAGONAL ELEMENTS IN VEHICLE AND SOLVE-FOR COVARIANCE PARTITIONS
- ZZ  STANDARD DEVIATION

COMMON USED:
- KPRINT  NDIM1  NDIM2  NDIM3  ONE
SUBROUTINE DATA

PURPOSE: TO READ INPUT DATA, TRANSLATE THIS DATA INTO PROPER
INTERNAL VALUES, ASSIGN VALUES TO UNSPECIFIED NAMELIST
VARIABLES, SET NECESSARY INITIAL VALUES, COMPUTE
DIMENSIONS OF STATE TRANSITION, OBSERVATION, AND
COVARIANCE MATRIX PARTITIONS, ORDER MEASUREMENT AND EVENT
SCHEDULES, AND PRINT OUT INITIAL CONDITIONS IN THE ERRAN
PROGRAM.

CALLING SEQUENCE: CALL DATA

ARGUMENTS: NONE

SUBROUTINES SUPPORTED: ERRON

SUBROUTINES REQUIRED: CONVRT EPHEM GHA ORB PECEQ
TIME TRANS XYZRVT

LOCAL SYMBOLS: AI INCLINATION
A0 MIN INTERMEDIATE VARIABLE
ANODE LONGITUDE OF ASCENDING NODE
A SEMIMAJOR AXIS
DUM1 INTERMEDIATE STORAGE ARRAY
DUM INTERMEDIATE STORAGE ARRAY
D INTERMEDIATE JULIAN DATE
DATE ARRAY CONTAINING FINAL JULIAN DATE
EARTH CALENDAR DATE AT WHICH EARTHS ORBITAL
ELEMENTS WILL BE CALCULATED
E ECCENTRICITY
FNDT DATE OF FINAL TIME
GAMMA PATH ANGLE
GH GRAVITATIONAL CONSTANT OF CENTRAL BODY
ICNT COUNTER
IDAY CALENDAR DAY OF FINAL TIME

163
IHR  CALENDAR HOUR OF FINAL TIME
IMIN  CALENDAR MINUTES OF FINAL TIME
IMO   CALENDAR MONTH OF FINAL TIME
IYR   CALENDAR YEAR OF FINAL TIME
JUPITER  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF JUPITER WILL BE CALCULATED
LDAy  CALENDAR DAY OF INITIAL TIME
LHR   CALENDAR HOURS OF INITIAL TIME
LMIN  CALENDAR MINUTES OF INITIAL TIME
LMO   CALENDAR MONTH OF INITIAL TIME
LYR   CALENDAR YEAR OF INITIAL TIME
MARS  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MARS WILL BE CALCULATED
MERCURY  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF MERCURY WILL BE CALCULATED
MOON  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF EARTHS MOON WILL BE CALCULATED
MENT  NUMBER OF ENTRIES IN MEASUREMENT SCHEDULE
NEPTUNE  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF NEPTUNE WILL BE CALCULATED
OME   ARGUMENT OF PERIAPSIS
PHIT  DECLINATION
PLUTO  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF PLUTO WILL BE CALCULATED
PRD   INITIAL SEMI-LATUS RECTUM OF SPACECRAFT ORBIT
RDS   GEOCENTRIC RADIUS OF VEHICLE
RD    MAGNITUDE OF POSITION VECTOR
SATURN  CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF SATURN WILL BE CALCULATED
SEC   INTERMEDIATE CALENDAR SECONDS
THETA  URANUS  VfL  VENUS

TRUE  AHOHRY  OF  XNSTAWTANEWS  POSXTXON  AND  VELOCXTY

RI  WT  ASCENSION  CALEND4R

DATE  AT  WHICH  ORBITAL  ELEMENTS  OF  URANUS  WILL  BE  CALCULATED

VEL  INJECTION  VELOCITY  RELATIVE  TO  EARTH

VENUS  CALENDAR  DATE  AT  WHICH  ORBITAL  ELEMENTS  OF  VENUS  WILL  BE  CALCULATED

VL  MAGNITUDE  OF  VELOCITY  VECTOR

VUNIT  INTERMEDIATE  VELOCITY  CONVERSION  FACTOR

COMMON  COMPUTED/USED:

ACCD  ACC  ALNGTH  CXSU  CXSV
CXU  CXV  CXXS  DELAXS  DELECC
DELICL  DELMA  DELMUP  DELMUS  DELNOD
DELP  DELW  DTMAX  DTPLAN  DTSUN
EM1  EM4  EM5  EM6  EM7
EM8  EPS  EP50  FACP  FACV
FNTM  FOP  FOV  IAUGIN  IBARY
ICOQ3  ICDT3  ICOORD  ICOOR  IDNF
IEIG  IE,  HEM  IEVT  IHYP1  IMNF
INPR  IOPT7  IPRINT  IPRT  ISP2
ISTMC  ISTM1  KPRTINT  MNCN  NBOD
NOAGC  NOIM1  NOIM2  NOIM3  NEV10
NEV1  NEV1  NEV2  NEV3  NEV4
NEV5  NEV6  NEV7  NEV9  NEV
NNN  NO  NISTC  NMC  ONE
PS  P  RAD  SAL  SIGALP
SIGBET  SIGPRO  SIGRES  SLAT  SLON
SSS  TEV  TMN  TM  TRTM1
TNO  T  UST  UD  VST
V0  WST  XI  XI  ZERO

COMMON  COMPUTED:

BDTSI1  BCRS12  BORS13  BDTSI1  BDTSI2
BDTSI3  BSI1  BSI2  BSI3  CXSUB
CXSUB  CXSUG  CXSVB  CXSVG  CXUB  CXUG
CXVB  CXVG  CXXSB  CXXSG  DELTM
EM13  EM2  EM3  EM9  EM
HALF  IAUGMC  IAUGMC  IAUG  ICA1
ICA2  ICA3  ICL2  ICL  IIPOL
COMMON USED:

<table>
<thead>
<tr>
<th>AINC7</th>
<th>ANODE7</th>
<th>DATEJ</th>
<th>DNCH</th>
<th>ECCC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMNT</td>
<td>EVNM</td>
<td>F</td>
<td>G</td>
<td>HPC7</td>
</tr>
<tr>
<td>IPROB</td>
<td>MNNAME</td>
<td>NB</td>
<td>NLP</td>
<td>NTP</td>
</tr>
<tr>
<td>PERP7</td>
<td>PI</td>
<td>PLANET</td>
<td>PMASS</td>
<td>P7</td>
</tr>
<tr>
<td>TAU7</td>
<td>TPT2</td>
<td>XLAB</td>
<td>XMM</td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM DATAS

PURPOSE: TO READ INPUT DATA, TRANSLATE THIS DATA INTO PROPER INTERNAL VALUES, ASSIGN VALUES TO UNSPECIFIED NAMELIST VARIABLES, SET NECESSARY INITIAL VALUES, COMPUTE DIMENSIONS OF STATE TRANSITION, OBSERVATION, AND COVARIANCE MATRIX PARTITIONS, ORDER MEASUREMENT AND EVENT SCHEDULES, AND PRINT OUT INITIAL CONDITIONS IN THE SIMUL PROGRAM.

CALLING SEQUENCE: CALL DATAS

SUBROUTINES SUPPORTED: MAIN

SUBROUTINES REQUIRED: CONVRT DATA1S ELCAR EPHEM ORB PESEQ TIME TRANS

LOCAL SYMBOLS

AI INITIAL INCLINATION OF SPACECRAFT ORBIT
AMODE INITIAL LONGITUDE OF ASCENDING NODE OF SPACECRAFT ORBIT
A INITIAL SEMI-MAJOR AXIS OF SPACECRAFT ORBIT
DATE ARRAY CONTAINING FINAL JULIAN DATE
DUM1 PLANETO-CENTIC ECLIPTIC SPACECRAFT STATE
DUM COORDINATE TRANSFORMATION FROM PLANETO-CENTRIC EQUATORIAL TO PLANETO-CENTRIC ECLIPTIC COORDINATES
D JULIAN DATE AT LAUNCH
EARTH CALENDAR DATE AT WHICH ORBITAL ELEMENTS OF EARTH WILL BE CALCULATED
E INITIAL ECCENTRICITY OF SPACECRAFT ORBIT
FNDT FINAL JULIAN DATE
GAMMA INJECTION PATH ANGLE
GM GRAVITATIONAL CONSTANT OF TARGET PLANET
IDAY DAY OF FINAL COMPUTATION
IMHR HOUR OF FINAL COMPUTATION
MIN MINUTE OF FINAL COMPUTATION
IMO MONTH OF FINAL COMPUTATION
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IYR</td>
<td>Year of Final Computation</td>
</tr>
<tr>
<td>JUP</td>
<td>Jupiter Calendar Date at which Orbital Elements of Jupiter Will Be Calculated</td>
</tr>
<tr>
<td>LDAY</td>
<td>Launch Day</td>
</tr>
<tr>
<td>LHR</td>
<td>Launch Hour</td>
</tr>
<tr>
<td>LMIN</td>
<td>Launch Minute</td>
</tr>
<tr>
<td>LMO</td>
<td>Launch Month</td>
</tr>
<tr>
<td>Lyr</td>
<td>Launch Year</td>
</tr>
<tr>
<td>MARS</td>
<td>Mars Calendar Date at which Orbital Elements of Mars Will Be Calculated</td>
</tr>
<tr>
<td>MERCURY</td>
<td>Mercury Calendar Date at which Orbital Elements of Mercury Will Be Calculated</td>
</tr>
<tr>
<td>MOON</td>
<td>Moon Calendar Date at which Orbital Elements of Moon Will Be Calculated</td>
</tr>
<tr>
<td>NENT</td>
<td>Number of Entries in Measurement Schedule</td>
</tr>
<tr>
<td>NEPTUNE</td>
<td>Neptune Calendar Date at which Orbital Elements of Neptune Will Be Calculated</td>
</tr>
<tr>
<td>OME</td>
<td>Initial Argument of Periapsis of Spacecraft Orbit</td>
</tr>
<tr>
<td>PHIT</td>
<td>Injection Declination</td>
</tr>
<tr>
<td>PLUTO</td>
<td>Pluto Calendar Date at which Orbital Elements of Pluto Will Be Calculated</td>
</tr>
<tr>
<td>PRD</td>
<td>Initial Semi-Latus Rectum of Spacecraft Orbit</td>
</tr>
<tr>
<td>RDS</td>
<td>Earth-Centered Injection Radius</td>
</tr>
<tr>
<td>RD</td>
<td>Dummy Variable</td>
</tr>
<tr>
<td>SATURN</td>
<td>Saturn Calendar Date at which Orbital Elements of Saturn Will Be Calculated</td>
</tr>
<tr>
<td>SECl</td>
<td>Second of Final Computation</td>
</tr>
<tr>
<td>SECl</td>
<td>Launch Second</td>
</tr>
<tr>
<td>SEC</td>
<td>Second of Calendar Date at which Orbital</td>
</tr>
</tbody>
</table>
**Elements of a planet will be calculated**

<table>
<thead>
<tr>
<th><strong>Sigma</strong></th>
<th>Injection Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ta</strong></td>
<td>Initial Spacecraft True Anomaly</td>
</tr>
<tr>
<td><strong>Theta</strong></td>
<td>Injection Right Ascension</td>
</tr>
<tr>
<td><strong>Uranus</strong></td>
<td>Calendar Date at which Orbital Elements of Uranus Will be Calculated</td>
</tr>
<tr>
<td><strong>Vel</strong></td>
<td>Injection Velocity Relative to Earth</td>
</tr>
<tr>
<td><strong>Venus</strong></td>
<td>Calendar Date at Which Orbital Elements of Venus Will be Calculated</td>
</tr>
<tr>
<td><strong>Vl</strong></td>
<td>Dummy Variable</td>
</tr>
<tr>
<td><strong>Vunit</strong></td>
<td>Velocity Conversion Factor</td>
</tr>
</tbody>
</table>

---

**Common Computed/Used:**

<table>
<thead>
<tr>
<th>AALP</th>
<th>AGEN</th>
<th>ACCNO</th>
<th>ACC1</th>
<th>ACC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADEVX</td>
<td>ALMGTH</td>
<td>APRO</td>
<td>ARES</td>
<td>BIA</td>
</tr>
<tr>
<td>CXSU</td>
<td>CXSV</td>
<td>CXU</td>
<td>CXV</td>
<td>CXXS</td>
</tr>
<tr>
<td>DAB</td>
<td>DEB</td>
<td>DELAXS</td>
<td>DELECC</td>
<td>DELICL</td>
</tr>
<tr>
<td>DELMA</td>
<td>DELMUP</td>
<td>DELMUS</td>
<td>DELNOD</td>
<td>DELTP</td>
</tr>
<tr>
<td>DELW</td>
<td>DIB</td>
<td>DMAB</td>
<td>DMUPB</td>
<td>DMUSB</td>
</tr>
<tr>
<td>DNO8</td>
<td>DTMAX</td>
<td>DTPLAN</td>
<td>DTSUN</td>
<td>DBW</td>
</tr>
<tr>
<td>EM1</td>
<td>EM4</td>
<td>EM5</td>
<td>EM6</td>
<td>EM7</td>
</tr>
<tr>
<td>EM8</td>
<td>EPS</td>
<td>EP50</td>
<td>FACP</td>
<td>FACV</td>
</tr>
<tr>
<td>FNTM</td>
<td>FOP</td>
<td>FOV</td>
<td>IAMNF</td>
<td>IAUGIN</td>
</tr>
<tr>
<td>I3ARY</td>
<td>ICDT3</td>
<td>ICOORD</td>
<td>ICOOR</td>
<td>IDNF</td>
</tr>
<tr>
<td>IEIG</td>
<td>IEPEM</td>
<td>IHYP1</td>
<td>IMNF</td>
<td>INPR</td>
</tr>
<tr>
<td>IOP7</td>
<td>IPRINT</td>
<td>IPRT</td>
<td>ISP2</td>
<td>ISTMC</td>
</tr>
<tr>
<td>ISTM1</td>
<td>KPRINT</td>
<td>MNCN</td>
<td>NBOD1</td>
<td>NBOD</td>
</tr>
<tr>
<td>NB1</td>
<td>NDACC</td>
<td>NDIM1</td>
<td>NDIM2</td>
<td>NDIM3</td>
</tr>
<tr>
<td>NEV10</td>
<td>NEV11</td>
<td>NEV1</td>
<td>NEV2</td>
<td>NEV3</td>
</tr>
<tr>
<td>NEV4</td>
<td>NEV5</td>
<td>NEV6</td>
<td>NEV7</td>
<td>NEV9</td>
</tr>
<tr>
<td>KAT</td>
<td>NST</td>
<td>NTMC</td>
<td>ONE</td>
<td>PS</td>
</tr>
<tr>
<td>P</td>
<td>RAQ</td>
<td>SAL</td>
<td>SIGALP</td>
<td>SIGBET</td>
</tr>
<tr>
<td>SIGPRO</td>
<td>SIGRES</td>
<td>SLAT</td>
<td>SLB</td>
<td>SLOP’</td>
</tr>
<tr>
<td>SSS</td>
<td>TM</td>
<td>TRTM1</td>
<td>TIM1</td>
<td>TIM2</td>
</tr>
<tr>
<td>TWC</td>
<td>T</td>
<td>UNMAC</td>
<td>UST</td>
<td>UO</td>
</tr>
<tr>
<td>VST</td>
<td>V0</td>
<td>WST</td>
<td>XI</td>
<td>XP</td>
</tr>
</tbody>
</table>

**Common Computed:**

<table>
<thead>
<tr>
<th>ADEVXS</th>
<th>BORS11</th>
<th>BORS2</th>
<th>BORS3</th>
<th>BORS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOTS12</td>
<td>BOTS3</td>
<td>BS1</td>
<td>BS2</td>
<td>BS3</td>
</tr>
<tr>
<td>DELTM</td>
<td>EM13</td>
<td>EM</td>
<td>EM3</td>
<td>E MI</td>
</tr>
<tr>
<td>HALF</td>
<td>IAUGDC</td>
<td>I AUGC</td>
<td>IAUG</td>
<td>JCA1</td>
</tr>
<tr>
<td>ICA1</td>
<td>ICA3</td>
<td>ICGW</td>
<td>IGL2</td>
<td>IC</td>
</tr>
<tr>
<td>INCMT</td>
<td>INITIAL</td>
<td>ISO11</td>
<td>ISO12</td>
<td>ISO13</td>
</tr>
<tr>
<td>ISPH</td>
<td>ITR</td>
<td>NBO0YI</td>
<td>NEV8</td>
<td>RCA1</td>
</tr>
<tr>
<td>RCA2</td>
<td>RCA3</td>
<td>RS011</td>
<td>RS012</td>
<td>RS013</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>XSL</td>
<td>XU</td>
<td>XV</td>
<td>Z1</td>
<td></td>
</tr>
</tbody>
</table>

**COMMON_USED:**

<table>
<thead>
<tr>
<th>AINC7</th>
<th>ANODE7</th>
<th>AVARM</th>
<th>DATEJ</th>
<th>DNCN</th>
<th>ECC7</th>
<th>ELMNT</th>
<th>F</th>
<th>HP7</th>
<th>IPROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAF6</td>
<td>N8</td>
<td>NLP</td>
<td>NTP</td>
<td>PERP7</td>
<td>PI</td>
<td>PLANET</td>
<td>PMASS</td>
<td>P7</td>
<td>TAU7</td>
</tr>
<tr>
<td>TPT2</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
<td>T6</td>
<td>T7</td>
<td>XNM</td>
<td></td>
</tr>
</tbody>
</table>

**DATAS-D**

170
SUBROUTINE DATA1

PURPOSE: TO CONTINUE THE INITIALIZATION PROCESS DESCRIBED UNDER DATA.

CALLING SEQUENCE: CALL DATA1(NENT)

ARGUMENTS: NENT I NUMBER OF CARDS IN THE MEASUREMENT SCHEDULE

SUBROUTINES SUPPORTED: DATA

SUBROUTINES REQUIRED: GHA

LOCAL SYMBOLS: AMIN INTERMEDIATE VARIABLE
AP INTERMEDIATE TIME ARRAY
ICNT COUNTER ON MEASUREMENT SCHEDULE CARDS
IROW INTERMEDIATE ROW INDEX
MEAS MEASUREMENT CODES
NOUT DIMENSION OF AUGMENTED COVARIANCE MATRIX
SCHED ARRAY OF TIMES IN MEASUREMENT SCHEDULE

COMMON COMPUTED/USED: IEVNT NEV MMH SLAT SLON
TEV TMN T1 T2 T3
T4 T5 T6 T7 T

COMMON COMPUTED: CXSUB CXSUG CXSVB CXSVG CXUB
CXUG CXVB CXVG CXXSB CXXSG
EM EPS IIPOL IPOL MCNTR
MCODE NAE NGE NPE NQE
OMEGA P8 PG PSB PSG
TG XB XF XG

COMMON USED: CXSU CXSV CXU CXV CXXS
DATEJ DNCN EM7 EM8 EPO
EVNN FACP FAGV FNTM ICDQ3
IONF IGUID IMYP1 IMNF ISTH4
MNCN MNNNAME NDMIN1 NDMIN2 NDMIN3
NEV1 NEV2 NEV3 NEV4 NEV5
NEV6 NEV7 MST ONE PS
P RAD SAL TPT2 TRTM1
VO V0 XI ZERO
### SUBROUTINE DATA1S

**PURPOSE:** TO CONTINUE THE INITIALIZATION PROCESS DESCRIBED UNDER DATA1S.

**CALLING SEQUENCE:** CALL DATA1S(NENT)

**ARGUMENT:**
- **NENT** I NUMBER OF CARDS IN THE MEASUREMENT SCHEDULE

**SUBROUTINES SUPPORTED:** DATA1S

**SUBROUTINES REQUIRED:** GHA

**LOCAL SYMBOLS:**
- **AMIN** INTERMEDIATE VARIABLE
- **AP** INTERMEDIATE TIME ARRAY
- **ICNT** COUNTER ON MEASUREMENT SCHEDULE CARDS
- **IRON** INTERMEDIATE ROW INDEX
- **KEAS** MEASUREMENT CODES
- **NOUT** DIMENSION OF AUGMENTED COVARIANCE MATRIX
- **PARAM** ARRAY OF AUGMENTED BIASES
- **SCHED** ARRAY OF TIMES IN MEASUREMENT SCHEDULE

#### COMMON COMPUTED/USED:

<table>
<thead>
<tr>
<th>ADEVXS</th>
<th>IEVNT</th>
<th>NEV</th>
<th>MMN</th>
<th>SLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLB</td>
<td>SLON</td>
<td>TEV</td>
<td>TMN</td>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
<td>T6</td>
</tr>
<tr>
<td>T7</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADEVSB</th>
<th>ADEVXB</th>
<th>CXSUB</th>
<th>CXSUG</th>
<th>CXSVB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXSVG</td>
<td>CXUB</td>
<td>CXUG</td>
<td>CXVB</td>
<td>CVVG</td>
</tr>
<tr>
<td>CXSBB</td>
<td>CXSSG</td>
<td>EDEVXS</td>
<td>EDEVX</td>
<td>EM</td>
</tr>
<tr>
<td>EP5</td>
<td>IPOL</td>
<td>IPOL</td>
<td>MCNTR</td>
<td>MCODE</td>
</tr>
<tr>
<td>NAE</td>
<td>NGE</td>
<td>NPE</td>
<td>NQE</td>
<td>OMEGA</td>
</tr>
<tr>
<td>PB</td>
<td>PG</td>
<td>PSB</td>
<td>PSG</td>
<td>TG</td>
</tr>
<tr>
<td>XB</td>
<td>XF</td>
<td>XG</td>
<td>XI1</td>
<td>Z1</td>
</tr>
</tbody>
</table>

#### COMMON COMPUTED:

<table>
<thead>
<tr>
<th>ACC1</th>
<th>ADEVX</th>
<th>AAVRY</th>
<th>BIA</th>
<th>CXSU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXSV</td>
<td>CXU</td>
<td>CXV</td>
<td>CXXS</td>
<td>DAB</td>
</tr>
<tr>
<td>DATEJ</td>
<td>DEB</td>
<td>DIB</td>
<td>DMAB</td>
<td>DMUPB</td>
</tr>
<tr>
<td>DMUSB</td>
<td>DMCN</td>
<td>DNOB</td>
<td>DNB</td>
<td>EM7</td>
</tr>
<tr>
<td>EM8</td>
<td>EP50</td>
<td>EVNM</td>
<td>FACP</td>
<td>FACV</td>
</tr>
<tr>
<td>FNTM</td>
<td>IAMNF</td>
<td>IAUIN</td>
<td>ICOQ3</td>
<td>ICDT3</td>
</tr>
<tr>
<td>IDNF</td>
<td>IMNF</td>
<td>ISTMC</td>
<td>MNCN</td>
<td></td>
</tr>
<tr>
<td>MMNAME</td>
<td>NB0D1</td>
<td>NB1</td>
<td>NDIM1</td>
<td>NDIM2</td>
</tr>
<tr>
<td>NDIM3</td>
<td>NEV1</td>
<td>NEV2</td>
<td>NEV3</td>
<td>NEV4</td>
</tr>
<tr>
<td>NEV5</td>
<td>NEV6</td>
<td>NEV7</td>
<td>NST</td>
<td>ONE</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>PLANET</td>
<td>PS</td>
<td>P</td>
<td>RAD</td>
<td>SAL</td>
</tr>
<tr>
<td>TPT2</td>
<td>TPTM1</td>
<td>TTM1</td>
<td>TTM2</td>
<td>UNMAG</td>
</tr>
<tr>
<td>U0</td>
<td>V0</td>
<td>XDOM</td>
<td>XI</td>
<td>ZERO</td>
</tr>
</tbody>
</table>
SUBROUTINE DESENT

PURPOSE: TO COMPUTE A CORRECTION TO AN INITIAL VELOCITY BY THE STEEPEST DESCENT OR CONJUGATE GRADIENT TECHNIQUES FOR USE BY TARGET.

CALLING SEQUENCE: CALL DESENT(ERC, IT, KREK, GMP, PP)

ARGUMENTS:

ERC  I  SCALAR ERROR OF CURRENT ITERATE
IT   I/O  ITERATION COUNTER
KREK I  STEEPEST DESCENT RECTIFICATION NUMBER
GMP  I/O  PREVIOUS GRADIENT MAGNITUDE (INPUT) CURRENT GRADIENT MAGNITUDE (OUTPUT)
PP(3) I/O  PREVIOUS GRADIENT (INPUT) CURRENT GRADIENT (OUTPUT)

SUBROUTINES SUPPORTED: TARGET
SUBROUTINES REQUIRED: TARPIT VHP

LOCAL SYMBOLS:

ACK  CURRENT ACCURACY LEVEL
AER  ABSOLUTE ERRORS OF TARGET PARAMETERS
AUXM VALUES OF AUXILIARY PARAMETERS OF CURRENT ITERATE
DD   DIRECTIONAL DERIVATIVE
DELVM MAGNITUDE OF PREDICTED CORRECTION
DEVI DEVIATION OF NOMINALLY-PREDICTED AUXILIARY PARAMETERS FROM CURRENT ITERATE VALUES
DUMMY DUMMY VARIABLES
DUMM DUMMY VARIABLES
DVEE VELOCITY PERTURBATIONS
DVHM MAXIMUM ALLOWABLE VELOCITY INCREMENT
ERB  SCALAR ERROR OF NOMINALLY-PREDICTED STEP
GC   CURRENT GRADIENT
GMC  MAGNITUDE OF GC
MB   NOMINALLY PREDICTED STEP MAGNITUDE
HH  CORRECTION MAGNITUDE AFTER CONSTRAINTS
HS  CORRECTION MAGNITUDE AFTER PARABOLIC FIT
IEND  FLAG SET TO 1 IF TOLERANCES ACCEPTABLE
       ON PERTURBED TRAJECTORY
ISP2  SOI STOPPING CONDITION FLAG
       =0 DO NOT STOP AT SOI
       =1 STOP AT SOI
PC  DIRECTION OF CORRECTION
PERR  PERTURBED ERRORS
PM  MAGNITUDE OF UNNORMALIZED DIRECTION VECTOR
QC  UNIT VECTOR IN DIRECTION OF GRADIENT
RSF  FINAL STATE OF INTEGRATION

COMMON COMPUTED/USED:
DELTA V ISPH RIN TEN

COMMON COMPUTED:
ICL2 ICL INGRT RRF

COMMON USED:
AAUX AC ATAR CTGL DAUX
DELTAT DTAR DVMAX D1 FAC
IPHASE ISTOP KUR LEV LVLS
NOPAR PERV TRIM TWO ZERO
DESENT Analysis

DESENT computes a correction to an initial velocity by the steepest descent or conjugate gradient techniques for use by TARGET.

The technique used is determined by the value of METHOD. DESENT takes n steps in the conjugate gradient directions before rectifying by making a steepest descent step where n = METHOD - 1. Thus if METHOD = 1, all steps are taken in the steepest descent direction.

Let the current iterate initial state be denoted \( \mathbf{r}, \mathbf{v} \). Let the scalar error of the auxiliary parameters corresponding to this state be denoted \( \epsilon \). Let the perturbation size for the sensitivities be \( \mathbf{dv} \).

The current gradient \( \mathbf{g}_c \) is computed by numerical differencing. For the k-th component of \( \mathbf{g}_c \) the corresponding component of velocity is perturbed by \( \mathbf{dv} \).

\[
\mathbf{v}_p = \mathbf{v} + \mathbf{dv} \left[ \delta_{1k}, \delta_{2k}, \delta_{3k} \right]^T
\]

The initial state \( (\mathbf{r}, \mathbf{v}_p) \) is then propagated to the final stopping conditions. Let the auxiliary parameters of that trajectory be denoted \( \mathbf{a}_p \).

The error associated with the perturbed state is then

\[
\epsilon_p = \mathbf{w} \cdot (\mathbf{a}_p - \mathbf{a}^*)
\]

where \( \mathbf{w} \) represents the weighting factors and \( \mathbf{a}^* \) are the desired target conditions. The k-th component of the current gradient is then

\[
\mathbf{g}_{ck} = \frac{\epsilon_p - \epsilon}{\mathbf{dv}}
\]

The corrected gradient is given by

\[
\mathbf{g}_c = \mathbf{g}_{ck}
\]

steepest descent step

\[
= \frac{|\mathbf{g}_{ck}|^2}{|\mathbf{g}_p|^2} \mathbf{v}_p + \mathbf{g}_c
\]

conjugate gradient step

where the subscript \( c \) refers to a current parameter, \( p \) refers to a previous-step parameter.
The unit vector in the direction of the next step is then given by

$$\vec{q}_c = -\frac{\vec{p}_c}{p_c} \quad (5)$$

The directional derivative of the scalar error in the direction $\vec{q}_c$ is

$$d = \vec{g}_c \cdot \vec{q}_c \quad (6)$$

The nominal step size $\bar{h}$ is computed from a linear approximation to null the error

$$\bar{h} = \frac{\varepsilon}{-d} \quad (7)$$

The initial state corrected by this nominal correction is then propagated to the final stopping conditions and the resulting error $\overline{\varepsilon}$ computed. The three conditions

$$y(o) = \varepsilon$$

$$y(h) = \overline{\varepsilon}$$

$$y'(o) = d \quad (8)$$

may now be applied to the formula of a parabola $y - \varepsilon^* = a(x - h^*)^2$ to predict the optimal step size $h^*$ yielding the minimum error $\varepsilon^*$

$$h^* = \frac{-2\frac{dh}{\varepsilon}}{2(dh + \varepsilon + \overline{\varepsilon})} \quad (9)$$

The correction for the current is then given by

$$\Delta v = h^* \vec{q}_g \quad (10)$$

Set KOMP = 0, set up accuracy level, perturbation Δv, and save nominal auxiliary values \( \vec{\alpha}_1 \).

Set up ISP2, ICL2 Flags based on ISTOP flag.

A

KOMP = KOMP + 1

B

\[ \vec{v} = \vec{v} + \Delta \vec{v}(KOMP) \]

Call VMP to integrate trajectory to stopping conditions.

Did trajectory miss SOI and ISTOP \( \neq 1 \)

NO

Call TAROPT(3) to compute and store trajectory target parameters \( \vec{\alpha}_p \) and auxiliary parameters \( \vec{\alpha}_p^* \).

B

Do target parameters \( \tau_p \) satisfy tolerances?

YES

IEND = 1

RETURN

NO

Compute error of component \( \varepsilon_k \)

\[ \varepsilon_k = \vec{v} \cdot (\vec{\alpha} - \vec{\alpha}^*) \]

and comp of grad = \( \delta_k = \frac{\varepsilon_k}{\Delta v_k} \)

A

KOMP = NOPAR

C
C

Does $\text{IT} = 1$ or $\text{IT} = 0 \pmod{\text{KREK}}$?

\[ p = \frac{|g_c|^2 p + g_c}{|g_p|^2} \]

\[ p_c = \frac{g_c}{p} \]

\[ q_c = -\frac{p_c}{p} \]

\[ \frac{\partial}{\partial x} = \text{grad} \epsilon \cdot q_c \]

\[ h = \epsilon / |\partial x| \]

Integrate nominally corrected state (VMP) and call TAROFT to compute auxiliary values to generate nominal error $\tilde{\epsilon}$.

Compute optimal step size $h^*$ by parabolic fit.

Compute correction $\Delta \bar{v} = h^* q_c$.

Update parameters for next iterate

\[ \text{IT} = \text{IT} + 1 \]
\[ p_p = \bar{p} \]
\[ |g_p| = |g_0| \]

RETURN
SUBROUTINE DYNO

PURPOSE: COMPUTE DYNAMIC NOISE COVARIANCE MATRIX IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL DYNO(ICODE)

ARGUMENTS: ICODE I ZERO IN ERROR ANALYSIS MODE

SUBROUTINES SUPPORTED: ERRANN SETEYN GUIDM PRED

LOCAL SYMBOLS: D2 SQUARE OF (DELM^TM)

COMMON COMPUTED: Q

COMMON USED: DELTM DMGN IDNF TM
DYNO Analysis

Subroutine DYNO evaluates the dynamic noise covariance matrix $Q$ over the time interval $\Delta t = t_{k+1} - t_k$. The matrix $Q$ is assumed to have the form

$$Q = \text{diag}\left( \frac{1}{6} K_1 \Delta t^4, \frac{1}{4} K_2 \Delta t^4, \frac{1}{4} K_3 \Delta t^4, K_1 \Delta t^2, K_2 \Delta t^2, K_3 \Delta t^2 \right)$$

where dynamic noise constants $K_1$, $K_2$, and $K_3$ have units of km$^2$/sec$^4$. 
SUBROUTINE DYNOS

PURPOSE: COMPUTE DYNAMIC NOISE COVARIANCE MATRIX AND THE ACTUAL DYNAMIC NOISE (UNMODELED ACCELERATION) IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL DYNOS(ICODE)

ARGUMENTS: ICODE I INTERNAL CODE TO DETERMINE IF THE DYNAMIC NOISE MATRIX IS COMPUTED OR IF THE ACTUAL DYNAMIC NOISE IS CALCULATED

SUBROUTINES SUPPORTED: SIMULL SETEVS GUISIM PRESIM

LOCAL SYMBOLS: DT INTERNAL TIME INCREMENT
D2 SQUARE OF (DELM*TM)
IC INTERNAL CODE ON DT CALCULATION
T1 CURRENT TIME
T2 CURRENT TIME + DELTA TIME

COMMON COMPUTED/USED: W
COMMON COMPUTED: Q
COMMON USED: DELTM DMCN HALF IDNF TM TRTM1 TTIM1 TTIM2 UNMAC ZERO
DYNOS Analysis

Subroutine DYNOS performs two functions. Its first function is identical to that of subroutine DYNJ, namely, to evaluate the dynamic noise covariance matrix $Q$ over the time interval $\Delta t = t_{k+1} - t_k$.

The second function of subroutine DYNOS is to compute the actual dynamic noise $\overrightarrow{\omega}_{k+1}$, which represents the integrated effect of unmodelled accelerations acting on the spacecraft over the time interval $\Delta t$. Actual dynamic noise $\overrightarrow{\omega}_{k+1}$ is used elsewhere in the program to compute the actual state deviations of the spacecraft from the most recent nominal trajectory.

If we define $\overrightarrow{\omega}_{k+1} = \begin{bmatrix} \overrightarrow{\omega}_{r_{k+1}} & \overrightarrow{\omega}_{v_{k+1}} \end{bmatrix}^T$, where

\[
\begin{align*}
\overrightarrow{\omega}_{r_{k+1}} &\quad \text{and} \quad \overrightarrow{\omega}_{v_{k+1}}
\end{align*}
\]

denote the contributions of unmodelled accelerations to spacecraft position and velocity, respectively, and if we assume constant unmodelled acceleration $\overrightarrow{a}$, then

\[
\begin{align*}
\overrightarrow{\omega}_{r_{k+1}} &= \frac{\overrightarrow{a}}{2} (t_{k+1} - t_k)^2 + \overrightarrow{\omega}_v (t_{k+1} - t_k) + \overrightarrow{\omega}_r \\
\overrightarrow{\omega}_{v_{k+1}} &= \overrightarrow{a} (t_{k+1} - t_k) + \overrightarrow{\omega}_v
\end{align*}
\]

The program permits the entire trajectory to be divided into three arbitrary consecutive intervals, over each of which a different constant unmodelled acceleration $\overrightarrow{a}$ can be specified. These intervals are represented by $(t_0, t_1)$, $(t_1, t_2)$, and $(t_2, t_f)$, where $t_0$ is the initial trajectory time and $t_f$ is the final trajectory time. If $t_k$ and $t_{k+1}$ occur in different intervals, then the above equations must be evaluated piece-wise over $(t_k, t_{k+1})$. 
Compute dynamic noise covariance matrix $Q$.

Initialize $\omega$ to zero at $t_k$.

Compute $\Delta t^2$.

Compute actual dynamic noise $\omega$ at the end of the interval $\Delta t$.

$I = 1$ (1st interval)

$\Delta t = t_{k+1} - t_k$

$IC = 1$

$I = 2$

$\Delta t = t_{k+1} - t_1$

$IC = 1$

$I = 3$

$\Delta t = t_{k+1} - t_2$

$IC = 1$

$I = 2$

$\Delta t = t_2 - t_k$

$IC = 1$
SUBROUTINE EIGHY

PURPOSE: TO CONTROL THE COMPUTATION OF EIGENVALUES, EIGENVECTORS, AND HYPERELLIPSOIDS.

CALLING SEQUENCE: CALL EIGHY(VEIG,FOX,HARG,IFMT)

ARGUMENTS:
VEIG I MATRIX TO BE DIAGONALIZED
FOX I FINAL OFF-DIAGONAL ANNihilation VALUE
HARG I MATRIX FOR WHICH THE HYPERELLIPSOID IS TO BE COMPUTED
IFMT I FORMAT FLAG
   =1, PRINT POSITION EIGENVALUE TITLE
   =2, PRINT VELOCITY EIGENVALUE TITLE
   =3, PRINT EIGENVALUE TITLE

SUBROUTINES SUPPORTED: SETEVS GUISSIM GUISS PRESIM GUIDM

SUBROUTINES REQUIRED: HYELS JACOBI

LOCAL SYMBOLS:
EGVCT EIGENVECTOR MATRIX
EGVL EIGENVALUE MATRIX
OUT SQUARE ROOTS OF EIGENVALUES

COMMON USED: IMYP1
EIGHY Flow Chart

ENTER

Call JACOBI to compute eigenvalues and eigenvectors of the input matrix.

Write out eigenvalues, their square roots, and eigenvectors.

INHYPI = 2 ?

YES

INHYPI = 1 ?

YES

RETURN

NO

INHYPI = 1 ?

NO

Call HYELS to compute and write out the 1σ hyperellipsoid associated with the input matrix.

Call HYELS to compute and write out the 3σ hyperellipsoid associated with the input matrix.
SUBROUTINE ELCAR

PURPOSE: TRANSFORMATION OF CONIC ELEMENTS TO CARTESIAN COORDINATES


ARGUMENTS
GM I GRAVITATIONAL CONSTANT OF CENTRAL BODY
A I SEMIMAJOR AXIS
E I ECCENTRICITY
W I ARGUMENT OF PERIAPSIS
XI I INCLINATION IN REFERENCE SYSTEM
XM I LONGITUDE OF ASCENDING NODE
TA I TRUE ANOMALY
R(3) O POSITION VECTOR IN REFERENCE SYSTEM
RM O POSITION MAGNITUDE
V(3) O VELOCITY VECTOR IN REFERENCE SYSTEM
VM O VELOCITY MAGNITUDE
TFP O TIME FROM PERIAPSIS

SUBROUTINES SUPPORTED: DATAS VMP NONLIN COPINS NOMINS DATA HELIO MULTAR CPROP

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AUXF ECCENTRIC ANOMALY (HYPERBOLIC CASE)
AVA MEAN ANOMALY (ELLIPTIC CASE)
CI COSINE OF INCLINATION
CK VELOCITY FACTOR USED TO CALCULATE FINAL VELOCITY VECTOR
CN COSINE OF LONGITUDE OF ASCENDING NODE
COSEA COSINE OF ECCENTRIC ANOMALY (ELLIPTIC CASE)
CT COSINE OF TRUE ANOMALY
CW  COSINE OF SUM OF ARGUMENT OF PERIAPSIS AND TRUE ANOMALY/ COSINE OF ARGUMENT OF PERIAPSIS
DIV  THE SUM 1.4E*(COS(TA/RAD)). USED AS A DIVISOR IN SUBSEQUENT EQUATIONS TO CALCULATE TFP
EA  ECCENTRIC ANOMALY (ELLIPTIC CASE)
P  SEMI-LATUS RECTUM
RAD  DEGREES TO RADIANS CONVERSION FACTOR
SINEA  SINE OF ECCENTRIC ANOMALY (ELLIPTIC CASE)
SINHF  HYPERBOLIC SINE OF AUXF
SI  SINE OF INCLINATION
SM  SINE OF LONGITUDE OF ASCENDING NODE
ST  SINE OF TRUE ANOMALY
SW  SINE OF THE SUM OF ARGUMENT OF PERIAPSIS AND TRUE ANOMALY/ SINE OF ARGUMENT OF PERIAPSIS
TANG  INTERMEDIATE VARIABLE USED TO CALCULATE SINHF
ELCAR Analysis

ELCAR transforms the standard conic elements of a massless point referenced to a gravitational body to cartesian position and velocity components with respect to that body.

Let the gravitational constant of the body be denoted \( \mu \) and the given conic elements \((a, e, i, \omega, \Omega, f)\). The semilatus rectum \( p \) is

\[
p = a (1 - e^2)
\]

(1)

Then the magnitude of the radius vector is given by

\[
r = \frac{p}{1 + e \cos f}
\]

(2)

The unit vector in the direction of the position vector is

\[
\begin{align*}
\frac{u}{x} &= \cos(\omega + f) \cos \Omega - \cos i \sin(\omega + f) \sin \Omega \\
\frac{u}{y} &= \cos(\omega + f) \sin \Omega + \cos i \sin(\omega + f) \cos \Omega \\
\frac{u}{z} &= \sin(\omega + f) \sin i
\end{align*}
\]

(3)

The position vector \( \vec{r} \) is therefore

\[
\vec{r} = r \hat{\Omega}
\]

(4)

The velocity vector \( \vec{v} \) is given by

\[
\begin{align*}
\frac{v}{x} &= \sqrt{\frac{\mu}{p}} \left[ (e + \cos f)(-\sin \omega \cos \Omega - \cos i \sin \Omega \cos \omega) \\
&\quad - \sin f (\cos \omega \cos \Omega - \cos i \sin \Omega \sin \omega) \right] \\
\frac{v}{y} &= \sqrt{\frac{\mu}{p}} \left[ (e + \cos f)(-\sin \omega \sin \Omega + \cos i \cos \Omega \cos \omega) \\
&\quad - \sin f (\cos \omega \sin \Omega + \cos i \cos \Omega \sin \omega) \right] \\
\frac{v}{z} &= \sqrt{\frac{\mu}{p}} \left[ (e + \cos f) \sin i \cos \omega - \sin f \sin i \sin \omega \right]
\end{align*}
\]

(5)

The conic time from periapsis \( t_p \) is computed from different formulae depending upon the sign of the semi-major axis. For \( a > 0 \) (elliptical motion)

\[
t_p = \sqrt{\frac{3}{\mu}} (E - e \sin E)
\]
\[
\cos E = \frac{e + \cos f}{1 + e \cos f} \quad \quad \sin E = \frac{\sqrt{1 - e^2} \sin f}{1 + e \cos f} \tag{6}
\]

For \(a < 0\) (hyperbolic motion) the time from periapsis is:

\[
\tau_p = \sqrt{\frac{a^3}{\mu}} \left( e \sinh H - H \right)
\]

\[
\tanh \frac{H}{2} = \sqrt{\frac{e - 1}{e + 1}} \tan \frac{f}{2} \tag{7}
\]
SUBROUTINE EPHEM

PURPOSE: TO COMPUTE THE CARTESIAN STATE OF DESIRED OBJECTS AT
SPECIFIED TIMES ACCORDING TO TWO OPTIONS:
(1) ECLIPTIC COORDINATES OF ONE BODY RELATIVE TO ITS
REFERENCE BODY (SUN FOR PLANETS, EARTH FOR MOON)
(2) ECLIPTIC COORDINATES OF ALL GRAVITATIONAL BODIES
RELATIVE TO THE INERTIAL COORDINATE SYSTEM (EITHER
HELI OCENTRIC OR BARYCENTRIC).

CALLING SEQUENCE: CALL EPHEM(IC,D,N)

ARGUMENTS:  
D  I  JULIAN DATE OF REFERENCE TIME (REFERENCED
      1950)
IC  I  FLAG SET EQUAL TO 1 FOR OPTION 1 AND TO 0
     FOR OPTION 2
N  I  NUMBER OF GRAVITATIONAL BODIES TO BE
     COMPUTED

SUBROUTINES SUPPORTED:
HELIO  LAUNCH  LUNAR  MULCON  MULTAR
EXECUTE TRAPAR VMP  DATAS  PCTM
PRINT4  PSIM  TRAKS  GUI SIM  GUISS
PRINTS4  DATA  PRINT3  TRAKM  GUIDM
GUID

SUBROUTINES REQUIRED:
CENTER

LOCAL SYMBOLS: A  SEMI-MAJOR AXIS OF LUNAR CONIC
DD  ONE TEN-THOUSANDTH TIMES THE INPUT
     ARGUMENT D FOR COMPUTATIONS IN FN1, FN2
E  ECCENTRICITY OF LUNAR CONIC
ECAM  ECCENTRIC ANOMALY USED TO SOLVE KEPLER
      EQUATION
ECC  ECCENTRICITY USED TO SOLVE KEPLER EQUATION
EM  MEAN ANOMALY OF LUNAR CONIC
E2  E SQUARED
E3  E CUBED
FCDR  VELOCITY DIVIDED BY RADIUS
FN1  STATEMENT FUNCTION DEFINING A THIRD ORDER
     POLYNOMIAL, USED IN COMPUTATION OF MEAN
     ANOMALY OF INNER PLANETS AND OF MOON
STATEMEN/ FUNCTION DEFINING A FIRST ORDER POLYNOMIAL USED IN MEAN ANOMALY COMPUTATIONS OF THE OUTER PLANETS

INDEX FOR LOGIC CONTROL

INCREMENT COUNTER IN SOLUTION OF KEPLER EQUATION

INDEX, ROW OF $F$ OF LUNAR COORDINATES

INDEX, ROW OF $F$ OF COORDINATES OF THE $I$-TH PLANET

INTERMEDIATE VARIABLE USED TO NORMALIZE CONIC ANGLES

INTERNAL CODE WHICH DETERMINES IF COORDINATES OF EARTH ARE BEING CALCULATED IN ORDER TO COMPUTE THOSE OF MOON

INTERNAL CODE WHICH DETERMINES IF COORDINATES OF EARTH HAVE BEEN COMPUTED PRIOR TO COMPUTING THOSE OF THE MOON

INDEX USED IN CALCULATION OF MEAN ANOMALY

SEMI-LATUS RECTUM

TWO TIMES THE MATHEMATICAL CONSTANT PI

HELIOPHICIENT RADIUS OF PLANET

ARRAY OF TRIGONOMETRIC FUNCTIONS OF SPECIFIED ANGLES

VELOCITY OF PLANET

X-COMPONENT OF INTERMEDIATE VECTOR, $W$

Y-COMPONENT OF INTERMEDIATE VECTOR, $W$

Z-COMPONENT OF INTERMEDIATE VECTOR, $W$

CO/MON CO/PUTED/USED:

ELMNT, $F$, $T$, $XP$

CO/MON USED:

CN, EMN, IBARY, NBODY, NO

ONE, PMASS, ST, TWOPI, TWO
ZERO
EPHEM Analysis

EPHEM first determines the current value for the mean anomaly of the pertinent body. The mean anomaly \( M \) is computed from

\[
M = M_0 + M_1 t + M_2 t^2 + M_3 t^3 \quad \text{for inner planets}
\]
\[
M = M_0 + M_1 t \quad \text{for outer planets}
\]
\[
M = L_0 + L_1 t + L_2 t^2 + L_3 t^3 - \dot{\omega}(t) \quad \text{for the moon}
\]

Kepler's equation \( M = E - e \sin E \) is then solved iteratively to determine the eccentric anomaly \( E \). The subsequent computations are basic conic manipulations:

\[
p = a(1 - e^2)
\]
\[
r = a(1 - e \cos E)
\]
\[
v = \sqrt{\mu \left( \frac{1}{r} - \frac{1}{a} \right)}
\]
\[
\cos f = \frac{p - r}{er} \quad \sin f = \sqrt{1 - \cos^2 f} \quad \text{sgn}(\sin E)
\]
\[
\cos \gamma = \frac{\sqrt{\mu r}}{rv} \quad \sin \gamma = \sqrt{1 - \cos^2 \gamma} \quad \text{sgn}(\sin E)
\]
\[
\omega = \dot{\omega} - \Omega
\]

The cartesian position and velocity relative to the reference body are then

\[
\overrightarrow{r} = r_x \hat{i} + r_y \hat{j} + r_z \hat{k}
\]
\[
r_x = r \cos(\omega + f) \cos \Omega - r \sin(\omega + f) \sin \Omega \cos i
\]
\[
r_y = r \cos(\omega + f) \sin \Omega + r \sin(\omega + f) \cos \Omega \cos i
\]
\[
r_z = r \sin(\omega + f) \sin i
\]
\[
\overrightarrow{v} = \frac{v}{r} \left[ (\hat{\omega} \times \overrightarrow{r}) \cos \gamma + \overrightarrow{r} \sin \gamma \right]
\]

where \( \hat{\omega} = (\sin i \sin \Omega) \hat{i} - (\sin i \cos \Omega) \hat{j} + (\cos i) \hat{k} \)
When option 1 is used, the reference body for all the planets is the sun while the reference body for the moon is the earth.

When option 2 is used with heliocentric inertial coordinates, the cartesian state of the earth is added to the cartesian state of the moon to convert the state of the moon to heliocentric coordinates before storing that state in the Y-array.

When option 2 is used with barycentric inertial coordinates, subroutine CENTER is called to convert all elements to barycentric coordinates before storing in the Y-array.
PROGRAM ERRAN

PURPOSE: TO CONTROL THE COMPUTATIONAL FLOW THROUGH THE BASIC CYCLE (MEASUREMENT PROCESSING) AND ALL EVENTS IN THE ERROR ANALYSIS MODE.

SUBROUTINES SUPPORTED: ERRON

SUBROUTINES REQUIRED: SCHED NTM PSIM DYNO TLAKM MEMO NAVM PRINT3 SETEVM GUIDM

LOCAL SYMBOLS: AY DUMMY VARIABLE
ICODE EVENT CODE
IPRN MEASUREMENT COUNTER FOR PRINTING
MNCODE MEASUREMENT CODE
NEVENT EVENT COUNTER
NR NUMBER OF ROWS IN THE OBSERVATION MATRIX
TRTM2 TIME OF THE MEASUREMENT

COMMON COMPUTED/USED: ICOWDE MCONTR RI TEVN TRTM1 XF XI

COMMON COMPUTED: DELTM

COMMON USED: FNTM IEVNT IPRINT ISTMC NEV NMN NR NTMC RF TEV
ERRAN Analysis

Subroutine ERRAN controls the computational flow through the basic cycle (measurement processing) and all events in the error analysis mode.

In the basic cycle the first task of ERRAN is to control the generation of the targeted nominal spacecraft state $\mathbf{x}_{k+1}$ at time $t_{k+1}$, given the state $\mathbf{x}_k$ at time $t_k$. Then, calling PDLN, DYNx, TRAXx, and MENx, successively, ERRAN controls the computation of all matrix information required by subroutine NAVM in order to compute the covariance matrix partitions at time $t_{k+1}$ immediately following the measurement.

At an event ERRAN simply calls the proper event subroutine or overlay where all required computations are performed.
ERRAN Flow Chart

ENTER

Initialize event counter NEVENT and print counter IPRN.

Define state $\bar{X}_k$ at time $t_k$.

Call SCHED to obtain the time $t_{k+1}$ of the measurement and the measurement code.

Define time interval $\Delta t = t_{k+1} - t_k$.

Does an event occur before $t_{k+1}$?

YES

290

NO

Call NTM to compute state $\bar{X}_{k+1}$.

Increment measurement counter MCNTR.

Call PSIM to compute state transition matrix partitions over $[t_k, t_{k+1}]$.

Call DYN9 to compute $Q_{k+1,k}$.

Call TRAKh to compute the observation matrix partitions at $t_{k+1}$.
Call MENV@ to compute $R_{k+1}$.

Call NAVM to compute covariance matrix partitions at $t_{k+1}$.

Increment print counter IPRN.

Is it time to print?

NO

YES

Call PRINT3 to write out all basic cycle data.

Reset time and state in preparation for next basic cycle.

$t_{k+1} > t_f$?

YES

RETURN

NO

Have all measurements been processed?

NO

220

YES

Have all events been performed?

NO

RETURN

YES

$t_{k+1} = t_f$?

NO

Define state $\overline{x}_{k+1}$ at time $t_{k+1}$.

290

NO
Define time interval $\Delta t = t_f - t_k$ and state $\overline{X}_{k+1}$.

Call NTM to compute state $\overline{X}_f$.

Call PSIM to compute state transition matrix partitions over $[t_{k+1}, t_f]$.

Call DYNØ to compute $Q_{f,k+1}$.

Call NAVM to compute covariance matrix partitions at $t_f$.

RETURN
Define event code ICODE and event time $t_1$.

ICODE < 5?

YES

Call SETEVN to compute information common to most types of events.

ICODE = ?

1. Call prediction event overlay.
2. Call CONCO event overlay.
3. Call GUIDM.
5. Write: Adaptive filtering event not available with error analysis.
6. Increment event counter.

296
PROGRAM  ERRON
PURPOSE: TO CONTROL THE ERROR ANALYSIS OVERLAY SCHEME
SUBROUTINES SUPPORTED: NONE
SUBROUTINES REQUIRED: DATA  ERRAN  FRNTS3
LOCAL SYMBOLS: IRUNX  TOTAL NUMBER OF DATA CASES
                IRUN  DATA CASE COUNTER
SUBROUTINE ESTMT

PURPOSE: TO UPDATE THE FINAL VALUES OF THE PRECEDING COMPUTATION INTERVAL WHICH SERVE AS INITIAL VALUES FOR THE NEW STEP, TO DETERMINE THE DESIRED SIZE OF THE NEXT TIME INCREMENT ON THE BASIS OF TRUE ANOMALY OR REQUESTED PRINTTIME, AND TO ESTIMATE THE FINAL POSITION AND MAGNITUDE OF THE VIRTUAL MASS.

CALLING SEQUENCE: CALL ESTMT(D1,DELM,T1)

ARGUMENTS
D1 I JULIAN DATE, EPOCH 1900, OF THE INITIAL TRAJECTORY TIME
DELM I TIME INTERVAL OVER WHICH THE TRAJECTORY WILL BE PROPAGATED (DAYS)
T1 I INITIAL TRAJECTORY TIME (DAYS) REFERENCED TO INJECTION

SUBROUTINES SUPPORTED: VM

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: INCHNT V
COMMON COMPUTED: ITRAT KOUNT
COMMON USED: INCPR INC IPR
ESTMT Analysis

The initial values of the state variables are first set equal to the values at the end of the previous interval. The nominal time interval to be used during the current step is computed from

$$\Delta t_k = \frac{c_2}{V_{S_B}}$$

(1)

where \(c_2\) is the constant input true anomaly increment relative to the virtual mass trajectory.

The time interval to the final time \(t_f\) or to the next time printout \(t_p\) is computed and the current time interval \(\Delta t\) is adjusted if necessary.

Finally the virtual mass final position and magnitude are estimated by the expansions

$$\mu_{V_E} = \mu_{V_B} + \dot{\mu}_{V_B} \Delta t + \ddot{\mu}_{V_B} \Delta t^2$$

$$\vec{r}_{V_E} = \vec{r}_{V_B} + \vec{r}_{V_B} \Delta t + \vec{a}_{sv} \Delta t^2$$

(2)
ESTMT Flow Chart

ENTER

Update state variables for next step.

First interval? NO Is printout to occur after this increment?

YES

KOUNT=1

NO

ITERAT=1

Calculate nominal value of $\Delta t_k$.

$\Delta t_k = t_f - t_p$

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

YES

YES

YES

YES

YES

Estimate final magnitude $\mu_{vE}$ and position $\bar{r}_{vE}$ of virtual mass.

RETURN

$\Delta t_k = t_f - t_p$

$\Delta t_k = t_f - t_p$

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$

KOUNT = 1

$\Delta t_k = t_f - t_B$

$\Delta t_k = t_f - t_E$
SUBROUTINE EULMX

PURPOSE: TO COMPUTE THE MATRIX REQUIRED TO DEFINE TRANSFORMATIONS FROM ONE COORDINATE SYSTEM TO ANOTHER.

CALLING SEQUENCE: CALL EULMX(ALP,NN,BET,MM,GAM,LL,P)

ARGUMENTS:    ALP        I FIRST ROTATION ANGLE (RADIANS)
                NN         I FIRST AXIS OF ROTATION
                BET        I SECOND ROTATION ANGLE (RADIANS)
                MM         I SECOND AXIS OF ROTATION
                GAM        I THIRD ROTATION ANGLE (RADIANS)
                LL         I THIRD AXIS OF ROTATION
                P(3,3)     O TRANSFORMATION MATRIX

SUBROUTINES SUPPORTED: PECEQ

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:    A       INTERMEDIATE ROTATION MATRIX
                    ALPHA   TEMPORARY LOCATION FOR EACH OF THE ROTATION ANGLES: ALP, BET, AND GAM
                    D       INTERMEDIATE PRODUCT MATRIX
                    F       TRANSFORMATION MATRIX FOR ANGLE ALP
                    G       TRANSFORMATION MATRIX FOR ANGLE BET
                    H       TRANSFORMATION MATRIX FOR ANGLE GAM
                    N       COUNTER SHOWING NUMBER OF COORDINATE AXES FOR WHICH CALCULATIONS REMAIN
                    NAXIS   TEMPORARY LOCATION FOR EACH OF THE AXES OF ROTATION: NN, MM, AND LL

COMMON USED: ONE ZERO
SUBROUTINE EXCUT

PURPOSE: CONTROL EXECUTION OF A VELOCITY CORRECTION MODELED AS AN IMPULSE SERIES IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL EXCUT

SUBROUTINES SUPPORTED: GUIDK

SUBROUTINES REQUIRED: PREPUL PULCOV PULSEX

COMMON COMPUTED/USED: XXIN

COMMON COMPUTED: QK

COMMON USED: DELPX, DIPX, TM, INPX
SUBROUTINE EXECUTE

PURPOSE: TO CONTROL THE ACTUAL EXECUTION OF THE VELOCITY INCREMENT DELTAV.

CALLING SEQUENCE: CALL EXECUTE

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: PREPUL PULSEX CAREL PECEQ

LOCAL SYMBOLS:
- A: SEMIMAJOR AXIS OF DOMINANT BODY CONIC
- DVM: MAGNITUDE OF VELOCITY INCREMENT
- E: ECCENTRICITY OF DOMINANT BODY CONIC
- INDEX: CODE OF BODY BEING TESTED FOR DOMINANT BODY
- IND: INDEX OR CODE OF DOMINANT BODY
- ISUN: SUN VALUE OF IND
- I: INDEX
- JX: INDEX OF S/C-REL-TO-BODY ROW OF F-ARRAY
- MODEL: EXECUTION MODEL (1=IMPULSIVE, 2=PULSE ARC)
- PP: UNIT VECTOR TO PERIAPSIS IN ORBITAL PLANE
- QQ: UNIT VECTOR NORMAL TO PP IN ORBITAL PLANE
- RN: POSITION AND VELOCITY OF S/C AT END OF EXECUTION BY PULSING ARC
- RSI: POSITION VECTOR OF S/C RELATIVE TO DOMINANT BODY AT EXECUTION TIME
- RTB: RADIUS MAGNITUDE TO BODY BEING TESTED FOR DOMINANT BODY
- TA: TRUE ANOMALY ON DOMINANT BODY CONIC
- TFP: TIME FROM PERIAPSIS ON DOMINANT BODY CONIC
- VSI: VELOCITY VECTOR OF S/C RELATIVE TO DOMINANT BODY AT EXECUTION TIME
- WN: UNIT NORMAL TO ORBITAL PLANE
- W: ARGUMENT OF PERIAPSIS OF DOMINANT BODY
**CONIC**

**XI**  
INCLINATION OF DOMINANT BODY CONIC

**XMU**  
GRAVITATIONAL CONSTANT OF DOMINANT BODY

**XN**  
LONGITUDE OF ASCENDING NODE OF DOMINANT BODY

**COMMON COMPUTED/USED:**  
DELTAV  D1  RIN  TRTM

**COMMON COMPUTED:**  
KTIM

**COMMON USED:**  
ALNGTH  DELV  F  KUR  MDL  
NBOD  NB  PMASS  PULT  SPHERE  
TM  TWO  V

**EXECUTE-B**
EXCUTE Analysis

EXCUTE is the executive subroutine controlling the actual execution of the velocity increment $\Delta v$. The $\Delta v$ is computed by TARGET or INSEBS or read in by the user.

Before executing the correction EXCUTE computes peripheral information of interest to the user. It first determines the dominant body acting on the spacecraft. If the spacecraft is in the moon's SOI (with respect to the earth), the moon is the dominant body. If not in the moon's SOI but in any of the planets' SOI (with respect to the sun) that planet is the dominant body. Otherwise the sun is the dominant body.

Having determined the dominant body EXCUTE computes the state of the spacecraft relative to that body. It then computes the conic elements of the trajectory both before and after an impulsive addition of the $\Delta v$ in ecliptic coordinates.

If the dominant body is not the sun, it makes the same computations in equatorial coordinates.

EXCUTE then operates on the current value MODEL of the array MDL. If MODEL = 1, the impulsive model of execution is commanded. The $\Delta v$ is therefore added to the current inertial ecliptic velocity before returning to GIDANS.

If MODEL = 2, the pulsing arc model of execution is required. PREPUL is called to perform the preliminary work needed for the pulsing arc. PULSEX then actually propagates the trajectory through the series of pulses. At the completion of the arc EXCUTE updates the time and inertial ecliptic state (both position and velocity) of the nominal trajectory to the state determined by PULSEX.
**Excute Flow Chart**

1. **Enter**
   - Compute index and code of dominant body (DB).
   - Determine spacecraft state to DB and μ of DB.
   - Compute ecliptic conic of S/C wrt DB before and after impulsive correction.
   - Compute equatorial conic of S/C wrt DB before and after impulsive correction.

2. **Model**
   - Case 1: \[ \vec{v} = \vec{v} + \Delta \vec{v} \]
   - Case 2

3. Call PREPUL for preliminary work for pulses.
4. Call PULSEX for actual execution by pulse model.
5. Update time and state to values at end of arc.

**RETURN**
SUBROUTINE EXCUTS

PURPOSE  CONTROL EXECUTION OF A VELOCITY CORRECTION MODELED AS AN IMPULSE SERIES IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL EXCUTS

SUBROUTINES SUPPORTED: GUISIM

SUBROUTINES REQUIRED: PREPUL, PULCOV, PULSEX

LOCAL SYMBOLS  RN  EFFECTIVE SPACECRAFT STATE AFTER A VELOCITY CORRECTION MODELED AS AN IMPULSE SERIES

COMMON COMPUTED/USED:  XXIN

COMMON COMPUTED:  QK

COMMON USED:  DELPX, DIPX, TM, INPX
EXCUTS Flow Chart

ENTRY

CALL PREPUL

IF INFX = 2

YES
CALL PULQOV

NO
CALL PULSEX

RETURN
SUBROUTINE FLITE

PURPOSE: TO SOLVE THE TIME OF FLIGHT EQUATION (LAMBERT'S THEOREM) USING BATTIN'S UNIVERSAL EQUATION FORMULATION.

CALLING SEQUENCE: CALL FLITE(R1, R2, THETA, GM, TF, A, E, K)

ARGUMENTS: R1 I INITIAL RADIUS
            R2 I FINAL RADIUS
            THETA I CENTRAL ANGLE
            GM I GRAVITATIONAL CONSTANT
            TF I TIME OF FLIGHT
            A O SEMIMAJOR AXIS
            E O ECCENTRICITY
            K O ERROR CODE
                0 NO ERROR
                1 ERROR

SUBROUTINES SUPPORTED: HELIO

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AMISS ERROR IN ITERATE
                BIGNO CONSTANT = E+25
                B1 CONSTANT = S 3/2
                CHECK ERROR IN ITERATE
                CX BATTIN C-FUNCTION OF X
                CY BATTIN C-FUNCTION OF Y
                C CHORD LENGTH
                DEM INTERMEDIATE VARIABLE
                P SEMILATUS RECTUM
                ROOT INTERMEDIATE VARIABLE
                SLOP VALUE OF DERIATIVE OF T(X)
                SX BATTIN S-FUNCTION OF X
FLITE-B

SY  BATTIN S-FUNCTION OF Y
S1  SEMIPERIMETER
S  =INTERMEDIATE VARIABLE (=1-C/S1)
TIME FLIGHT TIME CORRESPONDING TO ITERATE X
T  NORMALIZED TIME OF FLIGHT
U  FLAG SET TO 1 IF X LESS THAN PI 2,-1 ELSE
VB1 INTERMEDIATE VARIABLE
V  FLAG SET TO 1 FOR TYPE I,-1 FOR TYPE II
X1 STARTING VALUE FOR X
X  VARIABLE INTRODUCED TO REPLACE A
Y  INTERMEDIATE VARIABLE AS FUNCTION OF X
FLITE Analysis

FLITE solves the time of flight equation (Lambert's theorem) using Battin's universal equation formulation. Stated functionally, Lambert's theorem states that the time of flight $t_f$ is a function

$$t_f = t_f(r_1 + r_2, c, a) \quad (1)$$

solely of the sum $r_1 + r_2$ of the distances of the initial and final points of the trajectory from the central body, the length $c$ of the chord joining these points, and the length $a$ of the semimajor axis of the trajectory. Usually the time of flight is known and it is desired to solve for the semimajor axis. The standard formulation involves different equations for the elliptic, parabolic, and hyperbolic cases, all of which then iterate on $a$ to determine the solution.

In Battin's approach the semimajor axis $a$ is replaced by a new variable $x$. By further introducing two new transcendental functions $S(x)$ and $C(x)$, the special cases of the flight-time equation are combined into one single, better behaved formula. The functions $S(x)$ and $C(x)$ are defined by

$$S(x) = \frac{\sqrt{x} - \sin \sqrt{x}}{3} \quad C(x) = \frac{1 - \cos \sqrt{x}}{x} \quad x > 0$$

$$S(x) = \frac{\sinh \sqrt{-x} - \sqrt{-x}}{3} \quad C(x) = \frac{\cosh \sqrt{-x} - 1}{-x} \quad x < 0$$

$$S(x) = \frac{1}{6} \quad C(x) = \frac{1}{2} \quad x = 0 \quad (2)$$

A parameter $Q$ is introduced as

$$Q = \frac{a-c}{s}$$

where

$$c = (r_1^2 + r_2^2 - 2r_1 r_2 \cos \theta)^{\frac{1}{2}}$$

$$s = \frac{1}{2} (r_1 + r_2 + c) \quad (3)$$

The universal flight-time formula is
where \( T = \sqrt[3]{\frac{\mu}{a}} t_f \). The choice of the upper or lower sign is made according to whether the transfer angle \( \theta \) is less or greater than 180° respectively.

The development of equations (4) is too long and complex to be given here. It may be obtained from the first reference listed below. The following steps of that reference are noted:

1. The two body problem on pp. 15, 16.
2. The "vis viva" equation and Kepler's equation on pp. 50, 51.
3. Lambert's theorem proved from Kepler's equation on p. 71.
4. The basic flight-time formula and detailed analysis on pp. 72-78.
5. The universal formulation on pp. 80, 81.

Instead of using the equations (4) the authors of reference 2 (listed below) determined \( y \) as a function of \( x \) as

\[
y = 4 \arcsin^2 \left( \frac{\sqrt{x} s_c(x)}{2} \right) \quad x \geq 0
\]

\[
y = -4 \arcsin^2 \left\{ \sqrt{x} s_c(x) \right\} + \sqrt{\frac{x s_c(x)}{2}} + 1 \frac{1}{2} \quad x < 0
\]

Therefore a single variable iteration is possible. Newton's method is used to solve (4a) given \( T \) and \( Q \) as

\[
x_n = x - \frac{T(x_n) \cdot T}{T'(x_n)}
\]

where

\[
T(x) = \frac{s(x)}{C^{3/2}(x)} + Q^{3/2} \frac{s(y)}{C^{3/2}(y)}
\]

\[
T'(x) = \frac{1 + k \left[ + Q^{3/2} - 1.5 \sqrt{2-yC(y)} \right] T(x)}{2x \sqrt{C(x)}}
\]
As \(|2-yC(y)| \to 0\), \(k \to 1\). Therefore if \(|2-yC(y)| < 10^{-4}\) \(k\) is set to 1. Also \(T'(x)\) breaks down as \(x \to 0\). Therefore the approximation is used:

\[
T'(x) = \frac{1 + \frac{3}{2}Q}{2x^2} \quad |x| < 10^{-6}
\]

The starting value for \(x\) is given by \(x = x_1 - \Delta x(T,Q)\) where

\[
x_1 = 82.1678 + 352.8045 T
- (123954.8504 T^2 + 4304.0083 T + 13423.6819)^{1/2}
\]

\[
\Delta x(T,Q) = \frac{1}{T + 0.15} \left( \frac{2.36}{T} + \frac{3.6}{T + 0.1} \right) (0.3 Q^2 + 0.7 Q)
\]

To insure that the routine will not fail for large or small values of \(T\) certain restrictions on \(T\) are built into the program. The nominal value of \(T\) is forced to be no larger than 950,000 and no smaller than \(10^{-6}\). This forces the corresponding limits for \(x\) of \([-623.0473, 39.14553]\).

Finally convergence is achieved when \(|T(x_n) - T| < \frac{T}{100000}\).

Having solved for semimajor axis \(a\), the semilatus rectum \(p\) is given by

\[
p = \frac{1}{2} \left\{ \frac{r_1 r_2 \sin \theta}{c} \sqrt{\frac{1}{s-c} - \frac{1}{2a}} \pm \frac{\text{sgn} (t - t_m) \sqrt{\frac{1}{s} - \frac{1}{2a}}}{2} \right\}^2
\]

Then the eccentricity \(e\) is given by

\[
e = 1 - \frac{p}{a}
\]

References:


(2) Lesh, H. F., and Travis, C., FLIGHT: a Subroutine to Solve the Flight Time Problem, JPL Space Programs Summary 37-53, Vol. II.
FLITE Flow Chart

ENTER

Enforce restrictions on $T$

Compute zero iterate $x$

Compute $S(x), C(x)$

Compute $S(y), C(y)$

Compute $Q$ and $T(x)$

$|T(x) - T| : \frac{T}{S} \leq 10$

Enforce restriction on $x$

$a = \frac{s}{xC(x)}$

Compute $p,e$

RETURN
SUBROUTINE GHA

PURPOSE: TO COMPUTE THE GREENWICH HOUR ANGLE AND THE UNIVERSAL TIME (IN DAYS) WHICH IS USED IN THE TRACKING MODULE TO ORIENT THE TRACKING STATIONS ON A SPHERICAL ROTATING EARTH.

CALLING SEQUENCE: CALL GHA

ARGUMENTS: NONE

SUBROUTINES SUPPORTED: DATA15 DATA1

LOCAL SYMBOLS:  
  D  NUMBER OF DAYS IN TSTAR
  EQMEG EARTH ROTATION RATE
  GM GREENWICH HOUR ANGLE
  ID INTERMEDIATE VARIABLE
  REFJD JULIAN DATE OF JAN. 0, 1950
  TFRAC FRACTION OF DAY IN TSTAR
  TSTAR JULIAN DATE, EPOCH JAN. 0, 1950, OF INITIAL TRAJECTORY TIME

COMMON COMPUTED:  UNIVT

COMMON USED:  DATEJ EM13
Subroutine GHA computes the Greenwich hour angle in degrees and days at some epoch $T^*$ referenced to 1950 January $1^d0^h$. Epoch $T^*$ is computed from

$$ T^* = J.D.0 + 2415020.0 - J.D.\text{REF} $$

where

$J.D.0 = \text{Julian date at launch time } t_0 \text{ referenced to 1900 January } 0^d12^h.$

$J.D.\text{REF} = \text{Reference Julian date 2433282.5 }$

$= 1950 \text{ January } 1^d0^h \text{ referenced to January } 0^d12^h \text{ of the year } 4713 \text{ B.C.}$

and $2415020.0 = 1900 \text{ January } 0^d12^h \text{ referenced to January } 0^d12^h \text{ of the year } 4713 \text{ B.C.}$

Then $T^*$ is the Julian date at launch time $t_0$ referenced to 1950 January $1^d0^h$.

The Greenwich hour angle corresponding to $T^*$ is given by

$$ \text{GHA}(T^*) = 100.0755426 + 0.985647x + 2.9015 x 10^{-13} d^2 + \omega t $$

where $0 \leq \text{GHA}(T^*) < 360^\circ$.

and $d = \text{integer part of } T^*$, $t = \text{fractional part of } T^*$,

and $\omega = \text{Earth's rotation rate is degrees/day}$.

The Greenwich hour angle in days is given by $\frac{\text{GHA}}{\omega}$.
Define Earth's rotation rate in degrees/day and reference Julian date.

Compute $T^*$ and the integer and fractional parts of $T^*$.

Compute the Greenwich hour angle $GH$ at time $T^*$.

If $GH < 0$?

YES

$GH = GH + 360$

NO

$GH = GH + 360$

YES

$GH - 360$ $\leq 0$?

NO

$GH = GH - 360$

YES

Compute Greenwich hour angle in days.

RETURN
SUBROUTINE GIDANS

PURPOSE: EXECUTIVE ROUTINE FOR COMPUTATION OF REQUIRED GUIDANCE EVENT

CALLING SEQUENCE: CALL GIDANS

SUBROUTINES SUPPORTED: NOMINAL NONLIN

SUBROUTINES REQUIRED: EXECUTE ZERIT TARGET INSERS TIME VMP

LOCAL SYMBOLS:

D TIME  DELTA TIME (DAYS) BETWEEN ORBIT INSERTION COMPUTATION AND EXECUTION

IZER  VECTOR OF CODES USED FOR RETARGETING
IZER(KUR)=0, DO NOT RECOMPUTE ZERO ITERATE
IZER(KUR) #0, RECOMPUTE ZERO ITERATE

I  INDEX

K TYPE  VALUE OF KTYP(KUR) INDICATING TYPE OF EVENT

MODEL  DOES NOT APPEAR IN CURRENT VERSION SEE EXECUTE

COMMON COMPUTED/USED:

KMXQ  KWIT  TIMG

COMMON COMPUTED:

DELV  KTIM  ZDAT

COMMON USED:

DELTAV  KTYP  KUR  MOL  RIM
GIDANS Analysis

GIDANS is an executive routine responsible for processing a guidance maneuver from the computation of the velocity increment $\Delta v$ to the execution of that correction.

Before entry to GIDANS, TRJTRY has computed the index of the current event (KUR) and has integrated the nominal trajectory to the time of the event. GIDANS now evaluates the KUR component of two integer arrays KTYP and KMXQ. The values of these flags determines the operation of GIDANS. The flag KTYP specifies the type of guidance event to be performed while KMXQ prescribes the compute/execute mode to be used according to:

<table>
<thead>
<tr>
<th>KTYP</th>
<th>KMXQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1</td>
<td>Termination event: Compute $\Delta v$ only</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Targeting event: Execute $\Delta v$ only</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Retargeting event: Compute and execute $\Delta v$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Orbit Insertion: Compute but execute $\Delta v$</td>
</tr>
</tbody>
</table>

GIDANS first checks for a termination event. If the current index prescribes such an event, the flag KWIT is set to 1 and a return is made to the main program NOMINAL.

In preparation for a normal guidance event, GIDANS calls VMP with the current spacecraft heliocentric state and a time increment of zero to restore the $F$ and $V$ arrays providing the current geometry of spacecraft and planets. If the current event is an execute-only node, the transfer is made to the execution section of GIDANS for the addition of the pre-set velocity increment.

Otherwise GIDANS interrogates KTYP for the type of maneuver to be computed. For a targeting event, subroutine TARGET is called directly for the computation of the $\Delta v$ necessary to satisfy input target conditions. After calling TARGET the $F$ and $V$ arrays are restored as indicated above.

A retargeting event is defined as a targeting event which requires the computation of a new zero iterate. Thus a retargeting event is an event in which the current nominal state when integrated forward would miss the target conditions badly. Such an event would be the broken-plane correction. For this event TRJTRY stores the current position (and possibly the target position) in the ZDAT array. It then calls ZEROIT for the computation of the massless-planets initial velocity consistent with the target conditions. It then operates identically to the targeting event.

The remaining guidance maneuver is the insertion event. GIDANS calls INSERS for the computation of the velocity increment $\Delta v$ and the time interval $\Delta t$ before it is to be executed.

Subroutines TARGET and INSERS signal trouble to GIDANS via the flag KWIT. If problems are encountered during their operation such as nonconvergence in TARGET or no insertions possible in INSERS KWIT is set to a 1. Otherwise KWIT = 0. Upon return to NOMINAL, if KWIT = 1 the current case will be terminated while KWIT = 0 will allow the current case to continue.
If the current event is a compute-only mode, TRJTRY now sets KWIT = 0 (so that the program will continue regardless of whether the correction computations were successful or not) and returns to NOMINAL. However if the current event failed (KWIT = 1) and was to be executed (KXXQ ≠ 1) CIDANS considers this a fatal error for the current case and returns with KWIT = 1.

If the compute/execute mode is compute-execute later (KXXQ = 4) as is the insertion event, CIDANS now sets up for the subsequent execute-only event. The $\Delta V$ computed is stored in the DELV array, the time of the execution is computed ($t_{ex} = t_k + \Delta t$) and stored in the TIMG array, and the KMXQ flag is set to a 2 (execute-only). The return is then made to NOMINAL.

For an event to be executed at the current time (KXXQ = 2,3) CIDANS now calls EXECUTE for the completion of that task.

It should be noted that for all events that are completed at this time the KUR component of the KTIM array are set equal to 0 so that they are no longer considered in determining the next event in TRJTRY. Only in the case of KXXQ = 4 is the KTIM flag non-zero upon exit from CIDANS.
Set up geometric arrays P,V for current state

If KTYP = 1

Set up ZDAP array.
Call ZEDP.
Call TARGET.
Restore F,V array.

If KTYP = 0

Call TARGET.

If KTYP = -1

Store Δv,t
KMXQ=2
KTIM=1

RETURN

Flag Definitions

KTYP = -1 Termination event
    = 1 Targeting event
    = 2 Retargeting event
    = 3 Orbit insertion

KXXQ = 1 Compute Δv only
    = 2 Execute Δv only
    = 3 Compute and execute Δv
    = 4 Compute but execute Δv later

KWIT = 0 Continue case
    = 1 Problem, terminate case

KTIM = 0 Guidance event to be processed
    = 0 Event already processed.
SUBROUTINE GUID

PURPOSE 
COMPUTE GUIDANCE MATRIX, VARIATION MATRIX, AND TARGET
CONDITION COVARIANCE MATRIX AT A MIDCOURSE GUIDANCE
EVENT IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL GUID

SUBROUTINES SUPPORTED: GUID

SUBROUTINES REQUIRED: EPHEM HYELS JACOBI MATIN NTM
ORB PARTL PSIM STMPR VARADA

LOCAL SYMBOLS
A TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX A
BB TWO-VARIABLE B-PLANE GUIDANCE SUB-MATRIX B
BDRS B DOT R
BDR1 VALUE OF B DOT R RETURNED FROM PARTL (NOT
USED)
BOTS L DOT T
BDT1 VALUE OF B DOT T RETURNED FROM PARTL (NOT
USED)
BS MAGNITUDE OF B VECTOR
B1 VALUE OF B RETURNED FROM PARTL (NOT USED)
D INTERMEDIATE JULIAN DATE
DUM1 ARRAY OF EIGENVECTORS
EGVCT ARRAY OF EIGENVECTORS
EGVL ARRAY OF EIGENVALUES
ICS INTERMEDIATE STORAGE FOR ICL2
ICLS INTERMEDIATE STORAGE FOR ICL
INCMTS INTERMEDIATE STORAGE FOR INCMT
IPR INTERMEDIATE STORAGE FOR IPRINT
ISP INTERMEDIATE STORAGE FOR ISP2
PBR PARTIAL OF B DOT R WITH RESPECT TO STATE
VECTOR
PBT PARTIAL OF B DOT T WITH RESPECT TO STATE
| PHI1  | INTERMEDIATE ARRAY   |
| PHI2  | INTERMEDIATE ARRAY   |
| PHI3  | INTERMEDIATE ARRAY   |
| RI    | NOMINAL SPACECRAFT STATE AT GUIDANCE EVENT |
| RSW   | TARGET CONDITION CORRELATION MATRIX |
| RTPS  | INERTIAL SPACECRAFT STATE AT SPHERE OF INFLUENCE |
| SQP   | TARGET CONDITION STANDARD DEVIATIONS |
| TCA   | TRAJECTORY TIME AT CLOSEST APPROACH |
| TSI   | TRAJECTORY TIME AT SPHERE OF INFLUENCE |
| XCA   | INERTIAL SPACECRAFT STATE AT CLOSEST APPROACH |
| XSIP  | SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE |
| XSIV  | SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE |

**COMMON COMPUTED/USED:**
- ICL2  
- IPRINT  
- ISP2  
- NO  
- XP  
- DELTH  
- EM  
- TRTM1  
- TS011  

**COMMON COMPUTED:**
- ALNGTH  
- BOR  
- BDT  
- B  
- DATEJ  
- DC  
- OSI  
- FNTM  
- FOM  
- F  
- IBARY  
- ICL  
- IHYP1  
- ITMC  
- N800  
- NB  
- NTMC  
- NTP  
- ONE  
- PHI  
- P  
- RC  
- RSI  
- TH  
- VSI  
- ZERO
GUID Analysis

Subroutine GUID is used in the error analysis mode to compute the same quantities which subroutine GUID computes in the simulation mode. Subroutine GUID differs from GUID in that instead of calling NTMS and ARSIM as does GUID, subroutine GUID calls NTM and VARADA. In addition, the state transition and variation matrices computed in GUID are referenced to the targeted nominal since the most recent nominal is not defined for the error analysis mode. These differences entail only minor logic differences in the flow chart for GUID, and for this reason no GUID flow chart is presented. See subroutine GUID analysis and flow chart for further details.
SUBROUTINE GUIDM

PURPOSE CONTROL EXECUTION OF A GUIDANCE EVENT IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL GUIDM

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED: CORREL DYN0 GUID MYELS JACOBI NAVM PSIM STMPR

LOCAL SYMBOLS: ADA VARIATION MATRIX

AMAX INTERMEDIATE VARIABLE USED TO FIND MAXIMUM EIGENVALUE OF VELOCITY CORRECTION COVARIANCE MATRIX (S MATRIX)

CXSU1 STORAGE FOR CXSU KNOWLEDGE COVARIANCE

CXSV1 STORAGE FOR CXSV KNOWLEDGE COVARIANCE

CXU1 STORAGE FOR CXU KNOWLEDGE COVARIANCE

CXV1 STORAGE FOR CXV KNOWLEDGE COVARIANCE

CXXS1 STORAGE FOR CXXS KNOWLEDGE COVARIANCE

DUM1 INTERMEDIATE VARIABLE

DUM VECTOR SUM OF UPDATE AND STATISTICAL VELOCITY CORRECTIONS

EGM MAXIMUM EIGENVALUE OF S MATRIX

EGVCT ARRAY OF EIGENVECTORS

EGVL ARRAY OF EIGENVALUES

EXEC EXECUTION ERROR COVARIANCE MATRIX

EXV EXPECTED VALUE OF VELOCITY CORRECTION

GA GUIDANCE MATRIX

GAP INTERMEDIATE ARRAY EQUAL TO GA TIMES P

ICODE INTERNAL CONTROL FLAG

ICODE2 INTERNAL CONTROL FLAG

IGP MIDCOURSE GUIDANCE POLICY CODE
**EXECUTION ERROR CODE**

**ISPHC**
TEMPORARY STORAGE FOR ISPH

**MAP**
INDEX OF MAXIMUM EIGENVALUE OF S

**OUT**
SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET IN PLANETO-CENTRIC EQUATORIAL COORDINATES

**PS1**
STORAGE FOR PS KNOWLEDGE COVARIANCE

**P1**
STORAGE FOR P KNOWLEDGE COVARIANCE

**RF**
NOMINAL TRAJECTORY STATE AT GUIDANCE EVENT

**RHO**
MAGNITUDE OF STATISTICAL DELTA-V

**ROW**
INTERMEDIATE VECTOR

**SDV**
STANDARD DEVIATION OF MAGNITUDE OF STATISTICAL DELTA-V

**SOP**
INTERMEDIATE VECTOR

**TRS**
TRACE OF S MATRIX

**U**
INTERMEDIATE VARIABLE

**VEIG**
MATRIX TO BE DIAGONALIZED

**Z**
INTERMEDIATE ARRAY

<table>
<thead>
<tr>
<th>COMMON COMPUTED/USED</th>
<th>CXSUG</th>
<th>CXSU</th>
<th>CXSVG</th>
<th>CXSV</th>
<th>CXUG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CXU</td>
<td>CXVG</td>
<td>CXV</td>
<td>CXS6</td>
<td>CXS</td>
</tr>
<tr>
<td>ISPH</td>
<td>NGE</td>
<td>PG</td>
<td>PSG</td>
<td>PS</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>TG</td>
<td>XG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMON COMPUTED</th>
<th>DELTM</th>
<th>TRTM1</th>
<th>XI</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>COMMON USED</th>
<th>FOP</th>
<th>FOV</th>
<th>ICDQ3</th>
<th>ICOT3</th>
<th>IEIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>INYPI</td>
<td>ISTMC</td>
<td>NDIM1</td>
<td>NDIM2</td>
<td>NDIM3</td>
<td></td>
</tr>
<tr>
<td>ONE</td>
<td>Q</td>
<td>SIGALP</td>
<td>SIGBET</td>
<td>SIGPRO</td>
<td></td>
</tr>
<tr>
<td>SIGRES</td>
<td>TWO</td>
<td>U0</td>
<td>V0</td>
<td>XF</td>
<td></td>
</tr>
<tr>
<td>ZERO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

231
GUIDM Analysis

Subroutine GUIDM is the executive guidance subroutine in the error analysis program. In addition to controlling the computational flow for all types of guidance events, GUIDM also performs many of the required guidance computations itself.

Before considering each type of guidance event, the treatment of a general guidance event will be discussed. Let \( t_j \) be the time at which the guidance event occurs. Before any guidance event can be executed, the targeted nominal state \( \bar{x}_j \), knowledge covariance \( P_x^k \), and control covariance \( P_c^k \) must all be available, where \( (\cdot)^{-} \) indicates values immediately before the event. The first two quantities are available prior to entering GUIDM. However, GUIDM controls the preservation of the control covariance over the interval \([t_{j-1}, t_j]\), where \( t_j \) denotes the time of the previous guidance event.

The next step in the treatment of a general guidance event is concerned with the computation of the commanded velocity correction and the execution error covariance. In the error analysis program a non-statistical velocity correction is computed whenever the nominal target conditions are changed; otherwise, only a statistical velocity correction can be computed. The commanded velocity correction \( \Delta \bar{v}_j \) is then used to compute the execution error covariance matrix \( \bar{Q}_j \). A summary of the execution error model and the equations used to compute \( \bar{Q}_j \) can be found in the subroutine QCMP analysis section.

The last step is concerned with the updating of required quantities prior to returning to the basic cycle. An assumption underlying the modeled guidance process is that the targeted nominal is always updated by the commanded velocity correction. In the error analysis program only the non-statistical component is used to perform the state update and is indicated by the variable \( \Delta \bar{v}_{UP_j} \). Thus, the targeted nominal state immediately following the guidance event is given by

\[
\bar{x}_j^+ = \bar{x}_j^- + \begin{bmatrix} -0.5 \\ \Delta \bar{v}_{UP_j} \end{bmatrix}.
\]
The knowledge covariance is updated using the equation

\[ P_{K_j}^+ = P_{K_j}^- + \begin{bmatrix} 0 & 1 \\ -1 & Q_j \end{bmatrix} \]

if an impulsive thrust model is assumed. If the thrust is modeled as a series of impulses, then an effective execution error covariance \( Q_{eff} \) is computed and the knowledge covariance is updated using the equation

\[ P_{K_j}^+ = P_{K_j}^- + \tilde{Q}_{eff} \cdot \]

In either case the control covariance is updated simply by setting

\[ P_{c_j}^+ = P_{c_j}^- \cdot \]

This equation is a direct consequence of the assumption that the targeted nominal state is always updated at a guidance event.

A "compute only" option is available in GUIDM in which all of the \(( \cdot \cdot \cdot )^+\) quantities will still be computed and printed. However, the state and all covariances are then reset to their former \(( \cdot \cdot \cdot )^-\) values prior to returning to the basic cycle.

Each specific type of guidance event involves the computation of other quantities not discussed above. These will be covered in the following discussion of specific guidance events.

1. Midcourse and Biased Apoint Guidance

Linear midcourse guidance policies have form

\[ \Delta V_{N_j} = \Gamma_j \delta X_j \]

where the subscript \( N \) indicates that this is the velocity correction required to null out deviations from the nominal target state. This notation is required to differentiate between this type of velocity correction and velocity corrections required to achieve an altered target.
state. Linear midcourse guidance policies are discussed in more detail in the subroutine GUID analysis section.

Subroutine GUID calls GUID to compute the guidance matrix, $\Gamma_j$, and the target condition covariance immediately prior to the guidance event, $W_j$, and then uses $\Gamma_j$ to compute the velocity correction covariance $S_j$, which is defined as

$$S_j = E \left[ \Delta \hat{V}_{N_j} \Delta \hat{V}_{N_j}^T \right],$$

and is given by the equation

$$S_j = \Gamma_j \left( \sigma_{c_j}^{-2} - \nu_{k_j}^{-2} \right) \Gamma_j^T.$$

This equation assumes that an optimal estimation algorithm is employed in the navigation process, since the derivation of this equation requires the orthogonality of the estimate and the estimation error.

In the error analysis program $\Delta \hat{V}_{N_j}$ is never available since no estimates $\delta X_j$ are ever generated. Only the ensemble statistics of $\delta \hat{X}_j$ are available, which means only a statistical or effective velocity correction $E[\Delta \hat{V}_{N_j}]$ can be computed. In the STEAP error analysis program this effective velocity correction is assumed to have form

$$E[\Delta \hat{V}_{N_j}] = \rho_j \frac{\sigma_1}{|\alpha_1|}.$$

The magnitude $\rho_j$ is given by the Hoffman-Young approximation

$$\rho_j = \sqrt{\frac{2A}{\pi}} \left( 1 + \frac{B(\pi-2)}{A^2\sqrt{5.4}} \right)$$
where

\[
A = \text{trace } S_j = \lambda_1 + \lambda_2 + \lambda_3,
\]

\[
B = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3,
\]

and \(\lambda_1, \lambda_2, \lambda_3\) are the eigenvalues of \(S_j\). The direction of the effective velocity correction is assumed to coincide with the eigenvector corresponding to the maximum eigenvalue of \(S_j\). This eigenvector is denoted by \(\sigma_j\). An alternate model assumes the direction coincides with the vector \((\lambda_1, \lambda_2, \lambda_3)\).

If planetary quarantine constraints must be satisfied at a midcourse correction, GUIDM calls BLAIM to compute the new aimpoint \(\mu_j\) and the (non-statistical) bias velocity correction \(\Delta V_{Bj}\). All computations in BLAIM are based on linear guidance theory. However, an option is available in GUIDM to recomputes \(\Delta V_{Bj}\), but not \(\mu_j\), using nonlinear techniques. This option is recommended

if a biased aimpoint guidance event occurs at \(t_j = injection\ time\). It should also be noted that \(\beta_j\) is set to zero if \(t_j = injection\ time\ since\ it\ is\ assumed\ that\ the\ injection\ covariance\ does\ not\ change\ for\ small\ changes\ in\ injection\ velocity.\n
After the updated control covariance \(\Sigma_{ej}^+\) has been computed, the target condition covariance matrix \(\Omega_j^+\) following the guidance correction is computed using the equation

\[
\Omega_j^+ = \eta_j \Sigma_{ej}^+ \eta_j^T
\]

where variation matrix \(\eta_j\) has been previously computed in subroutine GULD.

2. Re-targeting

In the error analysis (and simulation) program a re-targeting event is defined to be the computation of a velocity correction \(\Delta \theta_{RT}\) required to achieve a new set of target conditions using nonlinear techniques. Since the original targeted nominal will be used as the zero-th iterate in the re-targeting process, the new target conditions must be close enough to the original nominal target condition to ensure a convergent process.
It should be noted that after a re-targeting event the new target conditions are henceforth treated as the nominal target conditions.

3. Orbital insertion

An orbital insertion event is divided into a decision event and an execution event. At a decision event the orbital insertion velocity correction $\Delta \dot{V}_{PI}$ and the time interval $\Delta t$ separating decision and execution are computed based on the targeted nominal state at $t_j$. The relevant equations can be found in the subroutine COPINS analysis section for co-planar orbital insertion; in NOPSINS, for non-planar orbital insertion. Before returning to the basic cycle, GUIDM schedules the orbital insertion execution event to occur at $t_j + \Delta t$ and re-orders the necessary event arrays accordingly.

At an orbital insertion execution event the targeted nominal state is updated using the previously computed $\Delta \dot{V}_{PI}$. In addition, the planeto-centric equatorial components of $\Delta \dot{V}_{PI}$ and the nominal spacecraft cartesian and orbital element state following the insertion maneuver are computed.

4. Externally-supplied velocity correction

At this type of guidance event the targeted nominal state is simply updated using the externally-supplied velocity correction $\Delta \dot{V}_{EX}$.

Because of the complexity of the GUIDM flow chart, a simplified flow chart depicting the main elements of the GUIDM structure precedes the complete GUIDM flow chart.
Simplified GUIDM Flow Chart

ENTER

Compute \( \mathbf{r}_{i,j} \).

-orbital insertion execution

Compute \( \Delta \hat{\theta}_{\text{UP},j} = \Delta \hat{\varphi}_1 \)

Define insertion execution event.

-orbital insertion decision

Compute \( r_j, \hat{w}_j, \Theta_j, \) and \( \Theta'_j \).

midcourse

Compute \( \Delta \hat{\theta}_{\text{RT}} \) to achieve new target conditions.

-re-targeting

Compute \( \Delta \theta_{\text{UP},j} = \Delta \theta_{\text{RT}} \)

externally-supplied \( \Delta V \)

Compute \( \Delta \theta_{\text{EX}} \) stored in DELV array.

Compute \( \hat{q}_j \).

Compute biased aimpoint, \( \Delta \hat{\theta}_{\text{B},j} \), and \( t_j \).

bias aimpoint?

-YES

Update nominal target conditions.

-NO

Compute nonlinear \( \Delta \theta_{\text{B},j} \).

Compute \( \Delta \theta_{\text{UP},j} = \Delta \theta_{\text{B},j} \)

Compute \( \hat{q}_j \).

nominal \( \Delta \theta_{\text{B},j} ? \)

-YES

Compute nonlinear \( \Delta \theta_{\text{B},j} \).

-compute \( \hat{q}_j \).

-NO

Compute \( \Delta \theta_{\text{B},j} \).

Compute \( \Delta \theta_{\text{UP},j} = \Delta \theta_{\text{B},j} \)

Compute \( \hat{q}_j \).

900
Define guidance event index II.

Store impulsive $\Delta V$ execution error variances in the DUMMYQ array.

Save ISP4 and set ICODE2 = 1.

Store all knowledge covariance matrix partitions in the P1, CXXS1, CXU1, CXV1, PS1, CXSU1, and CXSV1 arrays.

Store all control covariance matrix partitions in the P, CXXS, CXU, CXV, PS, CXSU, and CXSV arrays.

Call PSIM and STMPR to compute and write out the state transition matrix partitions over the interval $[t_{j-1}, t_j]$.

Call DYNØ to compute the dynamic noise covariance matrix. Write out.

Call NAVM to compute the control covariance matrix partitions over $[t_{j-1}, t_j]$.

A
Call CORREL to compute and write out control correlation matrix partitions and standard deviations.

Store position/velocity covariance matrix partition (P1 array) in the PP array for use in subroutine P1COMM.

Set ICODE2 = 2.

Store position partition of the P array in the Z and VEIG arrays.

Call EIGHY to compute and write out the eigenvalues, eigenvectors and hyperellipsoid of the position partition of the P array.

Store velocity partition of the P array in the Z and VEIG arrays.

Call EIGHY to compute and write out the eigenvalues, eigenvectors, and hyperellipsoid of the velocity partition of the P array.
Define guidance policy code IGP and execution error code IQP.

Call GUID to compute $\gamma_j$, $\eta_j$, $W_j$, and certain quantities required for biased airmass guidance.

ISPH = 0?

Return to state vector, time, and knowledge covariance matrix partitions to their $t_j$ values. Restore ISPH.

Compute velocity correction covariance matrix $S_j$. Write out correlation matrix and standard deviations. Compute and write out eigenvalues and eigenvectors.

IGP = 2?

Compute and write out hyperellipsoid of $S_j$.
I
E:

I
e
ellipsoid.

\[ YES \]

ICUID(3,II) = 0

Use simplified method for computing \[ E[\Delta \hat{V}_{N_j}] \]. Write out. Compute and write out \( \hat{E}_j \) and its eigenvalues, eigenvectors, and hyper-ellipsoid.

900

862

IGUID(3,II) = 0

NO

Store \[ E[\Delta \hat{V}_{N_j}] \] in DVN array for use in subroutine BIADM.

Call BIADM to perform biased aimpoint guidance event. Return aimpoint \( \mu_j \), bias velocity correction \( \Delta \hat{V}_{upj} \), and execution error covariance matrix \( \hat{Q}_j \).

\[ 3 \]

9761

858

YES

"QP = 0?"

Use Hoffman-Young formula to compute \( E[\Delta \hat{V}_{N_j}] \).

Write out.

900

\[ t_j = \text{injection time?} \]

YES

Set \( Q_j = 0 \) and \( \Delta \hat{V}_{upj} = 0 \).

\[ 900 \]

\[ 9761 \]

\[ \text{Set IREX} = 1 \]
Yes

IGUID(3,XX) = 1?

NO

Set IX = 1 and JX = II. Set XIN to state at t_d.

Write: Bias velocity correction will be recomputed using nonlinear guidance.

\[ \text{IF} IGUID(1,XX) = ? \]

FPA

Compute Julian date corresponding to t_CA and set IG0 = 1.

\[ \text{IG0 = ?} \]

Define target variables XSTAR, LKSTAR, LNPAR, and TGT3 for nonlinear 2VBP guidance.

\[ \text{IG0 = ?} \]

Define target variables XSTAR, LKSTAR, LNPAR, and TGT3 for nonlinear 3VBP guidance.

\[ \text{IG0 = ?} \]

Define target variables XSTAR, LKSTAR, LNPAR, and TGT3 for nonlinear PTA guidance.
Call NONLIN to compute nonlinear bias velocity correction. Return XDELV.

KWIT = 0?

Set \( \Delta \vec{v}_{UP} \) to XDELV.

\( \text{Compout magnitude. Write out } \Delta \vec{v}_{UP} \text{ and its magnitude.} \)

Write: Nonlinear guidance failed. Linear guidance will be employed.

\( t_j = \text{injection time?} \)

\( \text{Compute } \vec{q}_j \text{ using } \Delta \vec{v}_{UP} + \text{vec} \left[ \Delta \vec{v}_{E_j} \right]. \)

Set IRET = 2.

\( \text{Write out correlation matrix and standard deviations associated with } \vec{q}_j. \)

\( \text{IRET = ?} \)

Set \( \Delta \vec{v}_{UP} \) to zero.
Call CORREL to write out control and knowledge correlation matrix partitions and standard deviations just after the guidance correction.
Compute target condition covariance matrix $W_j$ following the guidance correction. Write out the associated correlation matrix and standard deviations.

Compute and write out eigenvalues, eigenvectors, and hyperellipsoid associated with $W_j$.

Store PSAVE array in the PI array so that the PI array contains the knowledge covariance just before the guidance event.

Update and write out targeted nominal immediately following the guidance correction.

Update state vector, times, and control and knowledge covariance matrix partitions in preparation for next cycle. Restore ISPH.

RETURN
Set $\Delta \hat{V}_{UP_j}$ to the pre-specified DELV array. Write out $\Delta \hat{V}_{UP_j}$ and its magnitude.

Define Julian date at orbital insertion and set $\text{ICQ} = 2$.

Write: Error in IGUID array.

Compute and write out nominal spacecraft state relative to the target planet immediately following orbital insertion.

Call PECEQ to compute the transformation from planeto-centric ecliptic to planeto-centric equatorial coordinates.

Compute and write out planeto-centric equatorial coordinates of $\Delta \hat{V}_{UP_j}$ and the relative spacecraft state.

Compute target planet gravitational constant. Call CAREL to compute the orbital elements of the spacecraft orbit. Write out elements.

Compute $\Omega_j$ using $\Delta \hat{V}_{UP_j}$.

Set IRET = 2.
Set IX = 2 and JX = II. Set XIN to state at $t_j$.

Define INTEK array for orbital insertion.

Call NONLIN to compute orbital insertion $\Delta V$ and time. Return XDELV and TCT3.

Set $\Delta V_{UP_j}$ to XDELV and DT$\Phi$ to TCT3.

KEXIT = 1 ?

YES

IGUID(5,II) $\neq$ 2 ?

YES

Compute $\Delta V_{UP_j}$. Set IRET = 3.

Define orbital insertion execution event. Write out time at which event will be executed.

Re-order event arrays as required by previous definition of orbital insertion event.

NO

Write: Orbital insertion failed.

IGUID(5,II) $\neq$ 2 ?

YES

EXIT

NO

105
Set IX = 1 and JX = II. Set XIN to state at t_j.

Call NONLINEAR to perform re-targeting event (target variables are defined in namelist). Return XDELV.

Set Δ\hat{v}_{UP_j} to XDELV.

KWIT = 0 ?

NO

IGUID(5,II) = 2 ?

YES

Write: Re-targeting failed.

NO

Update nominal target conditions in TNXMB and TNXMC arrays. Write out.

Compute \tilde{Q}_j using Δ\hat{v}_{UP_j}. Write out Δ\hat{v}_{UP_j} and its magnitude. Set IRET = 2.

YES

EXIT

105

9761

NO
Set XXIN to state at \( t \).
Set DELPX to \( \Delta \hat{V}_{UP_j} + \Delta \hat{E} \left[ \Delta \hat{S}_{N_j} \right] \) ".

Define Julian date at \( t_j \).

Set INPX = 2 and store impulse series execution error variances in the DUMMYQ array.

Call EXCUT to compute the effective execution error covariance matrix corresponding to the impulse series.
Return QK.

Restore single impulse execution error variances into the DUMMYQ array.

Store P1 array in the PSAVE array.
Update knowledge covariance by adding effective \( Q_j \) to the P1 array.
Program GUdS

Purpose: Compute guidance matrix, variation matrix, and target condition covariance matrix at a midcourse guidance event in the simulation program.

Subroutines supported: GUdSIM

Subroutines required: EPHEM HYELS JACOBI MATIN NTMS ORB PARTL PSIM STMPR VARSIM

Local symbols:

A  Two-variable B-plane guidance sub-matrix A
B  Two-variable B-plane guidance sub-matrix B
BDR1  Value of B dot R returned from PARTL (not used)
BDT1  Value of B dot T returned from PARTL (not used)
B1  Magnitude of B vector returned from PARTL (not used)
DUM1  Array of eigenvectors
DUM  Intermediate array
EGVCT  Array of eigenvectors
EGVL  Array of eigenvalues
ICLS  Intermediate storage for ICL
ICS  Intermediate storage for ICL2
IPR  Intermediate storage for IPRINT
ISPS  Intermediate storage for ISP2
PBR  Partial of B dot R with respect to state vector
PBT  Partial of B dot T with respect to state vector
PHI1  Intermediate array
PHI2  Intermediate array
PHI3  Intermediate array
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RII</td>
<td>MOST RECENT NOMINAL SPACECRAFT STATE AT GUIDANCE EVENT</td>
</tr>
<tr>
<td>RI</td>
<td>TARGETED NOMINAL SPACECRAFT STATE AT GUIDANCE EVENT</td>
</tr>
<tr>
<td>RMCA</td>
<td>SPACECRAFT DISTANCE FROM TARGET PLANET AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>RMSI</td>
<td>SPACECRAFT DISTANCE FROM TARGET PLANET AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>ROW</td>
<td>TARGET CONDITION CORRELATION MATRIX</td>
</tr>
<tr>
<td>RTPS</td>
<td>INERTIAL SPACECRAFT STATE AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>SQP</td>
<td>TARGET CONDITION STANDARD DEVIATIONS</td>
</tr>
<tr>
<td>TCA</td>
<td>TRAJECTORY TIME AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>TSI</td>
<td>TRAJECTORY TIME AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>VMCA</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT CLOSEST APPROACH</td>
</tr>
<tr>
<td>VMSI</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE</td>
</tr>
<tr>
<td>XCA</td>
<td>SPACECRAFT VELOCITY RELATIVE TO TARGET PLANET AT SPHERE OF INFLUENCE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMON COMPUTED/USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM ICL2 IPRINT ISPH ISP2 NO XP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMON COMPUTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTM TRTM1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMON USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALNGTH BDR BDT B DATEJ DC DSI FNTM FOV F IBARY IHYP1 ISO11 ISTMC NB NGE NQE NTMC NTP ONE PHI P RC RSI TM VSI ZERO</td>
</tr>
</tbody>
</table>
GUIG Analysis

Subroutine GUIG is called at a midcourse guidance event at \( t_j \) in the simulation mode to compute three primary quantities for the selected midcourse guidance policy. These three quantities are the variation matrix \( \eta_j \), the target condition covariance matrix prior to the velocity correction \( W_j \), and the guidance matrix \( \Gamma_j \). Three midcourse guidance policies are available: fixed-time-of-arrival (FTA), two-variable B-plane (2VBP), and three-variable B-plane (3VBP). All are linear impulsive guidance policies having form

\[
\Delta \hat{V}_j = \Gamma_j \delta \hat{X}_j
\]

where \( \Delta \hat{V}_j \) is the commanded velocity correction, and \( \delta \hat{X}_j \) is the estimate of the spacecraft position/velocity deviation from the targeted nominal. The relevant equations for each guidance policy will be summarized below.

The variation matrix \( \eta_j \) for FTA guidance relates deviations in spacecraft state at \( t_j \) to position deviations at time of closest approach \( t_{CA} \), and is given by

\[
\eta_j = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}
\]

where \( \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix} \) is the upper half of the state transition matrix \( \Phi(t_{CA}, t_j) \).

The guidance matrix for FTA guidance is given by

\[
\Gamma_j = \begin{bmatrix} -\eta_1^{-1} & 1 & -1 \end{bmatrix}
\]

The variation matrix for 3VBP guidance relates deviations in spacecraft state at \( t_j \) to deviations in B-T, B-R, and \( t_{SI} \), where \( t_{SI} \) is the time at which the sphere of influence is pierced. Unlike the variation matrix for FTA guidance, which can be computed analytically or by numerical differencing, the 3VBP variation matrix must always be computed using numerical differencing since no good analytical formulas are available which relate deviations in spacecraft state at \( t_j \) to deviations in \( t_{SI} \). If the variation matrix is written as

\[
\eta_j = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}
\]

then the guidance matrix for 3VBP guidance is given by

\[
\Gamma_j = \begin{bmatrix} -\eta_2^{-1} & 1 & -1 \end{bmatrix}
\]
The variation matrix for 2VBP guidance relates deviations in spacecraft state at \( t_j \) to deviations in \( B \cdot T \) and \( B \cdot R \) and is given by

\[
\eta_j = M \Phi (t_{SI}, t_j)
\]

where \( M \) is an analytically computed matrix relating \( B \cdot T \) and \( B \cdot R \) deviations to spacecraft state deviations at \( t_{SI} \), and \( \Phi (t_{SI}, t_j) \) is the state transition matrix over \([t_j, t_{SI}]\). If \( \eta_j \) is written as

\[
\eta_j = \begin{bmatrix} A & B \end{bmatrix}
\]

then the guidance matrix for 2VBP guidance is given by

\[
\Gamma = \begin{bmatrix} -B (BB)^{-1} T & -B (BB)^{-1} T \end{bmatrix}
\]

All state transition matrices and, hence, all variation matrices used by the above three guidance policies are referenced to the most recent nominal trajectory for improved numerical accuracy.

Once the variation matrix \( \eta_j \) is available for any of the above guidance policies, the target condition covariance matrix can be computed using

\[
W_j = \eta_j P_{c_j}^{-1} \eta_j^T
\]

where \( P_{c_j}^{-1} \) is the control covariance matrix immediately prior to the guidance event.
GUIS Flow Chart

ENTER

ICP = ?

1 ~ PTA

Call NIMS to integrate targeted nominal to final time $t_f$ and to define closest approach conditions.

YES

Was SOI encountered over $[c_j, t_f]$?

NO

Call PARTL to compute partials of $B^T$ and $B^R$ with respect to state at $t_{SI}$. Compute $M$ matrix.

Write: SOI not encountered on targeted nominal. Returning to basic cycle.

RETURN

Compute time $t_{CA}$ and position $x_{CA}$ and velocity magnitudes at closest approach.

Write out closest approach conditions for targeted nominal. Write out $M$ matrix.

Define variables ATRANS, VINF, and TEMP required for biased aimpoint guidance.
recent nominal to final time $t_f$ and to define closest approach conditions.

Was SOI encountered over $[t_j, t_r]$?

- Write: SOI not encountered on most recent nominal. Returning to basic cycle.
- Call NMS to integrate most recent nominal to final time $t_f$ and to define closest approach conditions.
- Set most recent nominal to targeted nominal at $t_f$.

Write out closest approach conditions on most recent nominal.

Compute inertial ecliptic position and velocity components of spacecraft at closest approach on most recent nominal.

Call PSIM to compute state transition matrix partitions over $[t_j, t_{CA}]$ on most recent nominal. Call STMFR to write out these partitions. Set ISPH = 1.

Compute variation matrix $\eta_j$ for PTA guidance policy.
Write out variation matrix $\eta_j$.

Compute target condition covariance matrix $W_j$ and write out associated correlation matrix and standard deviations.

Call JACOBI and HYELS to compute eigenvalues, eigenvectors, and hyperellipsoid of $W_j$. Write out.

Call MATINV to invert the second half of the variation matrix $\eta_j$.

$t_j$ = injection time?

YES

NO

Compute guidance matrix $\Gamma$.

Set $\Gamma = 0$.

NO

YES

RETURN

$|\Gamma| = 3$?

Write out guidance matrix $\Gamma$ for FTA guidance policy.

Write out guidance matrix $\Gamma$ for JVFP guidance policy.

RETURN
200

Has S0I been previously encountered on targeted nominal?

YES

Write: S0I has been previously encountered on targeted nominal. B-plane guidance policies undefined. Returning to basic cycle.

NO

Call NTMS to integrate targeted nominal to final time t_f and to define sphere of influence conditions.

YES

Was S0I encountered over [t_i, t_f]?

NO

Write: S0I not encountered on targeted nominal. Returning to basic cycle.

RETURN

ISPH = 0

RETURN

Call PAKIL to compute partials of B-T and B-R with respect to state at t_{SI}. Compute M matrix.

Compute time t_{SI} and position and velocity magnitudes at sphere of influence.

Write out S0I conditions for targeted nominal. Write out M matrix.

Define variables TPMR and VINF required for biased aimpoint guidance.

C
GUlS-1

C

NOM=BJ=0?  YES

Call NTMS to integrate most recent nominal to final time \( t_f \) and to define SØI conditions.

Set most recent nominal equal to targeted nominal at \( t_f \).

NO

Was SØI encountered over \([ t_0, t_f ]\)?

YES

Write: SØI not encountered on most recent nominal. Returning to basic cycle.

RETURN

NO

Write out SØI conditions on most recent nominal.

IJP=3?

YES

Call VARSIM to compute variation matrix \( \eta_j \) for 3VAR guidance policy.

NO

ISPh=Q?

YES

RETURN
Call PSM to compute state transition matrix partitions over \([t_j, t_{SI}]\) on most recent nominal. Call STMPR to write out these partitions. Set ISPH = 1.

Call PARTL to compute partials of B-T and B-R with respect to most recent nominal state at \(t\). Write out these partials.

Compute variation matrix \(\eta_j\) for 2VBP guidance policy. Write out.

Compute target condition covariance matrix \(W_j\) and write out associated correlation matrix and standard deviations.

Call JAC@BI and HYESL to compute eigenvalues, eigenvectors, and hyperellipsoid of \(W_j\). Write out.

Call MATIN to invert \(BB^T\).

If \(t_j = \) injection time? NO YES

Compute guidance matrix \(\Gamma\) for 2VBP guidance policy and write out. Set \(\Gamma = 0\).

RETURN
PROGRAM GUISIM

PURPOSE CONTROL EXECUTION OF GUIDANCE EVENT IN THE SIMULATION PROGRAM

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL DYNOS GUIS HYELS JACOBI NAVM PSIM SIMPR

LOCAL SYMBOLS: ADA VARIATION MATRIX
AK1 ACTUAL RESOLUTION ERROR
AL1 ACTUAL ERROR IN POINTING ANGLE ALPHA
BT1 ACTUAL ERROR IN POINTING ANGLE BETA
CXSU1 STORAGE FOR CXSU KNOWLEDGE COVARIANCE
CXSU1 STORAGE FOR CXSV KNOWLEDGE COVARIANCE
CXSU1 STORAGE FOR CXU KNOWLEDGE COVARIANCE
CXSU1 STORAGE FOR CXV KNOWLEDGE COVARIANCE
CXSU1 STORAGE FOR CXXS KNOWLEDGE COVARIANCE
DELX ESTIMATED STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY
DUM1 INTERMEDIATE VARIABLE
DUM2 ARRAY OF EIGENVECTORS
DVM MagNITUDE OF COMMANDED MIDCOURSE VELOCITY CORRECTION
DVC COMMANDED MIDCOURSE VELOCITY CORRECTION
DVE ERROR IN MIDCOURSE VELOCITY CORRECTION DUE TO NAVIGATION UNCERTAINTY
DV PERFECT MIDCOURSE VELOCITY CORRECTION
DX ACTUAL STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY
EGVCT ARRAY OF EIGENVECTORS
EGVL ARRAY OF EIGENVALUES
EXEC EXECUTION ERROR COVARIANCE MATRIX
EXN  MAGNITUDE OF UPDATE VELOCITY CORRECTION
GAP  INTERMEDIATE ARRAY EQUAL TO GA TIMES P
GA   GUIDANCE MATRIX
ICODE2 INTERNAL CONTROL FLAG
IGP  MIDCOURSE GUIDANCE POLICY CODE
OUT  SPACECRAFT VELOCITY RELATIVE TO TARGET
     PLANET IN PLANETO-CENTRIC EQUATORIAL
     COORDINATES
PS1  STORAGE FOR PS KNOWLEDGE COVARIANCE
P1   STORAGE FOR P KNOWLEDGE COVARIANCE
RF1  MOST RECENT NOMINAL SPACECRAFT STATE AT
     GUIDANCE EVENT
RF   TARGETED NOMINAL SPACECRAFT STATE AT
     GUIDANCE EVENT
ROW  INTERMEDIATE VECTOR
SQP  INTERMEDIATE VECTOR
S1   ACTUAL PROPORTIONALITY ERROR
VEIG MATRIX TO BE DIAGONALIZED
Z    INTERMEDIATE ARRAY

COMMON COMPUTED/USED:
     ADEVX  CXSUG  CXSU  CXSVG  CXSV
     CXUG  CXU  CXVG  CXV  CXXSG
     CXXS  EDEVX  IGODE  NGE  PG
     PSG  PS  P  R11  TG
     XF1  XG

COMMON COMPUTED:
     DELTM  TRTM1  XI1  XI

COMMON USED:
     AALP  ABET  ADEVXS  API0  ARES
     EDEVXS  F0P  F0V  ICDT3  IEIG
     IHYP1  ISPH  ISTMC  NDIM1  NDIM2
     NDIM3  Q  SIGALP  SIGBET  SIGPRO
     SIGRES  TEVN  U0  VO  W
     XF  XSL  ZERO
GUISIM Analysis

Subroutine GUISIM is the executive guidance subroutine in the simulation program. In addition to controlling the computational flow for all types of guidance events, GUISIM also performs many of the required guidance computations itself.

Before considering each type of guidance event, the treatment of a general guidance event will be discussed. Let \( t_j \) be the time at which the guidance event occurs. Before any guidance event can be executed the targeted nominal state \( X_{j}^* \), most recent nominal state \( X_{j}^* \), estimated state deviation \( \delta X_{j}^* \) from most recent nominal, actual state deviation \( \delta X_{j}^* \) from most recent nominal, knowledge covariance \( P_{K_j}^- \), and control covariance \( P_{C_j}^- \) must all be available, where \( (\cdot)^- \) indicates values immediately before the event. Only the control covariance is not available prior to entering GUISIM. The propagation of the control covariance over the interval \( [t_{j-1}, t_j] \), where \( t_{j-1} \) denotes the time of the previous guidance event, is performed within GUISIM.

The next step in the treatment of a general guidance event is concerned with the computation of the commanded velocity correction, execution error covariance, actual execution error, and actual velocity correction. In the simulation program a non-statistical commanded velocity correction can always be computed. This commanded velocity correction \( \Delta V_j \) is used to compute the execution error covariance matrix \( \tilde{Q}_j \) and the actual execution error \( \delta \Delta V_j \). A summary of the execution error model and the equations used to compute \( \tilde{Q}_j \) and \( \delta \Delta V_j \) can be found in the subroutine QCUMP analysis section. The actual velocity correction is then computed using the equation

\[
\Delta V_j = \Delta \tilde{V}_j + \delta \Delta V_j
\]

The last step is concerned with the updating of required quantities prior to returning to the basic cycle. An assumption underlying the modeled guidance process is that the targeted nominal is always updated by the commanded velocity correction. In the simulation program the update velocity correction \( \Delta \tilde{V}_{UP,j} \) is always identical to the commanded velocity correction \( \Delta \tilde{V}_j \). This is in contrast to the error analysis program where \( \Delta \tilde{V}_{UP,j} \) is equated with the non-statistical component of \( \Delta \tilde{V}_j \).
most recent and targeted nominal states immediately following the guidance event are updated using the equations

\[ \tilde{x}^+_j = \tilde{x}^-_j + \delta \tilde{x}^-_j + \begin{bmatrix} -0 & - \end{bmatrix} \]

\[ \hat{x}^+_j = \tilde{x}^+_j \]

The actual and estimated state deviations from the most recent nominal are given by

\[ \delta \tilde{x}^+_j = \delta \tilde{x}^-_j - \delta \tilde{x}^-_j + \begin{bmatrix} -0 & - \end{bmatrix} \]

\[ \delta \hat{x}^+_j = 0 \]

The previous 4 equations assume an impulsive thrust model. If, instead, the thrust is modeled as an impulse series, then an effective estimated state \( \hat{x}_{\text{eff}} \) and an effective actual state \( \tilde{x}_{\text{eff}} \) are computed. The equations used to compute these effective states are summarized in the subroutine PULSEX analysis section. The previous update equations are then replaced by the following equations

\[ \tilde{x}^+_j = \hat{x}_{\text{eff}} \]

\[ \tilde{x}^+_j = \hat{x}^+_j \]

\[ \delta \tilde{x}^+_j = \tilde{x}_{\text{eff}} - \tilde{x}_{\text{eff}} \]

\[ \delta \hat{x}^+_j = 0 \]

The knowledge covariance is updated using the equation

\[ P^+_j = F^-_j \begin{bmatrix} -0 & - \\ -0 & Q^-_j \end{bmatrix} \]
if an impulsive thrust model is assumed. If the thrust is modeled as a series of impulses, then an effective execution error covariance \( \hat{Q}_{\text{eff}} \) is computed and the knowledge covariance is updated using the equation

\[
P_k^+ = P_k^- + \hat{Q}_{\text{eff}}
\]

In either case the control covariance is updated simply by setting

\[
P_c^+ = P_k^+
\]

This equation is a direct consequence of the assumption that the targeted nominal is always updated at a guidance event.

A "compute only" option is available in GUISIM in which all of the \((+)\) quantities will still be computed and printed. However, states, deviations, and covariances are then reset to their former \((-)\) values prior to returning to the basic cycle.

Each specific type of guidance event involves the computation of other quantities not discussed above. These will be covered in the following discussion of specific guidance events.

1. Midcourse and biased aimpoint guidance.

Linear midcourse guidance policies have form

\[
\Delta \hat{V}_j = \Gamma_j \delta \hat{x}_j
\]

where the subscript \( N \) indicates that this is the velocity correction required to null out deviations from the nominal target state. This notation is required to differentiate between this type of velocity correction and velocity corrections required to achieve an altered target state. Linear midcourse guidance policies are discussed in more detail in the subroutine GUIS analysis section.

Subroutine GUISIM calls GUIS to compute the guidance matrix, \( \Gamma_j \), and the target condition covariance immediately prior to the guidance event, \( W_j^- \), and then uses \( \Gamma_j \) to compute the velocity correction covariance \( S_j \), which is defined as
\[ S_j = \mathbb{E} \left[ \Delta \hat{X}_{N_j} \Delta \hat{X}_{N_j}^T \right] , \]

and is given by the equation

\[ S_j = \Gamma_j \left( \mathcal{P}_j - \mathcal{P}_j^T \right) \Gamma_j^T \]

This equation assumes that an optimal estimation algorithm is employed in the navigation process, since the derivation of this equation requires the orthogonality of the estimate and the estimation error.

Since state estimates \( \Delta \hat{X}_j \) are generated in the simulation program, an actual \( \Delta \hat{X}_j \) can always be computed. This is in contrast to the error analysis program where only a statistical or effective \( \Delta \hat{X}_j \) can be computed.

The perfect velocity correction \( \Delta v_j \), defined as the velocity correction required to null out actual deviations from the nominal target state, is also computed for midcourse guidance events. Assuming linear guidance theory, the perfect velocity correction is given by

\[ \Delta v_j = \Gamma_j \delta X_j \]

where \( \delta X_j \) is the actual deviation from the targeted nominal. An option is also available in GUISIM for re-computing \( \Delta V_j \) using nonlinear techniques.

However, it should be noted that the nonlinear two-variable B-plane guidance policy, unlike the corresponding linear policy, constrains the \( z \)-component of \( \Delta V_N \) to be zero.

If planetary quarantine constraints must be satisfied at a midcourse correction, GUISIM calls BIAIM to compute the new aimpoint \( \mu_j \) and the bias velocity correction \( \Delta \hat{V}_j \). All computations in BIAIM are based on linear guidance theory. However, an option is available in GUISIM to re-compute the total velocity correction \( \Delta V_j = \Delta \hat{V}_j + \Delta V_N \), but not \( \mu_j \), using nonlinear techniques. This option is recommended if a biased aimpoint guidance event occurs at \( t_j \) = injection time. It should also be noted that \( Q_j \) is set to zero if \( t_j \) = injection time since it is assumed that the injection
covariance does not change for small changes in injection velocity.

After the updated control covariance $P_{c_j}^+$ has been computed, the target condition covariance matrix $W_j^+$ following the guidance correction is computed using the equation

$$W_j^+ = \eta_j P_{c_j}^+ \eta_j^T$$

where variation matrix $\eta_j$ has been previously computed in subroutine GUIS.

2. Re-targeting.

In the simulation (and error analysis) program a re-targeting event is defined to be the computation of a velocity correction $\Delta V_{RT}$ required to achieve a new set of target conditions using nonlinear techniques. Since the state estimate $\tilde{X}_j + \tilde{\delta X}_j$ is used as the zero-th iterate in the re-targeting process, the new target conditions must be close enough to the original nominal target conditions to ensure a convergent process.

It should be noted that after a re-targeting event the new target conditions are henceforth treated as the nominal target conditions.

3. Orbital insertion.

An orbital insertion event is divided into a decision event and an execution event. At a decision event the orbital insertion velocity correction $\Delta V_{p1}$ and the time interval $\Delta t$ is parating decision and execution are computed based on the state estimate $\tilde{X}_j + \tilde{\delta X}_j$. The relevant equations can be found in the subroutine CPFIN analysis section for cplanar orbital insertion; in NPFIN, for non-planar orbital insertion. Before returning to the basic cycle, GUISIM schedules the orbital insertion execution event to occur at $t_j + \Delta t$ and re-orders the necessary event arrays accordingly.

At an orbital insertion execution event the previously computed $\Delta V_{p1}$ is used to update the targeted nominal state. In addition, the planeto-centric equatorial components of $\Delta V_{p1}$ and the actual spacecraft cartesian and orbital element states following the insertion maneuver are computed.
4. Externally-supplied velocity correction.

At this type of guidance event the state estimate $\tilde{X}_j^- + \hat{X}_j^-$ is simply updated using the externally-supplied velocity correction $\Delta \theta_{EX}$.

Because of the complexity of the GUISIM flow chart, a simplified flow chart depicting the main elements of the GUISIM structure precedes the complete GUISIM flow chart.
Simplified CUISIM Flow Chart

ENTER

Compute $P_{c_j}$.

guidance event?

midcourse

externally-supplied $\Delta V$

Computes $\Delta_V_{UPj} = \Delta \hat{V}_{PI}$.

Define insertion execution event.

orbital insertion execution

orbital insertion decision

Compute $T_j$, $W_j$, $S_j$.

$\delta x_j$, $\delta \hat{v}_j$, $\Delta \hat{v}_{Nj}$, $\Delta V_j$

Computes $\Delta \hat{v}_{UPj} = \Delta \hat{v}_{RT}$

bias aimpoint?

YES

NO

$\Delta \hat{v}_{UPj} = \Delta \hat{v}_{Nj}$.

$\Delta \hat{v}_{UPj} = \Delta \hat{v}_{Bj} + \Delta \hat{v}_{Nj}$.

nonlinear $\Delta \hat{v}_{UPj}$?

YES

NO

nonlinear $\Delta \hat{v}_{Nj}$?

YES

NO

Compute nonlinear $\Delta \hat{v}_{j}$.

$\Delta \hat{v}_{UPj} = \Delta \hat{v}_{j}$.

Compute $\hat{t}_j$.

$\Delta \hat{v}_{EX}$ stored in DELV array.
Treat velocity correction at \( t_j \)?

- **YES**
  - Finite burn
  - Thrust model = ?
    - Single impulse
      - Use \( \delta V_j \) to update \( P_{x_j} \) and \( P_{c_j} \)
      - Compute effective \( \delta V_j \) and \( \Delta V_j \).
    - Impulse series?
      - **NO**
        - Compute effective \( X_j \) for impulse series.
      - **YES**
        - Compute effective \( X_j \) for impulse series.
    - Impulse series?
      - **NO**
        - Midcourse?
          - **YES**
            - Compute \( W_{+} \) and target errors.
          - **NO**
            - Orbital insertion?
              - **YES**
                - Compute actual spacecraft state relative to planet and orbital elements following insertion.
              - **NO**
                - Execute velocity correction?
                  - **YES**
                    - Restore state vectors, times, and covariance matrices.
                  - **NO**
                    - Update state vectors, times, and covariance matrices.
    - **NO**
      - Return

- **RETURN**
Enter

Store impulsive ∆V execution error variances in the DUMMYQ array.

Define guidance event index II.

ICOFDE2 = 1

ICOFDE2 = 2

Write out actual dynamic noise and the estimated and actual deviations from the most recent nominal at $t_j$.

Store all knowledge covariance matrix partitions in the Pl, CXXS1, CXU1, CXV1, PS1, CXXSU1, and CXSV1 arrays. Store all control covariance matrix partitions in the P, CXXS, CXU, CXV, PS, CXXSU, and CXSV arrays.

Call PSIM and STMPF to compute and write out the state transition matrix partitions over the interval $[t_{j-1}, t_j]$.

Call DYNOS to compute the dynamic noise covariance matrix. Write out.

Call ... to compute the control covariance matrix partitions over $[t_{j-1}, t_j]$.
Call CORREL to compute and write out control correlation matrix partitions and standard deviations.

Store position/velocity covariance matrix partition (Pl array) in the PP array for use in subroutine POSCOM.

ICDE = 2

Store position partition of the P array in the Z and VEIG arrays.

Call EIGHY to compute and write out the eigenvalues, eigenvectors, and hyperellipsoid of the position partition of the P array.

Store velocity partition of the P array in the Z and VEIG arrays.

Call EIGHY to compute and write out the eigenvalues, eigenvectors, and hyperellipsoid of the velocity partition of the P array.
Write out complete description of guidance event.

IGUID(5,11) = ?

Define guidance policy code IGP.

Call GUW to compute $\mathbf{f}_j$, $\mathbf{r}_j$, $\mathbf{w}_j$, and certain quantities required for biased aimpoint guidance.

YES

$\text{ISP}H = 0$ ?

$\mathbf{t}_j = \text{injection time}$?

Compute velocity correction covariance matrix $\mathbf{S}_j$. Write out correlation matrix and standard deviations. Compute and write out eigenvalues and eigenvectors.

Set $\Delta^0_{N_j} = 0$

NO

Compute and write out hyperellipsoid of $\mathbf{S}_j$.

NO

Restore state vector, time, and knowledge covariance matrix partitions to their $i - j$ values.

YES

IGP = 2 ?

RETURN
Compute and write out actual and estimated position/velocity deviations from the targeted nominal.

Compute and write out the commanded and perfect velocity corrections to null out errors from most recent target conditions. Compute and write out the error in the velocity correction due to navigation uncertainty.

\[
\text{IGUID} (3, 3) = 0 ?
\]

\begin{itemize}
\item NO
  \begin{itemize}
  \item Store \( \Delta \tilde{V}_N \) in DYN array for use in subroutine BLAIM.
  \item Call BLAIM to perform biased aimpoint guidance event. Return aimpoint \( \mu_j \), bias velocity correction \( \Delta \tilde{V}_p \), and execution error covariance matrix \( \tilde{Q}_j \).
  \end{itemize}
\item YES
  \begin{itemize}
  \item \( t_j = \) injection time?
    \begin{itemize}
    \item NO
      \begin{itemize}
      \end{itemize}
    \item YES
      \begin{itemize}
      \item Compute \( \bar{Q}_j \) using \( \Delta \tilde{V}_N \). Set \( \text{IRET} = 1 \).
      \item Set \( \bar{Q}_j \) to zero.
      \end{itemize}
    \end{itemize}
  \end{itemize}
\end{itemize}
Set IX = 1 and JX = II.
Set XIN to estimated state at t_j.

Write: Commanded velocity correction will be recomputed using nonlinear guidance.

1. FTA
   Compute Julian date corresponding to t_CA and set IG0 = 1.

2. 2VBP
   Define target variables XTAR, LXTAR, LNPAR, and TGT3 for nonlinear 2VBP guidance.

3. 3VBP
   Define target variables XTAR, LXTAR, LNPAR, and TGT3 for nonlinear 3VBP guidance.

IGUID(3,II) = 1 ?

NO

977

YES

900

9980

99731

1

9974

Compute position and velocity of target planet at the specified time.

IG0 = ?

Define target variables XTAR, LXTAR, LNPAR, and TGT3 for nonlinear FTA guidance.
Call \texttt{NONLIN} to compute nonlinear commanded velocity correction. Return $X_{DELV}$.

NO

$KWIT = 0$?

YES

Set $\Delta \hat{V}_{UP, j}$ to $X_{DELV}$. Compute magnitude. Write out $\Delta \hat{V}_{UP, j}$ and its magnitude.

Write: Nonlinear guidance failed. Linear guidance will be employed.

YES

$t_j$ - injection time?

NO

Set $\bar{q}_j = 0$.

NO

$t_j$ - injection time?

YES

Set $\bar{q}_j = 0$.

Compute $Q_j$ using $\Delta \hat{V}_{UP, j}$.

Set $IBET = 2$.

Write out correlation matrix and standard deviations associated with $\bar{q}_j$.

Set $\bar{V}_{UP, j}$ to $\bar{V}_{N, j}$.

Compute and write out eigenvalues, eigenvectors, and hyperellipsoid associated with $\bar{q}_j$.

Set $\bar{V}_{UP, j}$ to $\bar{V}_{N, j}$.

YES

$IGUID(2,11) = 0$?

NO

977
GUISIM-15

1a2a3

ICI

ID(5,II) = ?
(Is velocity correction to be treated at t_j?)

4

IGUID(1,II) = 4 or 5?

(NO)

(YES)

IGUID(4,II) = ?
(What kind of thrust model is to be employed?)

2

(impulse series)

3 (finite burn)

1 (single impulse)

Write: Finite burn not available.

Store PI array in PSAVE array. Update knowledge covariance by adding Q_j to the velocity partition of the PI array.

Call CORREL to write out control and knowledge correlation matrix partitions and standard deviations just after the guidance correction.

ICODE2 = 3

80

320

851

909

904

900

1, 2, 3

Write: Finite burn not available.

EXIT
t_j = injection time?

YES

NO

? ?

NO

YES

Use $\Delta V_{UPj}$ to compute the actual execution error $\delta \Delta V_j$. Compute the actual velocity correction $\Delta V_j$. Write out $\delta \Delta V_j$ and $\Delta V_j$.

IGUID(1,II) $\neq$ 4 or 5 ?

YES

Set $\delta \Delta V_j$ and $\Delta V_j$ to zero.

NO

Compute and write out planeto-centric equatorial components of $\Delta V_j$.

IGUID(4,II) $\neq$ 2 ?

YES

XXIN to the actual state -- $t_j$. Set DELPX to $\Delta V_j$.
Compute Julian date at $t_j$. Set
INPX = 1.

Call EXCUTS to compute the effective actual state. Return XXIN. Set XACT to XXIN.

NO

9102
Compute target condition covariance matrix $Y + f$ following the guidance correction. Write out the associated correlation matrix and standard deviations.

Compute and write out eigenvalues, eigenvectors, and hyperellipsoid associated with $W_j$.

Compute and write out actual target error due to navigation uncertainty, actual target error due to execution error, and total target error.

IGUID(5,II) = ?
(Is velocity correction to be executed?)

Store PSAVE array in the P1 array so that the P1 array contains the knowledge covariance just before the guidance event.

2 (NO)

1,3 (YES)
Update most recent nominal, targeted nominal, and estimated and actual state deviations for an impulsively-applied $\Delta V$.

Write out most recent nominal, targeted nominal, and estimated and actual state deviations.

Update state vectors, times, and control and knowledge covariance matrix partitions in preparation for next cycle. Set $N_{GBLJ} = 0$.

RETURN

Set $\Delta V_{UP_j}$ to the pre-specified DELV array. Write out $\Delta V_{UP_j}$ and its magnitude.

Compute $q_j$ using $\Delta V_{UP_j}$.

Set $IRET = 2$.

NO

IGUID(1,II) $\leq$ 3 or IGUID(1,II) = 7

YES

Write: Error in IGUID array.

EXIT
Compute actual spacecraft state relative to target planet immediately following orbital insertion.

IGUID(4, II) ≠ 1 ?

YES

Compute actual spacecraft state relative to target planet immediately following an orbital insertion employing an impulse series thrust model.

NO

IGUID(4, II) ≠ 2 ?

YES

Compute actual spacecraft state relative to target planet immediately following orbital insertion.

Compute target planet gravitational constant. Call CAREL to compute the orbital elements of the spacecraft orbit. Write out elements.
Set \( lx = 2 \) and \( jx = II \). Set \( xin \) to estimated state at \( t_j \).

Define \( LXTAR \) array for orbital insertion.

Call \( NONLIN \) to compute orbital insertion \( \Delta V \) and time. Return \( XDELV \) and \( TGT3 \).

Set \( \Delta V_{UP_j} \) to \( XDELV \) and \( DT\phi \) to \( TGT3 \).

\[ KMIT = 1 \] ?

\( \text{YES} \)\n
Write: Orbital insertion failed.

\( \text{NO} \)

\( IGUID(5,II) \neq 2 \) ?

\( \text{YES} \)

\( \text{NO} \)

Compute \( \dot{q}_j \) using \( \Delta V_{UP_j} \).

Set \( IRET = 3. \)

Define orbital insertion execution event. Write out time at which event will be executed.

Re-order event arrays as required by previous definition of orbital insertion event.

\( 9761 \)

\( 900 \)

\( 320 \) \( \text{EXIT} \)
Set $IX = I$ and $JX = II$. Set $XIN$ to estimated state at $t_j$.

Call NONLIN to perform re-targeting event (target variables are defined in namelist). Return $XI$ ELV.

Set $\Delta \vartheta_{UP_j}$ to $XDE.V$.

$KWIT = 0$ ?

YES

$IGUID(5,II) = 2$ ?

NO

Update nominal target conditions in $TNMB$ and $TNMC$ arrays. Write out.

Compute $\delta_j$ using $\Delta \vartheta_{UP_j}$. Write out $\Delta \vartheta_{UP_j}$ and its magnitude. Set $IRET = 2$.

NO

Write: Re-targeting failed.

$IGUID(5,II) = ?$

YES

NO

320

EXIT

9761
Set XXIN to estimate state at $t_j$. Set DELFX to $\Delta U_{ij}$.

Define Julian date at $t_j$.

Set INFX = 2 and store impulse series execution error variances in the DUMMYQ array.

Call EXECUTS to compute the effective execution error covariance matrix and the effective estimated state corresponding to the impulse series. Return QK and XX".

Restore single impulse execution error variances into the DUMMYQ array.

Store P1 array in the PSAVE array. Update knowledge covariance by adding effective $Q_j$ to the array.

Set XPST to XXIN.
SUBROUTINE HELIO

PURPOSE: TO COMPUTE THE ZERO ITERATE INJECTION STATE FOR INTERPLANETARY TARGETING

CALLING SEQUENCE: CALL HELIO

SUBROUTINES SUPPORTED: ZERIT

SUBROUTINES REQUIRED:  LAUNCH  FLITE  ELCAR  EPHEM  ORB  PECEQ  TIME

LOCAL SYMBOLS:  AMEL  SEMI-MAJOR AXIS OF THE HELIOCENTRIC CONIC
    ARGP  ARGUMENT OF PERIAPSIS OF THE HELIOCENTRIC CONIC IN RADIANS
    ASCND  LONGITUDE OF THE ASCENDING NODE OF THE HELIOCENTRIC CONIC IN RADIANS
    ATP  SEMI-MAJOR AXIS OF TARGET PLANETOCENTRIC CONIC
    AZF  AZIMUTH AT DF ON THE HELIOCENTRIC CONIC IN DEGREES
    AZI  AZIMUTH AT DI ON THE HELIOCENTRIC CONIC IN DEGREES
    B2  SQUARE OF THE B VECTOR MAGNITUDE OF THE TARGET PLANETOCENTRIC CONIC
    CBDR  DESIRED B.R MAGNITUDE AT DF OF THE TARGET PLANETOCENTRIC CONIC
    CBDT  DESIRED B.T MAGNITUDE AT DF OF THE TARGET PLANETOCENTRIC CENTER
    COSASH  COSINE OF ASCND
    COSB  INTERMEDIATE VARIABLE FOR AZI, AZF EQUATION
    COSFS  COSINE OF FS
    COSF  COSINE OF TAI
    COSPSI  COSINE OF PSI
    COSTHE  COSINE OF THETAI
    CRCA  DESIRED RCA MAGNITUDE AT DF OF THE TARGET PLANETOCENTRIC CONIC
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT</td>
<td>Time of Flight (Secs) of Heliocentric Conic</td>
</tr>
<tr>
<td>DF</td>
<td>Final Julian Date of Heliocentric Conic</td>
</tr>
<tr>
<td>DI</td>
<td>Initial Julian Date of Heliocentric Conic</td>
</tr>
<tr>
<td>DSICA</td>
<td>Delta Time (Days) from Sphere-of-Influence to Closest Approach of the Target Planetocentric Conic</td>
</tr>
<tr>
<td>EMEL</td>
<td>Eccentricity of the Heliocentric Conic</td>
</tr>
<tr>
<td>EQEC</td>
<td>Transformation Matrix from Ecliptic to Target Planet Equatorial at DF</td>
</tr>
<tr>
<td>ETP</td>
<td>Eccentricity of the Target Planetocentric Conic</td>
</tr>
<tr>
<td>FF</td>
<td>Intermediate Variable for Computation of DSICA</td>
</tr>
<tr>
<td>FS</td>
<td>True Anomaly of the Target Planetocentric Conic</td>
</tr>
<tr>
<td>IDAT</td>
<td>Calendar Date Corresponding to DF</td>
</tr>
<tr>
<td>IDUM</td>
<td>Dummy Argument for Call to Subroutine FLITE</td>
</tr>
</tbody>
</table>
| ITIM | Indicates Computation of Heliocentric States  
      | - 0, Compute Initial and Final States  
<pre><code>  | - 1, Compute Final State Only |
</code></pre>
<p>| I    | Index |
| J    | Index |
| IDAT | Calendar Date Corresponding to DI |
| OAPO | Apoapsis Radius of the Heliocentric Conic |
| OASM | ASCND Converted to Degrees |
| DECC | Output Eccentricity of the Heliocentric Conic |
| OGA F| Flight Path Angle at DF of the Heliocentric Conic |
| OGA I| Flight Path Angle at DI of the Heliocentric Conic |</p>
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMCA</td>
<td>Central Angle of the Heliocentric Conic</td>
</tr>
<tr>
<td>OINC</td>
<td>Inclination of the Heliocentric Conic</td>
</tr>
<tr>
<td>OLAF</td>
<td>Latitude at DF of Heliocentric Conic</td>
</tr>
<tr>
<td>OLAI</td>
<td>Latitude at DI of Heliocentric Conic</td>
</tr>
<tr>
<td>OLQF</td>
<td>Longitude at DF of Heliocentric Conic</td>
</tr>
<tr>
<td>OLOI</td>
<td>Longitude at DI of Heliocentric Conic</td>
</tr>
<tr>
<td>OPER</td>
<td>Argument of Periapsis of the Heliocentric Conic in Degrees</td>
</tr>
<tr>
<td>ORCA</td>
<td>Periapsis Radius of Heliocentric Conic</td>
</tr>
<tr>
<td>ORF</td>
<td>Magnitude of Heliocentric Position at DF in Output Units</td>
</tr>
<tr>
<td>ORI</td>
<td>Magnitude of Heliocentric Position at DI in Output Units</td>
</tr>
<tr>
<td>OSMA</td>
<td>Semi-Major Axis of the Heliocentric Conic in Output Units</td>
</tr>
<tr>
<td>OTAF</td>
<td>True Anomaly at DF of Heliocentric Conic in Degrees</td>
</tr>
<tr>
<td>OTAI</td>
<td>True Anomaly at DI of Heliocentric Conic in Degrees</td>
</tr>
<tr>
<td>OVF</td>
<td>Magnitude of Heliocentric Velocity at DF in Kilometers</td>
</tr>
<tr>
<td>OVI</td>
<td>Magnitude of Heliocentric Velocity at DI in Kilometers</td>
</tr>
<tr>
<td>OVPF</td>
<td>Velocity of Target Planet at DF</td>
</tr>
<tr>
<td>OVPi</td>
<td>Velocity of Target Planet at DI</td>
</tr>
<tr>
<td>PHEL</td>
<td>Semi-Latus Rectum of Heliocentric Conic</td>
</tr>
<tr>
<td>PLINC</td>
<td>Inclination (in Radians) of Heliocentric Conic</td>
</tr>
<tr>
<td>PSI</td>
<td>Central Angle (in Radians) of Heliocentric Conic</td>
</tr>
<tr>
<td>PIP</td>
<td>Semi-Latus Rectum of Target Planeto-centric Conic at DF, Used to Calculate OSICA</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>RF</td>
<td>Magnitude of target planet position at DF</td>
</tr>
<tr>
<td>RI</td>
<td>Magnitude of target planet position at DI</td>
</tr>
<tr>
<td>RTM</td>
<td>Magnitude of RT vector</td>
</tr>
<tr>
<td>RT</td>
<td>Heliocentric position vector of the final conic corresponding to OTAF</td>
</tr>
<tr>
<td>RZM</td>
<td>Magnitude of the RZ vector</td>
</tr>
<tr>
<td>RZ</td>
<td>Heliocentric position vector of the final conic corresponding to OTAI</td>
</tr>
<tr>
<td>SGN</td>
<td>Internal sign variable used to define the transfer plane orientation</td>
</tr>
<tr>
<td>SINASN</td>
<td>Sine of ASGND</td>
</tr>
<tr>
<td>SINF</td>
<td>Sine of TAI</td>
</tr>
<tr>
<td>SINHNF</td>
<td>Hyperbolic sine of the auxiliary variable F used to calculate DSICA</td>
</tr>
<tr>
<td>SIMPSI</td>
<td>Sine of PSI</td>
</tr>
<tr>
<td>SI</td>
<td>Seconds in calendar date IDAT</td>
</tr>
<tr>
<td>SL</td>
<td>Seconds in calendar date LOAT</td>
</tr>
<tr>
<td>SUNMU</td>
<td>Gravitational constant of Sun in km<strong>3/sec</strong>2</td>
</tr>
<tr>
<td>TAF</td>
<td>OTAF in radians</td>
</tr>
<tr>
<td>TAI</td>
<td>OTAI in radians</td>
</tr>
<tr>
<td>TANF</td>
<td>Tangent of the auxiliary variable F used to calculate DSICA</td>
</tr>
<tr>
<td>TERM</td>
<td>Intermediate variable used to calculate CRCA</td>
</tr>
<tr>
<td>TEST</td>
<td>Intermediate variable used to calculate azimuths and path angles</td>
</tr>
<tr>
<td>TFP</td>
<td>Dummy variable used to call ELCAR</td>
</tr>
<tr>
<td>THEATAI</td>
<td>Intermediate angle used to define ARGP</td>
</tr>
<tr>
<td>TSPh</td>
<td>Sphere-of-influence of target planet in kilometers</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>VF</td>
<td>Velocity of the target planet at DF</td>
</tr>
<tr>
<td>VMAT</td>
<td>Intermediate vector used to define AZI, AZF</td>
</tr>
<tr>
<td>VHP</td>
<td>Hyperbolic excess velocity of the target planetocentric conic at DF</td>
</tr>
<tr>
<td>VHPM</td>
<td>Magnitude of the VHP vector used to calculate OSICA</td>
</tr>
<tr>
<td>VI</td>
<td>Velocity of launch planet at DI</td>
</tr>
<tr>
<td>VMAG</td>
<td>Intermediate variable used to define AZI, AZF</td>
</tr>
<tr>
<td>VTM</td>
<td>Magnitude of VT vector</td>
</tr>
<tr>
<td>VT</td>
<td>Heliocentric velocity vector of the final conic corresponding to OTAF</td>
</tr>
<tr>
<td>VZM</td>
<td>Magnitude of the VZ vector</td>
</tr>
<tr>
<td>VZ</td>
<td>Heliocentric velocity vector of the final conic corresponding to OTAI</td>
</tr>
<tr>
<td>WHAT</td>
<td>Unit vector normal to the transfer plane</td>
</tr>
<tr>
<td>WMAG</td>
<td>Magnitude of the non-unitized WHAT vector</td>
</tr>
<tr>
<td>XF</td>
<td>Position of the target planet at DF</td>
</tr>
<tr>
<td>XI</td>
<td>Position of the launch planet at DI</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED:**
- TMU
- VHPM

**COMMON COMPUTED:**
- DPA
- NO
- RAP
- RIN
- TIN

**COMMON USED:**
- ALNGTH
- DG
- DT
- IZERO
- KTAR
- KUR
- NLP
- NTP
- ONE
- PI
- PMASS
- RAD
- SPHERE
- TAR
- TM
- TWO
- XP
- ZDAT
- ZERO
HELIO Analysis

HELIO computes the zero iterate initial state for interplanetary trajectories. The initial and final states are determined either by an arbitrary position vector or by the location of a specified planet at a specified time according to

\[ \text{IXERO} = \begin{cases} 
1 & \text{planet to planet} \\
2 & \text{planet to arbitrary final point} \\
3 & \text{arbitrary initial point to planet} \\
4 & \text{arbitrary initial point to final point} 
\end{cases} \]

The final time used in locating a planet must correspond to the closest approach (CA) to the planet. Therefore if the target time is read in as a sphere of influence (SOI) time, a modification is required. The heliocentric conic is computed (as described below) using the \( t_{\text{SI}} \) time to determine the final position. The approach asymptote \( \nabla_{\text{HP}} \) corresponding to that trajectory is used with the desired \( r_{\text{CA}} \) to compute the time from SOI to CA. If \( r_{\text{CA}} \) is not a target variable then the target values of \( B-T \) and \( B-R \) are used to estimate the \( r_{\text{CA}} \)

\[
\begin{align*}
    r_{\text{CA}} &= -\frac{\mu}{V_{\text{HP}}} + \frac{1}{2} \sqrt{\left(\frac{2\mu}{V_{\text{HP}}}\right)^2 + 4B^2} \\
\end{align*}
\]

(1)

Then the approximate approach hyperbola is given by

\[
\begin{align*}
    a_h &= \frac{\mu r_{\text{SI}}}{2\mu - V_{\text{HP}}^2 r_{\text{SI}}} \\
    e_h &= 1 - \frac{r_{\text{CA}}}{a} \\
    \beta_h &= \frac{\psi (1 - e_h^2)}{a} \\
    \Delta t_{\text{SI}CA} &= \frac{\mu}{V_{\text{HP}}} (e \sinh \psi - \psi) \\
\end{align*}
\]

(3)

where
\[
\tanh \frac{F}{2} = \sqrt{\frac{e - 1}{e + 1}} \tan \frac{f}{2}
\]

\[
\cos f = \frac{1}{e} \left( \frac{F}{r_{SI}} - 1 \right)
\] (4)

The final time is then given by \( t_f^* = t_{SI}^* + \Delta t_{BICA} \).

The initial and final positions \( \vec{r}_i \) and \( \vec{r}_f \) of the heliocentric conic are either input or computed from the positions of planets determined by ORB and EPHM. The unit normal to the heliocentric orbit plane is

\[
\hat{w} = \frac{\vec{r}_i \times \vec{r}_f}{|\vec{r}_i \times \vec{r}_f|}
\] (5)

The inclination to that plane is

\[
\cos i = \frac{w_z}{w_z}
\] (6)

The ascending node of the plane is given by

\[
\tan \Omega = \frac{\hat{w}_x}{\hat{w}_y}
\] (7)

The central angle of transfer is defined by

\[
\cos \Psi = \frac{\vec{r}_i \cdot \vec{r}_f}{r_i r_f}
\] (8)

The semi-major axis \( a \) and eccentricity \( e \) of the heliocentric conic are computed from Lambert's theorem in subroutine FLITE. The true anomaly \( f_i \) at the initial and final points are computed from

\[
p = a (1 - e^2)
\]

\[
\cos f_i = \frac{p - r_i}{e r_i} \quad \sin f_i = \frac{\cos f_i \cos \Psi - \frac{p - r_i}{e r_i}}{\sin \Psi}
\] (8)
Finally, the argument of periapsis $\omega$ is computed from

$$\cos(\omega + f_1) = \frac{\nabla_i \cdot \vec{u}}{r_1}$$

(10)

where $\vec{u} = (\cos \Omega, \sin \Omega, 0)$.

Therefore the initial or final states $(\nabla_i, \vec{u}_i)$ or $(\nabla_f, \vec{u}_f)$ may now be computed by ELCAR. Let $(\nabla, \vec{v})$ denote either state and let $(\nabla', \vec{v}_p)$ denote the state of the relevant planet. The departure (or approach) asymptote is then given by

$$\vec{v}_{EP} = \vec{v}_f - \vec{v}_p \quad \vec{v}_{HE} = \vec{v}_i - \vec{v}_p$$

(11)

The latitude and longitude of the position vector are

$$\sin \theta = \frac{\nabla}{r} \quad \tan \theta = \frac{\vec{v}}{r}$$

(12)

The path angle $\Gamma$ may be computed from

$$\cos \Gamma = \frac{\vec{v}_{EP}}{r \vec{v}}$$

(13)

The azimuth of the relevant state is

$$\sin \Sigma = \frac{(\nabla \times \vec{v}) \cdot \vec{u}}{|\nabla \times \vec{v}|}$$

(14)

$$\cos \Sigma = \frac{\vec{v} \cdot \vec{u}}{\vec{v} \cos \Gamma}$$

(15)

If the initial state is referenced to a planet, subroutine LAUNCH is called to convert the departure asymptote and launch profile into an injection radius, velocity, and time. Otherwise the initial state is returned as the initial state on the heliocentric conic.

Reference: Space Research Conic Program, Phase III, May 1, 1969, Jet Propulsion Laboratory, Pasadena, California.
HELIO Flow Chart

ENTER

Initialization
ITIM = 0
pol = DT(KUR)
po = DG(KUR)

A

Δt = tf - t1

ITIM = ?

= 1

= 0

= 1, 2

B

IZERO = ?

= 1, 3

= 2, 4

IZERO = ?

= 7

(\bar{x}, \bar{v}) = state of LP at t1
(ORB, EPHM)

\bar{x} = ZDAT(1), ZDAT(2), ZDAT(3)

(\bar{x}, \bar{v}) = state of TP at tf
(ORB, EPHM)

\bar{x} = ZDAT(4), ZDAT(5), ZDAT(6)

Compute heliocentric plane elements
Orbital plane normal \bar{\psi}
Central angle of transfer \bar{\psi}
Orbital plane inclination i

Compute heliocentric semi-major axis and eccentricity from Lambert's theorem (FLITE)
Compute heliocentric conic angles $f_x, f_y, \omega, \omega$

Compute initial velocity $\bar{v}_i$ on heliocentric conic (ELCAR)

- ITIM = 0
- ITIM = 1
- IZERO = 0
- IZERO = 2.4

Compute approach asymptote $\bar{v}_{HP}$

- Is $t_f = t_{SI}$ NO
- Yes

- Is $r_{CA}$ a target parameter?
  - Yes
  - No

- Compute $B^2 = B \cdot T^2 + B \cdot R^2$

- $r_{CA} = -\frac{\mu}{V_{HP}} + \frac{1}{2} \sqrt{\frac{2 \mu}{V_{HP}^2} + 4B^2}$

- Compute nominal time to go from SOI to CA = $\Delta t_{SICA}$
  - from nominal approach conic

Set $DF = BF + \Delta t_{SICA}$
- ITIM = 1
Compute initial state on heliocentric conic (ELCAR) and associated data.

\[ \text{IZERO} = ? \]

-1, 2

-\[ V_{HE} \text{ at launch planet.} \]

Compute launch profile to determine injection state (LAUNCH).

RETURN
SUBROUTINE HYELS

PURPOSE: TO COMPUTE AND PRINT THE TWO-DIMENSIONAL OR THREE-DIMENSIONAL HYPERELLIPSOID OF A SPECIFIED MATRIX.

CALLING SEQUENCE: CALL HYELS(KS,P,N)

ARGUMENTS: KS I SIGMA LEVEL OF THE HYPERELLIPSOID
            P I MATRIX FOR WHICH THE HYPERELLIPSOID IS TO BE COMPUTED
            N I DIMENSION LIMITS OF THE SQUARE MATRIX P

SUBROUTINES SUPPORTED: EIGHY GUJISIM GUISS SETEVM GUIDOM GUID PRED

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS: K2 SQUARE OF SIGMA LEVEL
                P1 INVERSE OF MATRIX P
                P12 TWICE THE VALUE OF (1,2) ELEMENT OF P1
                P13 TWICE THE VALUE OF (1,3) ELEMENT OF P1
                P23 TWICE THE VALUE OF (2,3) ELEMENT OF P1
                V TEMPORARY STORAGE VECTOR FOR ARRAY P

COMMON USED: TWO
HYELS Analysis

Subroutine HYELS computes and writes out hyperellipsoids associated with a 2 or 3 dimensional covariance matrix P.

If P is assumed to be the covariance matrix of an n-dimensional random variable \( \mathbf{x} \) having a gaussian distribution with mean zero, then the probability density function is given by

\[
p = \frac{1}{(2\pi)^{n/2} |P|^{1/2}} \exp \left[ -\frac{1}{2} \mathbf{x}^T P^{-1} \mathbf{x} \right]
\]

Re-writing this equation as

\[
\mathbf{x}^T P^{-1} \mathbf{x} = 2 \ln \left[ \frac{1}{(2\pi)^{n/2} |P|^{1/2}} \right] = k^2
\]

shows that the surface of constant probability density \( p \) is an \( n \)-dimensional ellipsoid, where \( m \) is the rank of \( P \). The constant \( k \) can be shown to correspond to the sigma level of the ellipsoid.

For \( n = 3 \), the above equation has form

\[
a x^2 + b y^2 + c z^2 + d xy + e xz + f yz = k^2
\]

where

\[
a = a_{11}, \quad d = 2a_{12}
\]

\[
b = a_{22}, \quad e = 2a_{13}
\]

\[
c = a_{33}, \quad f = 2a_{23}
\]

and the \( a_{ij} \) are the elements of \( P^{-1} \).

Subroutine HYELS uses this equation to compute a 3-dimensional hyperellipsoid, and sets the appropriate constants to zero to compute a 2-dimensional hyperellipsoid.

SUBROUTINE IMPACT

PURPOSE: TO COMPUTE THE ACTUAL IMPACT PLANE PARAMETERS BDT AND BDR CORRESPONDING TO ANY POINT ON AN INCOMING HYPERBOLA. IT HAS THE OPTION TO CONVERT TARGET VALUES OF INCLINATION XIN AND RADIUS OF CLOSEST RCA INTO EQUIVALENT TARGET VALUES OF BDT AND DBR.

CALLING SEQUENCE: CALL IMPACT(R,V,GMX,T,BDT,BDR,XIN,RCA,DBT,DBR, TCA,KOPT)

ARGUMENTS

R(3) I POSITION VECTOR TO CENTRAL BODY AT EPOCH
V(3) I VELOCITY VECTOR TO CENTRAL BODY AT EPOCH
GMX I GRAVITATIONAL CONSTANT OF CENTRAL BODY
T(3,3) I TRANSFORMATION MATRIX FROM REFERENCE TO INCLINATION SYSTEM
BDT O VALUE OF ACTUAL B.T EVALUATED AT EPOCH
BDR O VALUE OF ACTUAL B.R EVALUATED AT EPOCH
XIN I DESIRED INCLINATION (DEG) (OPTIONAL)
RCA I DESIRED RADIUS OF CLOSEST APPROACH (OPTION)
DBT O TARGET VALUE OF B.T BASED ON XIN, RCA
DBR O TARGET VALUE OF B.R BASED ON XIN, RCA
TCA O TIME FROM PERIAPSIS ON CONIC
KOPT I TARGET VALUE COMPUTATION FLAG
=0 DO NOT COMPUTE TARGET VALUES
=1 COMPUTE TARGET VALUES OF B.T, B.R (MUST READ IN OPTIONAL INPUT)

SUBROUTINES SUPPORTED: TAROPT LUNCON LUNTAR MULTAR VMP

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: AB INTERMEDIATE VARIABLE FOR CALCULATION OF RV,SV,TV SYSTEM
AIN TARGET INCLINATION IN RADIANS. AFTER NORMALIZATION
ANG OUTPUT VARIABLE WHEN DECLINATION CONSTRAINT IS VIOLATED
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUXF</td>
<td>ECCENTRIC ANOMALY (HYPERBOLIC CASE)</td>
</tr>
<tr>
<td>A</td>
<td>SEMI-MAJOR AXIS OF R-V CONIC</td>
</tr>
<tr>
<td>BMAG</td>
<td>MAGNITUDE OF DESIRED B VECTOR</td>
</tr>
<tr>
<td>BV</td>
<td>ACTUAL/DESIRED B VECTOR</td>
</tr>
<tr>
<td>B</td>
<td>MAGNITUDE OF ACTUAL B VECTOR</td>
</tr>
<tr>
<td>COECL</td>
<td>COSINE OF DECL</td>
</tr>
<tr>
<td>COELW</td>
<td>COSINE OF DELW</td>
</tr>
<tr>
<td>CTA</td>
<td>COSINE OF TA</td>
</tr>
<tr>
<td>CW</td>
<td>COSINE OF W</td>
</tr>
<tr>
<td>C1</td>
<td>MAGNITUDE OF VECTOR NORMAL TO ORBITAL PLANE IN INERTIAL SYSTEM</td>
</tr>
<tr>
<td>DEB</td>
<td>DESIRED MAGNITUDE OF DESIRED B VECTOR</td>
</tr>
<tr>
<td>DECL</td>
<td>DECLINATION OF APPROACH ASYMPTOTE IN INCLINATION SYSTEM</td>
</tr>
<tr>
<td>DELW</td>
<td>LONGITUDE OF ASCENDING NODE IN INCLINATION SYSTEM</td>
</tr>
<tr>
<td>E</td>
<td>ECCENTRICITY OF THE R-V CONIC</td>
</tr>
<tr>
<td>II</td>
<td>INCLINATION SIGN INDICATOR. =1, INCLINATION IS POSITIVE =-1, INCLINATION IS NEGATIVE</td>
</tr>
<tr>
<td>IM</td>
<td>INDICATOR FOR DIRECTION OF MOTION OF THE TRAJECTORY =1, MOTION IS POSITIVE =-1, MOTION IS RETROGRADE</td>
</tr>
<tr>
<td>PI</td>
<td>MATHEMATICAL CONSTANT 3.141592653589793</td>
</tr>
<tr>
<td>PV</td>
<td>INTERMEDIATE VECTOR USED TO CALCULATE DESIRED B VECTOR</td>
</tr>
<tr>
<td>P</td>
<td>SEMI-LATUS RECTUM</td>
</tr>
<tr>
<td>QV</td>
<td>INTERMEDIATE VECTOR USED TO CALCULATE ACTUAL B VECTOR</td>
</tr>
<tr>
<td>RAD</td>
<td>DEGREES TO RADIANS CONVERSION CONSTANT</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>RD</td>
<td>TIME DERIVATIVE OF ( R )</td>
</tr>
<tr>
<td>RM</td>
<td>MAGNITUDE OF THE POSITION VECTOR ( R )</td>
</tr>
<tr>
<td>RRD</td>
<td>DOT PRODUCT OF ( R ) AND ( V ) VECTORS</td>
</tr>
<tr>
<td>RV</td>
<td>VECTOR USED TO CALCULATE ACTUAL AND DESIRED ( B ) DOT ( R )</td>
</tr>
<tr>
<td>SDECL</td>
<td>SINE OF ( \text{DECL} )</td>
</tr>
<tr>
<td>SDELW</td>
<td>SINE OF ( \text{DELW} )</td>
</tr>
<tr>
<td>SIMMF</td>
<td>HYPERBOLIC SINE OF ( \text{AUXF} )</td>
</tr>
<tr>
<td>STA</td>
<td>SINE OF ( \text{TA} )</td>
</tr>
<tr>
<td>SV</td>
<td>VECTOR USED TO CONSTRUCT ( R ), ( T ) VECTORS. PARALLEL TO THE APPROACH ASYMPTOTE</td>
</tr>
<tr>
<td>SW</td>
<td>SINE OF ( \text{W} )</td>
</tr>
<tr>
<td>SX</td>
<td>VARIABLE USED TO DETERMINE SIGNS OF ( \text{DBI} ), ( \text{DBR} )</td>
</tr>
<tr>
<td>TANG</td>
<td>INTERMEDIATE VARIABLE FOR CALCULATION OF ( \text{AUXF} )</td>
</tr>
<tr>
<td>TA</td>
<td>TRUE ANOMALY FOR CALCULATION OF ( \text{AUXF} )</td>
</tr>
<tr>
<td>THS</td>
<td>INTERMEDIATE ANGLE FOR CALCULATION OF ( \text{W} )</td>
</tr>
<tr>
<td>TV</td>
<td>VECTOR USED TO CALCULATE ACTUAL AND DESIRED ( B ) DOT ( T )</td>
</tr>
<tr>
<td>VIMH</td>
<td>VELOCITY AT INFINITY</td>
</tr>
<tr>
<td>VX</td>
<td>MAGNITUDE OF THE VELOCITY VECTOR ( \text{V} )</td>
</tr>
<tr>
<td>WMAG</td>
<td>MAGNITUDE OF VECTOR NORMAL TO ORBITAL PLANE IN INCLINATION SYSTEM</td>
</tr>
<tr>
<td>WV</td>
<td>VECTOR NORMAL TO ORBITAL PLANE IN INCLINATION AND INERTIAL SYSTEMS</td>
</tr>
<tr>
<td>W</td>
<td>ARGUMENT OF PERIAPSIS</td>
</tr>
<tr>
<td>Z</td>
<td>APPROACH ASYMPTOTE IN INCLINATION SYSTEM</td>
</tr>
</tbody>
</table>

**COMMON USES:**
- NINETY ONE
- TWO
- ZERO
IMPACT Analysis

The impact parameters B-T and B-R form a convenient set of variables for the description of the approach geometry for lunar and interplanetary missions. Let a reference Cartesian coordinate system XYZ (ecliptic in S-T-R) be established at the center of the target body. Let \( \vec{V}_\infty \) denote the hyperbolic excess velocity of the spacecraft in the XYZ system. An auxiliary coordinate system B-S-T may be constructed relative to the \( \vec{V}_\infty \) by the definitions

\[
\hat{S} = \frac{\vec{V}_\infty}{V_\infty}, \quad \hat{T} = \frac{\hat{S} \times \hat{K}}{|\hat{S} \times \hat{K}|}, \quad \hat{R} = \hat{S} \times \hat{T} \tag{1}
\]

Therefore \( \hat{S} \) is in the direction of the approach asymptote, \( \hat{T} \) lies along the intersection of the impact plane (the plane normal to \( \hat{S} \) and passing through the center of the planet) and the reference plane (XY-plane), and \( \hat{R} \) completes the right-handed system. The \( \hat{R} \) vector lies in the impact plane and is directed to the incoming asymptote. Then B-T and B-R have the usual vector definitions.

![Figure 1. Impact Plane Parameters](image)

In the optional part of the subroutine, the target impact parameter \( \vec{E}_r \) associated with \( \hat{S} \) and a target inclination \( i \) (relative to target planet equator) and radius of closest approach \( r_{CA} \) is computed. However given an approach asymptote \( \hat{S} \) there are generally four trajectories with the same values of \( i \) and \( r_{CA} \). Two of these trajectories are retrograde and
two are posigrade. For each type of motion there are two distinct planes that have the same inclination and include the $\hat{S}$ vector. These are distinguished by the direction of motion when the approach asymptote is crossed, i.e., whether the motion is from north to south (northern approach) or from south to north (southern approach). Let $\theta = \alpha + 90^\circ$. Then setting the target inclinations to the following values determines the trajectory which will be specified:

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>posigrade with northern approach</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>posigrade with southern approach</td>
</tr>
<tr>
<td>$180 + \alpha$</td>
<td>retrograde with northern approach</td>
</tr>
<tr>
<td>$180 - \alpha$</td>
<td>retrograde with southern approach</td>
</tr>
</tbody>
</table>

The possible trajectories are illustrated in Figure 2.

![Figure 2. Possible Trajectories with Same Inclination](image)

The detailed computations for the basic part of the program are straightforward. Using the standard conic abbreviations,

\[ c = \frac{1}{\sqrt{r \times \vec{v}}} \]  
\[ \phi = \frac{r \times \vec{v}}{c} \]  
\[ p = \frac{c^2}{\mu} \]  
\[ a = \frac{r}{2 - rv^2/\mu} \]
The computations for the optional part of the program which converts the \( i \) and \( r_{\text{CA}} \) into an equivalent \( \overrightarrow{B} \) proceed as follows. The approach asymptote is first converted into target planet equatorial coordinates and its right ascension and declination computed.
\[ \hat{S}_q = \phi_{ECEQ} \hat{S} \]

\[ \theta_S = \tan^{-1} \left( \frac{(S_q)_y}{(S_q)_x} \right) \]  \hspace{2cm} (19)

\[ \delta_S = \sin^{-1} (S_q)_z \]

The angle \( \Delta \theta \) between the ascending node of the trajectory and the right ascension of the approach asymptote is from Napier's rule

\[ \sin \theta = \frac{\tan \delta_S}{\tan \iota} \]  \hspace{2cm} (20)

after assuring that \( |\iota| > |\delta_S| \). The ascending node of the trajectory is then computed recalling the definitions of the angle \( \iota \)

\[ \Omega = \theta_S + \Delta \theta \ (\pm \pi) \]  \hspace{2cm} (21)

Thus the unit vector to the ascending node is given by

\[ \hat{R}_A = (\cos \Omega, \sin \Omega, 0) \]  \hspace{2cm} (22)

The normal to the orbital plane (in target planet equatorial coordinates) is

\[ \hat{w}_q = \frac{\hat{S}_q \times \hat{R}_A}{|\hat{S}_q \times \hat{R}_A|} \]  \hspace{2cm} (23)

This is now converted to the ecliptic coordinate system

\[ \hat{w}_C = \rho_{ECEQ} \hat{w}_q \]  \hspace{2cm} (24)

The unit vector in the desired \( \hat{S}^{\infty} \) direction is

\[ \hat{S}^{\infty} = \frac{\hat{S} \times \hat{w}_C}{|\hat{S} \times \hat{w}_C|} \]  \hspace{2cm} (25)
The magnitude of the $\mathbf{E}^*$ vector is given by

$$B^* = r_{CA} \sqrt{1 + \frac{2 \mu}{r_{CA} V_{\infty}^2}}$$

Then the target impact parameter is $\mathbf{E}^* = \mathbf{E}^* \mathbf{E}^*$. The target values are then given by their obvious definitions

$$B \cdot T^* = E^* \cdot \hat{T}$$
$$B \cdot H^* = E^* \cdot \hat{R}$$

Finally, the hyperbolic time from $(\mathbf{F}, \mathbf{V})$ to pericenter is computed from the conic formula

$$\tanh \frac{P}{2} = \sqrt{\frac{e - 1}{e + 1}} \tan \frac{\Delta}{2}$$
$$t = \sqrt{-\frac{\Delta}{\mu}} \left( e \sinh P - P \right)$$

SUBROUTINE  INPUTZ

PURPOSE:  TO CONVERT THE INPUT INFORMATION FOR THE VIRTUAL MASS PROGRAM INTO VARIABLES COMPATIBLE WITH THE REST OF THE VIRTUAL MASS SUBROUTINES

CALLING SEQUENCE:  CALL INPUTZ(RS,NTP,IPRINT)

ARGUMENTS  RS(6)  I  INERTIAL STATE OF S/C AT INITIAL TIME
             NTP  I  CODE OF TARGET BODY
             IPRINT  I  INITIAL INFORMATION PRINT FLAG
                        =0  PRINT INITIAL DATA
                        =1  DO NOT PRINT INITIAL DATA

SUBROUTINES SUPPORTED:  VMP

SUBROUTINES REQUIRED:  TIME  SPACE

LOCAL SYMBOLS:  D  INTERMEDIATE VARIABLE FOR PRINTOUT PURPOSES
                 D2  JULIAN DATE OF FINAL TRAJECTORY TIME
                 IDAY  DAY OF CALENDAR DATE OF FINAL TRAJECTORY TIME
                 IHR  HOUR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
                 IMIN  MINUTE OF CALENDAR DATE OF FINAL TRAJECTORY TIME
                 IMO  MONTH OF CALENDAR DATE OF FINAL TRAJECTORY TIME
                 INERR  NOT USED
                 IP  CODE OF I-TH PLANET FOR STORAGE OF PMASS ARRAY
                 IYR  YEAR OF CALENDAR DATE OF FINAL TRAJECTORY TIME
                 LDAY  DAY OF CALENDAR DATE OF INITIAL TIME
                 LHR  HOUR OF CALENDAR DATE OF INITIAL TIME
                 LMIN  MINUTE OF CALENDAR DATE OF INITIAL TIME
                 LMO  MONTH OF CALENDAR DATE OF INITIAL TIME
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lyr</td>
<td>Year of calendar date of initial time</td>
</tr>
<tr>
<td>Seci</td>
<td>Second of calendar date of final time</td>
</tr>
<tr>
<td>Secl</td>
<td>Second of calendar date of initial time</td>
</tr>
<tr>
<td>Tp</td>
<td>Intermediate variable for calculation of computing interval</td>
</tr>
</tbody>
</table>

**Common computed/used: V**

**Common computed:**
- F
- Inc
- Ipr
- Itat
- Kount
- Nbody

**Common used:**
- Nbodyi
- No
- Pmass
- Zero
SUBROUTINE INSERS

PURPOSE: TO CONTROL THE PROCESSING OF AN ORBITAL INSERTION EVENT.

CALLING SEQUENCE: CALL INSERS(DTIME)

ARGUMENTS: DTIME = TIME INTERVAL FROM DECISION TO EXECUTION
(DAYS)

SUBROUTINES SUPPORTED: GIDANS

SUBROUTINES REQUIRED: COPINS NONINS PECEQ

LOCAL SYMBOLS: DA = DESIRED SEMIMAJOR AXIS
DE = DESIRED ECCENTRICITY
DI = DESIRED INCLINATION
DN = DESIRED LONGITUDE OF ASCENDING NODE
DWTP = DESIRED ARGUMENT OF PERIAPSIS SHIFT OR
DESIRED ARGUMENT OF PERIAPSIS
ECEQI = ECLIPTIC TO EQUATORIAL TRANSFORMATION
GM = GRAVITATIONAL CONSTANT OF TARGET BODY
IEX = UNEXECUTABLE EVENT CODE
=0 EVENT IS EXECUTABLE
=1 NO EXECUTABLE SOLUTION FOUND

IOPT = INSERTION STRATEGY OPTION
=1 COPLANAR INSERTION
=2 NONPLANAR INSERTION
RSP = SPACECRAFT POSITION IN ECLIPTIC COORDS
RSQ = SPACECRAFT POSITION IN EQUATORIAL COORDS
TEX = TIME INTERVAL TO EXECUTION (SECONDS)
VSP = SPACECRAFT VELOCITY IN ECLIPTIC COORDS
VSQ = SPACECRAFT VELOCITY IN EQUATORIAL COORDS

COMMON COMPUTED/USED: DELTAV

COMMON COMPUTED: DELV KTIM KWIT

COMMON USED: ALNGTH DI F KMXQ KTAR
KUR MBOD MB NTM PHASS
INSERS Analysis

INSERS controls the processing of an orbital insertion event. The subroutine COPINS and NONINS perform the actual computations for the co-planar and non-planar options respectively.

INSERS first records the specific parameter values for the current orbit insertion event.

It then computes the current state $(\mathbf{r}, \mathbf{v})$ of the spacecraft in target-planet centered ecliptic coordinates. Subroutine PECQ is called to compute the transformation matrix $\Phi_{ECEQ}$ from ecliptic to equatorial coordinates. The planet centered equatorial coordinates are then

\[
\mathbf{r}_q = \Phi_{ECEQ} \mathbf{r}
\]

\[
\mathbf{v}_q = \Phi_{ECEQ} \mathbf{v}
\]

This state is then sent to COPINS or NONINS for the computation of the insertion velocity $\Delta \mathbf{v}$ and the time interval $t$ between the current time and the time at which the insertion should take place (based on conic propagation about the target body). The correction $\Delta v_q$ is then converted to ecliptic coordinates

\[
\Delta \mathbf{v} = \Phi^T \Delta \mathbf{v}_q
\]

If the event is a compute-only mode, the return is made to GIDANS.

If the event is to be executed the flag IEX (set by COPINS or NONINS to indicate success or failure) is then interrogated. If IEX = 1, no acceptable insertion event was found and so the executive flag YMLT is set to 1 before returning. If IEX = 0 an acceptable insertion was determined and so it is set up.
**INSERS Flow Chart**

**ENTER**

- Store current insertion parameters

- Compute ecliptic state of spacecraft wrt target planet $\vec{r}, \vec{v}$, compute $\Phi_{ECEQ}$, and compute planetcentric equatorial state $\vec{r}', \vec{v}'$

- $I_{OPT} = ?$

- Call COPINS for coplanar insertion computation of $\Delta \vec{v}, \Delta t$

- Call NONINS for non-planar insertion computation of $\Delta \vec{v}, \Delta t$

- Convert $\Delta \vec{v}$ to ecliptic

- $KEXQ = ?$

- $KEX = ?$

- $\Delta t^* = \Delta t / TM$

- RETURN
SUBROUTINE JACOBI

PURPOSE: TRANSFORMATION OF A REAL SYMMETRIC MATRIX TO DIAGONAL FORM BY A SUCCESSION OF PLANE ROTATIONS TO ANNIMATE THE OFF-DIAGONAL ELEMENTS AND SUBSEQUENT COMPUTATION OF THE EIGENVALUES AND EIGENVECTORS OF THAT MATRIX

CALLING SEQUENCE: CALL JACOBI(A,W2,V,M,F00)

ARGUMENTS: A I MATRIX TO BE DIAGONALIZED (WILL BE DESTROYED)
W2 O VECTOR OF EIGENVALUES (LENGTH M)
V O MATRIX OF EIGENVECTORS (M BY M DIMENSION)
M I DIMENSION OF SQUARE MATRIX A
F00 I FINAL OFF-DIAGONAL ANNihilation VALUE

SUBROUTINES SUPPORTED: EIGHY GUSSIM GUSS PRESIM SETEVN GUIDM GUID PRED

LOCAL SYMBOLS: AIPP INTERMEDIATE VARIABLE
AIPIP INTERMEDIATE VARIABLE-A(IPIP)
AIPJP INTERMEDIATE VARIABLE-A(IPJP)
AJPJP INTERMEDIATE VARIABLE-A(JPJP)
CS INTERMEDIATE VARIABLE
DEL DIFFERENCE IN ELEMENTS OF A
IREDO COUNTER
KR DIMENSION OF A
KRPI KR + 1
MM1 M - 1
RAD INTERMEDIATE VARIABLE
SN INTERMEDIATE VARIABLE
TN INTERMEDIATE VARIABLE
T1 LARGEST OFF-DIAGONAL ELEMENT
VIIIP INTERMEDIATE VARIABLE
<table>
<thead>
<tr>
<th>COMMON USED</th>
<th>ONE</th>
<th>TWO</th>
<th>ZERO</th>
</tr>
</thead>
</table>
JACOBI Analysis

The Jacobi method subjects a real, symmetric matrix $A$ to a sequence of transformations based on a rotation matrix:

$$ Q_k = \begin{bmatrix} \cos \phi_k & -\sin \phi_k \\ \sin \phi_k & \cos \phi_k \end{bmatrix} $$

where all other elements of the rotation matrix are identical with the unit matrix. After $n$ multiplications $A$ is transformed into:

$$ A' = 0_N^{-1} \cdots 0_1^{-1} A_1 \cdots 0_N $$

If $Q_k$ is chosen at each step to make a pair of off-diagonal elements zero, then $A'$ will approach diagonal form with the eigenvalues on the diagonal. The columns of $0_1 0_2 \cdots 0_N$ correspond to the eigenvectors of $A$.

The angle of rotation $\phi$ is chosen in the following way. If the four entries of $Q_k$ are in $(i,i)$, $(i,j)$, $(j,i)$ and $(j,j)$ then the corresponding elements of $0_1^{-1} A_1$ are

$$ b_{ii} = a_{ii} \cos^2 \phi + 2a_{ij} \sin \phi \cos \phi + a_{jj} \sin^2 \phi $$

$$ b_{ij} = b_{ji} = (a_{jj} - a_{ii}) \sin \phi \cos \phi + a_{ij} (\cos^2 \phi - \sin^2 \phi) $$

$$ b_{jj} = a_{ii} \sin^2 \phi - 2a_{ij} \sin \phi \cos \phi + a_{jj} \cos^2 \phi $$

If $\phi$ is chosen so that $\tan 2\phi = 2a_{ij} / (a_{ii} - a_{jj})$ then

$$ b_{ij} = b_{ji} = 0 $$

Each multiplication creates a new pair of zeros but will introduce a non-zero contribution to positions zeroed out on previous steps. However, successive matrices of the form $0_2^{-1} 0_1^{-1} A_1 0_2$ will approach the required diagonal form.

JACOBI Flow Chart

ENTER

Set initial V matrix to unity.
Set W2(1) = A(1)

Is A a 1x1 matrix?

YES → RETURN

NO →

Set T1 = ABS(A(2)). Scan upper off-diagonal elements of matrix A by rows to find greatest element in absolute value. Set T1 equal to this element.

Set IREDO = 0. Scan upper off-diagonal elements of matrix A by rows until a value greater than T1 is found. Pivot on this element.

Compute rotation angle Ψ.
Set IREDO = 1

Compute eigenvectors and diagonalize matrix A.

IREDO = 1?

YES →

B

T1 .LE. 0.001?

YES → Place diagonal from A into W2

NO →

T1 = T1 * 0.001

B

RETURN
SUBROUTINE LAUNCH

PURPOSE: TO COMPUTE THE INJECTION TIME, POSITION, AND VELOCITY FROM THE DEPARTURE ASYMPTOTE AND THE LAUNCH PROFILE

ARGUMENTS:
DI  JULIAN DATE AT INJECTION (OUTPUT)
RI  POSITION VECTOR AT INJECTION (OUTPUT)
VZ  VELOCITY VECTOR AT INJECTION (OUTPUT)

SUBROUTINES SUPPORTED: HELIO

SUBROUTINES REQUIRED: EPHEM ORA PECEQ

LOCALS SYMBOLS:
ANGLE  INTERMEDIATE ANGLE USED TO DEFINE TL
AZI  PLANETOCENTRIC AZIMUTH AT INJECTION (DEG)
BNAI  UNIT VECTOR NORMAL TO BNAI AND WHAT USED TO DEFINE THE P-Q ELEMENTS OF THE DEPARTURE HYPERBOLA
BNAI  MAGNITUDE OF THE NON-UNITIZED BNAI VECTOR
COSFL  COSINE OF FL
COSFS  COSINE OF FS
COSGAM  COSINE OF GAMMAI
COSPHI  COSINE OF FI
COSSIG  COSINE OF SIGMAI
COSWL  COSINE OF WL
C3  VIS VIVA ENERGY ON THE DEPARTURE HYPERBOLA
DD  INTERMEDIATE VARIABLE USED TO CALCULATE GREENWICH HOUR ANGLE
DLA  PLANETOCENTRIC EQUATORIAL DECLINATION OF THE DEPARTURE ASYMPTOTE
EQEC  TRANSFORMATION MATRIX FROM ECLIPTIC TO LAUNCH PLANET EQUATORIAL

FL  TRUE ANOMALY OF LAUNCH SITE POSITION VECTOR
TRUE ANOMALY OF DEPARTURE ASYMPTOTE

FLIGHT PATH ANGLE AT INJECTION

GREENWICH HOUR ANGLE

GRAVITATIONAL CONSTANT OF THE LAUNCH PLANET IN KM**3/SEC**2

ECCENTRICITY OF THE DEPARTURE HYPERBOLA

INTERMEDIATE VARIABLE USED TO COMPUTE GREENWICH HOUR ANGLE

HOUR OF INJECTION

MINUTE OF INJECTION

INDEX

INDEX

HOUR OF LAUNCH

MINUTE OF LAUNCH

UNIT VECTOR POINTING TOWARD PERIAPSIS OF THE HYPERBOLA

LATITUDE OF INJECTION

THE ANGLE FROM LAUNCH TO INJECTION

UNIT VECTOR NORMAL TO PMAT PointING IN THE DIRECTION OF MOTION

RIGHT ASCENSION AT INJECTION

RIGHT ASCENSION OF DEPARTURE ASYMPTOTE

JULIAN DATE FOR 1950

MAGNITUDE OF THE SPACECRAFT POSITION AT INJECTION

SPACECRAFT POSITION AT INJECTION

LAUNCH SITE POSITION UNIT VECTOR

SECOND OF INJECTION

SECOND OF LAUNCH
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHAT</td>
<td>UNIT SPACECRAFT VELOCITY VECTOR IN EQUATORIAL SYSTEM AT INJECTION</td>
</tr>
<tr>
<td>SINF</td>
<td>SINE OF FL</td>
</tr>
<tr>
<td>SINF6</td>
<td>SINE OF FS</td>
</tr>
<tr>
<td>SINGAM</td>
<td>SINE OF GAMMA</td>
</tr>
<tr>
<td>SINPHI</td>
<td>SINE OF FI</td>
</tr>
<tr>
<td>SINSIG</td>
<td>SINE OF SIGMA</td>
</tr>
<tr>
<td>SINWL</td>
<td>SINE OF WL</td>
</tr>
<tr>
<td>SLR</td>
<td>SIMI-LATUS RECTUM OF THE DEPARTURE HYPERBOLA</td>
</tr>
<tr>
<td>TB</td>
<td>TIME BETWEEN LAUNCH AND INJECTION IN SECONDS</td>
</tr>
<tr>
<td>TC</td>
<td>LENGTH OF PARKING ORBIT COAST IN SECONDS</td>
</tr>
<tr>
<td>TEST</td>
<td>INTERMEDIATE VARIABLE TO TEST FOR VIOLATION OF AZIMUTH CONSTRAINT</td>
</tr>
<tr>
<td>TFRAC</td>
<td>INTERMEDIATE VARIABLE USED TO CALCULATE GREENWICH HOUR ANGLE</td>
</tr>
<tr>
<td>THETA1</td>
<td>LONGITUDE AT INJECTION</td>
</tr>
<tr>
<td>TH</td>
<td>INTERMEDIATE VARIABLE USED TO CALCULATE CLOCK TIMES OF LAUNCH AND INJECTION</td>
</tr>
<tr>
<td>TI</td>
<td>INJECTION TIME IN DAYS REFERENCED TO MIDNIGHT OF THE LAUNCH DAY</td>
</tr>
<tr>
<td>TL</td>
<td>LAUNCH TIME IN DAYS REFERENCED TO MIDNIGHT OF THE LAUNCH DAY</td>
</tr>
<tr>
<td>TMN</td>
<td>INTERMEDIATE VARIABLE USED TO CALCULATE CLOCK TIMES OF LAUNCH AND INJECTION</td>
</tr>
<tr>
<td>TSTAR</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE GREENWICH HOUR ANGLE</td>
</tr>
<tr>
<td>TWO10</td>
<td>CONSTANT VALUE, EQUAL TO 24</td>
</tr>
<tr>
<td>VHL</td>
<td>MAGNITUDE OF VZ, THE INPUT VECTOR OF THE DEPARTURE ASYMPTOTE</td>
</tr>
<tr>
<td>VIMAG</td>
<td>MAGNITUDE OF SPACECRAFT VELOCITY AT INJECTION</td>
</tr>
<tr>
<td>WHAT</td>
<td>UNIT VECTOR NORMAL TO THE LAUNCH PLANE IN EQUATORIAL SYSTEM</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>WL</td>
<td>RIGHT ASCENSION OF THE LAUNCH SITE</td>
</tr>
<tr>
<td>WMAG</td>
<td>MAGNITUDE OF THE NON-UNITIZED WHAT VECTOR</td>
</tr>
<tr>
<td>XTIN</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE CLOCK TIMES OF LAUNCH AND INJECTION</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED:** SIGNAL

**COMMON COMPUTED:** NO

**COMMON USED**:

- ALNGTH
- OPA
- FI
- FOUR
- KOAST
- NINETY
- NLP
- ONE
- PHILS
- PMASS
- PSI1
- PSI2
- RAD
- RAP
- RPRAT
- RP
- THEDOT
- THELS
- TIM1
- TIM2
- TM
- TWO
- VHPM
- XP
- ZERO
LAUNCH Analysis

LAUNCH computes the injection time, position and velocity from the departure velocity $v_{HE}$ (computed in HELIO) and the launch profile parameters input by the user.

The rotation matrix $\Phi_{ECEQ}$ defining the transformation from ecliptic to equatorial coordinates is first computed ($PECEQ$). The departure velocity $v_{HE}$ is then normalized and converted into ecliptic coordinates to yield the departure asymptote $\hat{S}$.

$$\hat{S} = \Phi_{ECEQ} \frac{v_{HE}}{v_{HE}}$$

Auxiliary information associated with $\hat{S}$ is then computed. The energy $C_3$, the declination $\phi_s$ and the right ascension $\theta_s$ of the departure asymptote, and the eccentricity of the departure hyperbola are given by

$$C_3 = v_{HE}^2$$
$$\sin \phi_s = \frac{S_z}{v_{HE}}$$
$$\tan \theta_s = \frac{S_y}{S_x}$$
$$e = 1 + \frac{r_p C_3}{\mu}$$

where $r_p$ is the desired parking orbit radius and $\mu$ is the gravitational constant of the launch planet.

The unit normal $\hat{W}$ to the launch plane in equatorial coordinates is then computed. $\hat{W}$ is defined by

$$W_x = \cos \phi_L \sin \Sigma_L$$
$$W_y = \frac{-W_x S_x + W_z S_z}{S_x^2 + S_y^2} \left[ 1 - \left( \frac{S_x^2 + W_z^2}{S_x^2 + S_y^2} \right)^{\frac{1}{2}} \right]$$
$$W_z = \frac{W_y S_y + W_z S_z}{S_x}$$

These equations are used to calculate the departure velocity and asymptote in the ecliptic frame.
where \( \phi_L \) is the launch site latitude, \( \Sigma_L \) is the launch azimuth, and \( k = +1 \) or \(-1\) for the long or short coast time models respectively. The second equation defines an implicit constraint on \( \Sigma_L \)

\[
\sin^2 \Sigma_L \leq \frac{\cos^2 \phi_s}{\cos^2 \phi_L} \tag{4}
\]

The right ascension at launch \( \Theta_L \) may now be defined by

\[
\cos \Theta_L = \frac{W \sin \phi_L \sin \Sigma_L + w \cos \Sigma_L}{W^2 - 1}
\]

\[
\sin \Theta_L = \frac{W \sin \phi_L \sin \Sigma_L - W \cos \Sigma_L}{W^2 - 1} \tag{5}
\]

and the unit vector toward the launch position is then

\[
\mathbf{R}_L = (\cos \phi_L \cos \Theta_L, \cos \phi_L \sin \Theta_L, \sin \phi_L) \tag{6}
\]

The complementary unit vectors \( \hat{F}, \hat{Q} \) defining the orientation of the hyperbola within the launch plane are now introduced. Let

\[
\hat{B} = \hat{S} \times \hat{W} \tag{7}
\]

The true anomaly of the departure asymptote is \( \cos f_s = -\frac{1}{e} \). Then \( \hat{F} \) and \( \hat{Q} \) are given as

\[
\hat{F} = \hat{S} \cos f_s + \hat{B} \sin f_s
\]

\[
\hat{Q} = \hat{S} \sin f_s - \hat{B} \cos f_s \tag{8}
\]

The true anomaly of the launch site \( f_L \) may now be given

\[
\cos f_L = \hat{R}_L \cdot \hat{F}
\]

\[
\sin f_L = \hat{R}_L \cdot \hat{Q} \tag{9}
\]

The angle \( \Psi_B \) between launch and injection is

\[
\Psi_B = 2\pi - f_L + f_I \tag{10}
\]
where $f_1$ is the desired true anomaly at injection read in as input.

The coast time $t_c$ may now be computed from

$$t_c = \left[ \Psi_B - (\Psi_1 + \Psi_2) \right] k_\phi$$

where $\Psi_1$ and $\Psi_2$ are the angles of the first and second burns and $k_\phi$ is the inverse of the parking orbit coast rate, all of which are read in as input.

The time between launch and injection is therefore

$$t_B = t_1 + t_2 + t_c$$

where $t_1$ and $t_2$ are the input time durations of the first and second burns.

The unit vector to injection is

$$\mathbf{R}_I = \mathbf{P} \cos f_1 + \mathbf{Q} \sin f_1$$

The semi-latus rectum $p$ is

$$p = \frac{\mu (e^2 - 1)}{c_3}$$

The radius magnitude to injection is

$$R_I = \frac{p}{1 + e \cos f_1}$$

The injection speed is

$$v_I = \sqrt{c_3 + \frac{2 \mu}{R_I}}$$

The path angle at injection is

$$\cos \Gamma_I = \frac{\sqrt{\mu p}}{R_I v_I}$$
The injection latitude is

\[ \sin \phi_I = \hat{e}_x \]

(18)

The injection right ascension is

\[ \tan \theta_I = \frac{e_y}{e_x} \]

(19)

The injection longitude is

\[ \theta_I = \theta_L + \theta_I - \theta_L - \omega t_B \]

(20)

where \( \theta_L \) is the longitude of the launch site and \( \omega \) is the rotation rate of the launch planet, both being read in as input.

The injection azimuth is

\[ \cos \Sigma_I = \frac{\cos (f - f_I) \sin \Phi_I}{\sin (f - f_I) \cos \Phi_I} \]

(21)

The launch time on the day of launch is

\[ t_L = \frac{(\theta_L - \theta_L - \text{GHA}) \mod 2\pi}{\omega} \]

(22)

where GHA is the Greenwich hour angle at 0h UT of the launch date

\[ \text{GHA} = 100.07554260 + 0.9856473460 T_d + 279015 \times 10^{-3} T_d^3 \]

(23)

where \( T_d \) = days past 0h January 1, 1950.

The injection radius vector is now computed from

\[ r_I = \hat{e}_x \]

(24)

\[ \hat{V}_I = \frac{V_I}{R_I} \left[ (\hat{w} \times \hat{r}_I) \cos \Gamma_I + \hat{r}_I \sin \Gamma_I \right] \]

The injection time is

\[ t_I = t_o + t_L + t_B \]

(25)
where \( T_0 \) is the Julian date of the launch calendar data.

The injection position and velocity are now rotated into the ecliptic plane. The position and velocity of the launch planet at the time \( T_x \) are computed and added to the injection state to get the heliocentric injection state.

Reference: Space Research Conic Program, Phase II, May 1, 1969, Jet Propulsion Laboratory, Pasadena, California.
LAUNCH Flow Chart

ENTER

Convert departure asymptote \( \vec{v}_H \) to equatorial coordinates, normalize to get \( \vec{s} \).

Correct nominal launch azimuth if needed.

Compute normal to launch plane \( \vec{w} \).

Compute launch state variables.

Compute \( \hat{F} \) and \( \hat{Q} \) of launch hyperbola.

Compute time from launch to injection.

Compute injection state \( \vec{r}_I, \vec{v}_I \) rel to planet.

Compute launch and injection time.

Compute state of L.P. at injection time
\[
\vec{r}_p, \vec{v}_p
\]

Compute heliocentric injection state
\[
\vec{r}_1 = \vec{r}_I + \vec{r}_p \\
\vec{v}_1 = \vec{v}_I + \vec{v}_p
\]

RETURN
SUBROUTINE LUNA

PURPOSE: TO CONTROL THE GENERATION OF THE ZERO ITERATE FOR LUNAR TARGETING

CALLING SEQUENCE: CALL LUNA

SUBROUTINES SUPPORTED: ZERIT

SUBROUTINES REQUIRED: LUNTAR MULTAR

LOCAL SYMBOLS: I INDEX

OSPH ORIGINAL SPHERE OF INFLUENCE OF TARGET PLANET IN A.U.

COMMON COMPUTED/USED: OTAR SPHERE

COMMON COMPUTED: BCON CAI IBARY ICOORD PCON
             RCA RPE SMA TCA TSPH
             TTOL

COMMON USED: ALNGTH DT FOUR KUR NTP
             ONE RP SPHFAC TEM ZDAT
LUNA Analysis

LUNA is the controlling subroutine for lunar zero iterate targeting. It first serves an interface role in which it initializes constants and renames variables for the other lunar targeting routines. It then calls LUNTA for the targeting of the lunar patched conic. When that is completed it calls MULTAR for the targeting of the multi conic trajectory. It then returns control to PRELM.

LUNA Flow Chart

1. ENTER
2. Initialize and rename constants.
3. Call LUNTA
4. Call MULTAR
5. RETURN
SUBROUTINE LUNCON

PURPOSE: TO COMPUTE THE ACTUAL VALUES OF THE TARGET PARAMETERS (A, BDT, BDR) FOR A LUNAR PATCHED CONIC TRAJECTORY DETERMINED BY CONTROL VALUES OF ALPHA, DELTA, AND THETA.

CALLING SEQUENCE: CALL LUNCON(ALPHAI, DELTAI, THETAI, AM, BDT, BDR, SIGNAL, ITR)

ARGUMENTS:
- ALPHAI I ANGLE DEFINING PERIGEE OF TRANSFER CONIC (RAD)
- DELTAI I DELINATION OF LSI POINT (RAD)
- THETAI I RIGHT ASCENSION OF LSI POINT (RAD)
- AM O SEMIMAJOR AXIS OF LUNAR CONIC
- BDT O IMPACT PARAMETER OF LUNAR CONIC
- BDR O IMPACT PARAMETER OF LUNAR CONIC
- SIGNAL I/O NOMINAL LAUNCH AZIMUTH OR THAT REQUIRED
- ITR O OUTPUT ITERATION COUNTER

SUBROUTINES SUPPORTED: LUNTAI

SUBROUTINES REQUIRED: CAREL IMPACT

LOCAL SYMBOLS:
- ALPHA ALPHAI IN DEGREES
- AQUT TEMPORARY LOCATION FOR AM
- CC ANGULAR MOMENTUM OF THE EARTH CENTERED TRANSFER CONIC
- CDDEL COSINE OF DELTAI
- CECC ECCENTRICITY OF THE EARTH CENTERED TRANSFER CONIC
- COSDEC COSINE OF DECLIN
- COSPL COSINE OF PHIL
- COSPS INTERMEDIATE VARIABLE TO TEST FOR VIOLATION OF SIGNAL CONSTRAINT
- COSSIG COSINE OF SIGNAL
- CP SEMI-LATUS RECTUM OF EARTH CENTERED TRANSFER CONIC
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMA</td>
<td>Semi-major axis of Earth centered transfer conic</td>
</tr>
<tr>
<td>CT</td>
<td>Cosine of ( \Theta_1 )</td>
</tr>
<tr>
<td>DELTA</td>
<td>Delta in degrees</td>
</tr>
<tr>
<td>EM</td>
<td>Eccentricity of lunar conic</td>
</tr>
<tr>
<td>GAMMI</td>
<td>Intermediate angle used to compute Earth centered transfer conic</td>
</tr>
<tr>
<td>I</td>
<td>Index</td>
</tr>
<tr>
<td>PHIL</td>
<td>Latitude of launch site</td>
</tr>
<tr>
<td>POS</td>
<td>Spacecraft position and velocity at LSI point in M^* centered Earth equatorial coordinates</td>
</tr>
<tr>
<td>PPM</td>
<td>Dummy variable for CALL CAREL</td>
</tr>
<tr>
<td>QQM</td>
<td>Dummy variable for CALL TO CAREL</td>
</tr>
<tr>
<td>RAD</td>
<td>Radians to degrees conversion factor</td>
</tr>
<tr>
<td>RMAG</td>
<td>Magnitude of the RI vector</td>
</tr>
<tr>
<td>ROUT</td>
<td>Velocity at LSI in geocentric equatorial system</td>
</tr>
<tr>
<td>RPM</td>
<td>Radius of periapsis of lunar conic</td>
</tr>
<tr>
<td>SDEL</td>
<td>Sine of delta</td>
</tr>
<tr>
<td>SHAT</td>
<td>Unit vector pointing from the Earth to the point defined by delta, ( \Theta_1 )</td>
</tr>
<tr>
<td>SIGH</td>
<td>Signal in degrees</td>
</tr>
<tr>
<td>SINDEC</td>
<td>Sine of Declin</td>
</tr>
<tr>
<td>SIMPS</td>
<td>Intermediate variable used to test for violation of signal constraint</td>
</tr>
<tr>
<td>SIMSIG</td>
<td>Sine of signal</td>
</tr>
<tr>
<td>SX</td>
<td>Sine of ( \Theta_1 )</td>
</tr>
<tr>
<td>TAM</td>
<td>True anomaly of the lunar conic corresponding to the RSI vector</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>TFLP</td>
<td>Time of flight from periapsis corresponding to the RSI vector</td>
</tr>
<tr>
<td>THETA</td>
<td>Theta in degrees</td>
</tr>
<tr>
<td>VMAG</td>
<td>Spacecraft velocity magnitude used to calculate declin</td>
</tr>
<tr>
<td>WHAT</td>
<td>Unit vector normal to the Earth-phase</td>
</tr>
<tr>
<td>WMAG</td>
<td>Angular momentum constant</td>
</tr>
<tr>
<td>WM</td>
<td>Argument of periapsis of the lunar conic</td>
</tr>
<tr>
<td>WMM</td>
<td>Dummy variable for call to careg</td>
</tr>
<tr>
<td>XHAT</td>
<td>Same as SHAT vector</td>
</tr>
<tr>
<td>XIM</td>
<td>Inclination of the conic</td>
</tr>
<tr>
<td>XMM</td>
<td>Longitude of the ascending node of the lunar conic</td>
</tr>
<tr>
<td>YHAT</td>
<td>Cross product of the WHAT and XHAT vectors</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED**

- RI
- RSI

**COMMON COMPUTED**

- DECLIN

**COMMON USED**

- EMU
- EQLQ
- KOAST
- NINETY
- ONE
- PHILS
- RMQ
- RPE
- SIGMA
- TAU
- TSPI
- TWO
- ZERO
LUNCON Analysis

The point of intersection of the Earth-centered conic with the lunar sphere of influence (LSI) is determined by the angles $\theta_L$ and $\delta_L$. Relative to the moon in Earth-equatorial coordinates that point is

$$\vec{r}_{SI} = \begin{bmatrix} R_{SI} \cos \delta \cos \theta_L \\ R_{SI} \cos \delta \sin \theta_L \\ R_{SI} \sin \delta \end{bmatrix} \quad (1)$$

where $R_{SI}$ is the radius of the LSI. Relative to the earth that point is

$$\vec{r}_q = \vec{r}_M + \vec{r}_{SI} \quad (2)$$

where $\vec{r}_M$ is the radius vector to the center of the moon at the time of LSI intersection $t_{SI}$ in earth equatorilial coordinates.

There are at most two planes which contain $\vec{r}_q$ and satisfy the launch latitude $\phi$ and azimuth $\Lambda$ constraints. Let $\vec{\phi}$ denote the unit normal to either of these planes. Now let $\vec{\phi}_L$, $\theta_L$, $\phi_L$ denote the unit vector, longitude, and latitude of the launch site. Construct a local horizon coordinate system at the launch site as indicated in Figure 1.

![Figure 1. Local Horizon Coordinate System](image-url)
Here \( \hat{\mathbf{h}} = \hat{\mathbf{r}} \), \( \hat{\mathbf{r}} \) is normal to \( \hat{\mathbf{h}} \) in the \( \hat{\mathbf{r}}-\hat{\mathbf{L}} \) plane, and \( \hat{\mathbf{h}} = \hat{\mathbf{r}} \times \hat{\mathbf{L}} \).

In the local horizon system, the position and velocity are very simply represented

\[
\begin{align*}
\mathbf{r}_h &= \mathbf{r} [0, 0, 1]^T \\
\mathbf{v}_h &= \mathbf{v} [\cos \delta \sin \Sigma, \cos \delta \cos \Sigma, \sin \delta]^T
\end{align*}
\]

(3)

where \( \Sigma \) is the launch azimuth and \( \delta \) is the declination wrt the local horizontal. Thus

\[
\hat{\mathbf{v}}_h = \frac{\mathbf{r}_h \times \mathbf{v}_h}{|\mathbf{r}_h \times \mathbf{v}_h|} = \begin{bmatrix} -\cos \delta \\ \sin \delta \\ 0 \end{bmatrix}
\]

(4)

The transformation matrix converting a vector in the local horizon system to the equatorial system is

\[
\mathbf{T} = \begin{bmatrix}
-\sin \theta_0 & -\sin \phi_0 \cos \theta_0 & \cos \phi_0 \cos \theta_0 \\
\cos \theta_0 & -\sin \phi_0 \cos \theta_0 & \cos \phi_0 \sin \theta_0 \\
0 & \cos \phi_0 & \sin \phi_0
\end{bmatrix}
\]

(5)

Therefore since \( \mathbf{v}_h = \mathbf{T} \mathbf{v}_h \), the \( \mathbf{z} \)-component of \( \mathbf{\hat{v}} \) in the equatorial coordinate system is

\[
\mathbf{v}_z = \cos \phi_0 \sin \Sigma
\]

(6)

Since \( \mathbf{\hat{v}} \) is a unit normal it must satisfy both \( \mathbf{\hat{v}} \cdot \mathbf{v} = 1 \) and \( \mathbf{\hat{v}} \cdot \mathbf{\hat{v}} = 0 \) where \( \mathbf{\hat{v}} = \frac{\mathbf{v}}{|\mathbf{v}|} \). Solving for the two remaining components of \( \mathbf{\hat{v}} \),

\[
\begin{align*}
\mathbf{\hat{v}}_y &= \frac{-\mathbf{\hat{v}}_z S_y + \mathbf{\hat{v}}_x S_z + \sqrt{1 - (S_z^2 + S_x^2)}}{S_x^2 + S_y^2} \\
\mathbf{\hat{v}}_x &= -\frac{(\mathbf{\hat{v}}_y S_y + \mathbf{\hat{v}}_z S_z)}{S_x}
\end{align*}
\]

(7)

To eliminate the ambiguity of sign in (7) the short-coast plane corresponding to the negative sign is used. Note that (7) also imposes a constraint on the launch azimuth.
\[
\sin^2 \Sigma \leq \frac{1}{2} \left( \frac{a^2}{c^2} \right)
\]

(9)

Now choose \( \mathbf{\hat{u}} = \mathbf{\hat{w}} \times \mathbf{\hat{e}} \) to complete a right hand system \((\mathbf{\hat{e}}, \mathbf{\hat{v}}, \mathbf{\hat{w}})\). Then the position at LSI relative to the earth is \((R, 0, 0)\). Now let \( \alpha \) determine the perigee point in the orbital plane \( (\mathbf{\hat{w}} = 0) \) measured counterclockwise from the \(-\mathbf{\hat{e}}\) axis. Then the perigee point is \((-r, \cos \alpha, -r \sin \alpha, 0)\) where \( r \) is the parking orbit radius (input). Therefore the true anomaly of the earth centered conic at the LSI is given by

\[
f_{SI} = 180 - \alpha
\]

(10)

The two equations \( x = \frac{a(1-e^2)}{1+e \cos f_{SI}} \) and \( x = a(1-e) \) may be solved simultaneously for the semi-major axis \( a \) and eccentricity \( e \) of the unique earth centered conic

\[
e = \frac{R - r_p}{r_p - R_1 \cos f_{SI}}
\]

(11)

\[
a = \frac{r_p}{1-e}
\]

(12)

Thus the velocity of the earth centered conic at the LSI is in the \((\mathbf{\hat{e}}, \mathbf{\hat{v}}, \mathbf{\hat{w}})\) system

\[
\mathbf{v}_o = \sqrt{\frac{a(1-e^2)}{1+e \cos f_{SI}}} \, e \sin f_{SI}
\]

\[
\begin{bmatrix}
\mu a(1-e^2) / R_1 \\
0
\end{bmatrix}
\]

(13)

Transforming to the earth equatorial coordinate system

\[
\mathbf{v}_q = \begin{bmatrix}
S_x & U_x & W_x \\
S_y & U_y & W_y \\
S_z & U_z & W_z
\end{bmatrix}
\]

(14)
Now if \((\vec{r}_{NQ}, \vec{v}_{NQ})\) are the position and velocity of the moon at \(t_{SI}\) Earth-centered coordinates and \((\vec{r}_Q, \vec{v}_Q)\) are the position and velocity of the spacecraft at \(t_{SI}\), then the state of the spacecraft with respect to the moon at \(t_{SI}\) is in earth equatorial coordinates

\[
\begin{align*}
\vec{r}_{SI} &= \vec{r}_Q - \vec{r}_{NQ} \\
\vec{v}_{SI} &= \vec{v}_Q - \vec{v}_{NQ}
\end{align*}
\]

(15)

Using the transformation matrix \(\Phi_{EQLQ}\) defining transformations from earth equatorial to lunar equatorial the state in the LQ system is

\[
\begin{align*}
\vec{r}_{SQ} &= \Phi_{EQLQ} \vec{r}_{SI} \\
\vec{v}_{SQ} &= \Phi_{EQLQ} \vec{v}_{SI}
\end{align*}
\]

(16)

The impact plane parameters \(B\cdot T\) and \(B\cdot R\), and the inclination \(\chi\), may now be computed by calling subroutines ACTB and CAREL.
SUBROUTINE LUNTAR

PURPOSE: TO GENERATE A PATCHED CONIC TRAJECTORY FOR LUNAR MISSIONS CONSISTENT WITH TARGET PARAMETERS AT THE MOON OF (ACA, RCA, ICA, TCA) AND LAUNCH PARAMETERS (PHIL, THETAL, SIGMA).

CALLING SEQUENCE: CALL LUNTAR

SUBROUTINES SUPPORTED: LUNCON

SUBROUTINES REQUIRED: LUNCON EPHEM IMPACT MATIN ORB PECEQ

LOCAL SYMBOLS:
- AA: SEMI-MAJOR AXIS OF THE LUNAR CONIC FOR THE NOMINAL TRAJECTORY
- ALNGTH: SAME AS AU
- ALPHAI: REFINED ANGLE (RADIANS) DEFINING POSITION OF PERIGEE ON THE TRANSFER CONIC (NOMINALLY SET TO FIVE DEGREES)
- ALPI: PERTURBED VALUE OF ALPHAI USED TO SOLVE FOR RCA, ICA, ACA
- AUDAY: CONVERTS KM/SEC TO AU/DAY
- AUS: SAME AS AU
- AU: CONVERTS KILOMETERS (KM) TO ASTRONOMICAL UNITS (AU)
- BDR: B DOT R FOR THE NOMINAL TRAJECTORY
- BDT: B DOT T FOR THE NOMINAL TRAJECTORY
- BING: OBTAINABLE INCLINATION USED TO CALCULATE DESIRED B DOT T, B DOT R
- DELI: PERTURBED VALUE OF DELTAI USED TO SOLVE FOR RCA, ICA, ACA
- DELTAI: REFINED ANGLE (RADIANS) DEFINING DECLINATION OF THE LSI POINT (NOMINALLY SET TO DELTA0)
- DELTAO: DECLINATION OF THE MOON'S POSITION AT TIME TSI
- DLET: TIME FROM TSI TO TCA IN SECONDS
- DEL: REFINING VALUES FOR ALPHAI, DELTAI, THETAL
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMON</td>
<td>INTERMEDIATE VARIABLE USED TO LIMIT THE DEL VALUES FOR EACH ITERATION</td>
</tr>
<tr>
<td>EGG</td>
<td>DESIRED ECCENTRICITY OF THE LUNAR CONIC</td>
</tr>
<tr>
<td>ECEQ</td>
<td>TRANSFORMATION MATRIX FROM ECLIPTIC TO EARTH EQUATORIAL</td>
</tr>
<tr>
<td>ECLQ</td>
<td>TRANSFORMATION MATRIX FROM ECLIPTIC TO LUNAR EQUATORIAL</td>
</tr>
<tr>
<td>ERR</td>
<td>VECTOR OF DIFFERENCES BETWEEN DESIRED AND NOMINAL VALUES OF B DOT T, B DOT R, ACA</td>
</tr>
<tr>
<td>ITAR</td>
<td>LOGIC CONTROLLING INDICATOR</td>
</tr>
<tr>
<td></td>
<td>=1 IMPROVE ACA ONLY</td>
</tr>
<tr>
<td></td>
<td>=2 IMPROVE RCA, ICA, ACA</td>
</tr>
<tr>
<td>ITER</td>
<td>ITERATION COUNTER FOR NOMINAL TRAJECTORIES</td>
</tr>
<tr>
<td>IT</td>
<td>ITERATION COUNTER FOR PERTURBED TRAJECTORIES</td>
</tr>
<tr>
<td>I</td>
<td>INDEX</td>
</tr>
<tr>
<td>J</td>
<td>INDEX</td>
</tr>
<tr>
<td>K</td>
<td>INDEX</td>
</tr>
<tr>
<td>OMEMAT</td>
<td>UNIT DUMMY MATRIX FOR CALL TO IMPACT</td>
</tr>
<tr>
<td>PAI</td>
<td>DUMMY VARIABLE FOR CALL TO LUNCON WHEN ITAR=1</td>
</tr>
<tr>
<td>PARP</td>
<td>DUMMY VARIABLE FOR CALL TO LUNCON WHEN ITAR=1</td>
</tr>
<tr>
<td>PARTA</td>
<td>INTERMEDIATE VARIABLE USED TO REFINE ACA WHEN ITAR=1</td>
</tr>
<tr>
<td>PARTH</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE DELT</td>
</tr>
<tr>
<td>PARTX</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE DELT</td>
</tr>
<tr>
<td>PARTY</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE DELT</td>
</tr>
<tr>
<td>PARTZ</td>
<td>INTERMEDIATE VARIABLE USED TO COMPUTE DELT</td>
</tr>
<tr>
<td>PHI</td>
<td>MATRIX RELATING PERTURBATIONS IN ALPHAI, DELTAI, AND THETAI TO CHANGES IN B DOT T, B DOT R, AND ACA</td>
</tr>
</tbody>
</table>
PSI  TARGETING MATRIX RELATING PERTURBATIONS IN B DOT T, B DOT R, AND ACA TO CHANGES IN ALPHAI, DELTAI, AND THETAI

PTAR PERTURBED VALUES OF AA, BDT, BDR, USED TO CALCULATE PHI

RAD CONVERTS DEGREES TO RADIANS

RMAG MAGNITUDE OF THE RMQ VECTOR

SIGMA LAUNCH AZIMUTH SET IN LUNCON (NOMINALLY 90 DEGREES)

TAR NOMINAL VALUES OF AA, BDT AND BDR USED TO CALCULATE PHI

THEI PERTURBED VALUES OF THETAI USED TO SOLVE FOR RCA, ICA, AND ACA

THETAI REFINED ANGLE (RADIANS) DEFINING RIGHT ASCENSION OF THE LSI POINT (NOMINALLY SET TO THETAO)

THETAO RIGHT ASCENSION OF THE MOONS POSITION AT TIME TSI

TM CONSTANT VALUE OF SECONDS PER DAY

TSICA DUMMY ARGUMENT FOR CALL TO IMPACT

COMMON COMPUTED/USED: DTAR EQLQ ITAG RMQ RSI

COMMON COMPUTED: EMU NO RME

COMMON USED: BCON DECLIN FIVE ONE OTAR
PCON PMASS RCA SMA TCA
TSPH ITOL TWO XP ZERO
LUNAR Analysis

LUNAR generates a patched conic trajectory arriving at closest approach to the Moon at a specified time \( t_{CA} \) and meeting prescribed target values at that point as well as standard launch quantities. The target parameters are

\[
\begin{align*}
    t_{CA} & \quad \text{Julian date of required closest approach (CA) referenced 1900} \\
    r_{CA} & \quad \text{Radius of CA} \\
    i_{CA} & \quad \text{Inclination (relative to lunar equator) at CA} \\
    a_{CA} & \quad \text{Semi-major axis at CA}
\end{align*}
\]

The launch parameters

\[
\begin{align*}
    \theta_L & \quad \text{Launch site latitude} \\
    \phi_L & \quad \text{Launch site longitude} \\
    \lambda_L & \quad \text{Launch azimuth (nominally set to 90\(^{\circ}\))} \\
    r_p & \quad \text{Parking orbit radius}
\end{align*}
\]

The eccentricity of the moon-centered hyperbola may be computed

\[
e_{CA} = 1 - \frac{r_{CA}}{a_{CA}} \quad (1)
\]

where \( a_{CA} < 0 \). The hyperbolic time \( \Delta t \) to go from \( r_{SI} \) (radius of lunar sphere of influence (LSI)) to periapsis may be computed from

\[
\Delta t = f(\mu, a_{CA}, e_{CA}, r_{SI}) \quad (2)
\]

where \( \mu \) is the lunar gravitational constant. The time at which the probe should intersect the LSI is then

\[
t_{SI} = t_{CA} - \Delta t \quad (3)
\]

1 The inclination must be specified according to the format described in IMPACT. For \( 0 \leq i < 90^\circ \) the inclinations \( \pm i \) prescribe prograde orbits while \( 180^\circ \pm i \) define retrograde orbits. The positive signs denote approaches from the north, the negative signs designate southern approaches.
The position $\mathbf{r}_{\text{EQ}}$ and velocity $\mathbf{v}_{\text{EQ}}$ of the moon at $t_{SI}$ relative to the earth in earth ecliptic (EC) coordinates are computed by calling ORB and EPHX. Transformation matrices $\mathbf{\phi}_{\text{ECQ}}$ and $\mathbf{\phi}_{\text{EQL}}$ defining transformations from EC to EQ (earth equatorial) and EQ to EQ (lunar equatorial) respectively are then computed by PECEQ. The position and velocity of the moon in the EQ system are

$$\mathbf{r}_{\text{EQ}} = \mathbf{\phi}_{\text{ECQ}} \mathbf{r}_{\text{EC}}$$

$$\mathbf{v}_{\text{EQ}} = \mathbf{\phi}_{\text{ECQ}} \mathbf{v}_{\text{EC}}$$

Call the point of intersection of the vector $\mathbf{r}_{\text{EQ}}$ with the LSI the bullseye point. Then in moon-centered Earth-equatorial coordinates the vector to the bullseye point is given by

$$\mathbf{r}_B = -\left( \frac{\mathbf{r}_{\text{EQ}}}{\mathbf{r}_{\text{EQ}}} \right) \mathbf{r}_{SI}$$

From this vector one can calculate a set of angular coordinates $$(\delta_o, \theta_o)$$ of the bullseye point. Any other point on the LSI is determined by giving general coordinates $$(\delta, \theta) = (\delta_o + \Delta\delta, \theta_o + \Delta\theta)$$. 

Now let such a set of coordinates by given. They determine a vector $\mathbf{r}_I$ from earth to the LSI (in the EQ system). The vector $\mathbf{r}_I$ along with the launch parameters $\theta_L$, $\theta_I$, $\Sigma_L$ then determines the plane of the Earth-LSI transfer (see LUNCON). Now let $\alpha$ be measured counter-clockwise in that plane from $-\mathbf{r}_I$. The parameter $\alpha$ specifies the location of the perigee point of the transfer conic, thus the vector to perigee is fixed as $\mathbf{r}_P$ where the perigee magnitude $r_P$ is fixed as input. The vectors $\mathbf{v}_P$ and $\mathbf{r}_I$ then determine a unique conic for the Earth-LSI phase (see LUNCON).

Let the state at the LSI on that conic (relative to Earth-equatorial coordinates) be denoted by $\mathbf{r}_1, \mathbf{v}_1$. The state relative to the moon may then be computed as

$$\mathbf{r}_1 = \mathbf{r}_I - \mathbf{r}_{\text{MQ}}$$

$$\mathbf{v}_1 = \mathbf{v}_I - \mathbf{v}_{\text{MQ}}$$

(6)
Thus the elements relative to the moon may be computed from standard conic formula. The three angles $(\delta, \theta, \alpha)$ form a set of independent controls to be varied to meet the three constraints $(r_{CA}, i_{CA}, a_{CA})$. The controls are depicted in Figure 1.

LUNAR uses the standard Newton-Raphson algorithm to refine the controls to meet the constraints. This targeting is done in two stages. In the first stage the controls $\delta$ and $\theta$ are held fixed at the bullseye point $(\delta_0, \theta_0)$ while $\alpha$ is varied until the semi-major axis target $a_{CA}$ is met. Then all three controls are varied to satisfy the three target constraints. The preliminary targeting of $a_{CA}$ is essential to the success of the procedure. Once the initial targeting is completed, the semi-major axis of future
iterations in the second stage will not vary much from the target value $e_{CA}$. For such iterates the excess hyperbolic velocity at the moon will be generally constant. This permits the substitution of the auxiliary impact plane parameters B-T and B-R for the less linear parameters of $r_{CA}$ and $e_{CA}$ (see IMPACT). In LUNTAR the impact plane parameters are referenced to the LQ system.

The procedure may now be described in detail. Suppose that in the first stage of targeting the current value of $\alpha$ is $\alpha_k$. Using the controls $(\alpha_k, \delta_0, \theta_0)$ the resulting semi-major axis is found to be $a_k$ (LUNCON). A perturbed value for the first control is then used $(\alpha_k + \Delta \alpha, \delta_0, \theta_0)$ producing a perturbed value of semi-major axis $(a_k + \Delta a)$. The $(k+1)^{st}$ value of $\alpha$ is then given by the standard numerical differencing approximation

$$\alpha_{k+1} = \alpha_k + \frac{\Delta \alpha}{\Delta a} (a_{CA} - a_k)$$  \hspace{1cm} (7)

The second stage of the targeting of the lunar patched conic uses the vector analogue of the above procedure. The current iterate $(\alpha_k, \delta_k, \theta_k)$ is input to LUNCON to obtain the current target values $(a_k, B_{T_k}, B_{R_k})$.

The target values B-T and B-R are determined from subroutine IMPACT and the errors of the $k^{th}$ iterate are computed $(e_{a}, e_{BT}, e_{BR})$. If all three errors are within tolerances, the procedure is terminated. Otherwise the sensitivity matrix $\phi$ is computed by numerical differencing as in the first stage

$$ \phi = \begin{bmatrix} \frac{\Delta a}{\Delta \alpha} & \frac{\Delta a}{\Delta \delta} & \frac{\Delta a}{\Delta \theta} \\ \frac{\Delta B_{T}}{\Delta \alpha} & \frac{\Delta B_{T}}{\Delta \delta} & \frac{\Delta B_{T}}{\Delta \theta} \\ \frac{\Delta B_{R}}{\Delta \alpha} & \frac{\Delta B_{R}}{\Delta \delta} & \frac{\Delta B_{R}}{\Delta \theta} \end{bmatrix}$$ \hspace{1cm} (8)

The inverse of $\phi$ is the targeting matrix. The $k+1$ iterate is then defined to be
\[
\begin{bmatrix}
\alpha \\
\delta \\
\theta_{x+1}
\end{bmatrix}
= 
\begin{bmatrix}
\alpha \\
\delta \\
\theta_x
\end{bmatrix}
+ \beta^{-1}
\begin{bmatrix}
a_{CA} - a_x \\
b_T - b_{TX} \\
b_R - b_{RX}
\end{bmatrix}
\] (9)

This procedure is repeated until convergence is achieved.
PROGRAM MAIN

PURPOSE: TO CONTROL THE SIMULATION OVERLAY SCHEME

SUBROUTINES SUPPORTED: NONE

SUBROUTINES REQUIRED: DATA SIMUL PRTS4

LOCAL SYMBOLS: IRUNX TOTAL NUMBER OF DATA CASES

IRUN DATA CASE COUNTER
SUBROUTINE MATIN

PURPOSE: TO COMPUTE THE INVERSE OF A MATRIX.

CALLING SEQUENCE: CALL MATIN(A,R,N)

ARGUMENTS
A(N,N) I MATRIX TO BE INVERTED
R(N,N) O RESULTANT INVERSE OF MATRIX A
N I DIMENSION OF A AND R

SUBROUTINES SUPPORTED: HYELS NAVM BIAIM POICOM GUIS
TARMAX GUID LUNTAR MULTAR

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:
AL A(1:1) + S (INTERMEDIATE VARIABLE)
ALBAR INTERMEDIATE VARIABLE
B INTERMEDIATE VECTOR
DETR INTERMEDIATE VECTOR
G INTERMEDIATE VECTOR
IX INTERMEDIATE VECTOR
JR DIMENSION OF A
MIXI INTERMEDIATE VARIABLE
MIXJ INTERMEDIATE VARIABLE
MIXL INTERMEDIATE VARIABLE
S INTERMEDIATE VARIABLE
X INTERMEDIATE VARIABLE
XOFF INTERMEDIATE VARIABLE

COMMON USED: EM7 EM9 ONE ZERO
SUBROUTINE MENU

PURPOSE: COMPUTE ASSUMED MEASUREMENT NOISE COVARIANCE MATRIX IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL MENU(MMCODE,ICODE)

ARGUMENTS:  
ICODE  I  INTERNAL CODE USED TO DISTINGUISH BETWEEN THE TWO ALTERNATIVES LISTED ABOVE

MMCODE  I  MEASUREMENT MODEL CODE

SUBROUTINES SUPPORTED: ERRANN

COMMON COMPUTED:  R

COMMON USED: IMNF MNCN
MEAN Analysis

The linearized observation equation employed by the navigation process is given by

$$\delta y_k = H_k^A \delta x_k^A + \eta_k$$

where $\delta y_k$ is the measurement deviation from the nominal measurement, $H_k^A$ is the augmented observation matrix, $\delta x_k^A$ is the augmented state deviation from the nominal augmented state, and $\eta_k$ is the assumed measurement noise.

The function of subroutine MEAN is to compute the measurement noise covariance matrix $R_k$ which describes the statistics of $\eta_k$. The constant variances for the measurement noises associated with all available measurement devices are stored in the vector MNMN. Subroutine MEAN selects the appropriate elements from this vector to construct the measurement noise covariance matrix $R_k$. 
MEMO Flow Chart

ENTER

Zero out $R_k$ matrix.

Only constant noise available at present. NO

Is measurement noise constant?

YES

RETURN

9

MHC0DE = ?

10, 11, 12, 13

A

1, 2, 3, 4, 5, 6, 7, 8

A

MHC0DE even?

NO

RETURN

YES

Compute the $R_k$ matrix for a range-rate measurement from the appropriate station.

Compute the $R_k$ matrix for a range-rate measurement from the appropriate station.

Compute the $R_k$ matrix for an apparent planet diameter measurement.

RETURN

MHC0DE = ?

10

11, 12, 13

Compute the $R_k$ matrix for the appropriate star-plane angle measurement.

Compute the $R_k$ matrix for the appropriate star-plane angle measurement.
SUBROUTINE MENOS

PURPOSE: COMPUTE ASSUMED AND ACTUAL MEASUREMENT NOISE COVARIANCE MATRICES IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL MENOS(MM CODE,ICODE)

ARGUMENTS: I CODE I INTERNAL CODE USED TO DISTINGUISH BETWEEN THE TWO ALTERNATIVES LISTED ABOVE

MM CODE I MEASUREMENT MODEL CODE

SUBROUTINES SUPPORTED: SIMULL

COMMON COMPUTED/USED: R

COMMON COMPUTED: AR

COMMON USED: AVARM IAMNF IMNF MNCN ZERO
MEMOS Analysis

The linearized observation equation employed by the navigation process is given by

\[ \delta Y_k = H_k^A \delta X_k^A + \eta_k \]

where \( \delta Y_k \) is the measurement deviation from the nominal measurement, \( H_k^A \) is the augmented observation matrix, \( \delta X_k^A \) is the augmented state deviation from the nominal augmented state, and \( \eta_k \) is the assumed measurement noise.

The actual measurement \( Y_k^a \) is given by

\[ Y_k^a = Y_k + b_k + \nu_k \]

where \( Y_k \) is the ideal measurement, which would be made in the absence of instrumentation errors, \( b_k \) is the actual measurement bias, and \( \nu_k \) represents the actual measurement noise.

Subroutine MENOS performs two functions. Its first function, which is identical to that of subroutine MENO, is to compute the measurement noise covariance matrix \( R_k \) which describes the statistics of noise \( \eta_k \). The constant variances for the assumed measurement noise associated with all available measurement devices are stored in the vector MNOCH. Subroutine MENOS selects the appropriate elements from this vector to construct the measurement noise covariance matrix \( R_k \).

The second function of MENOS is to compute the measurement noise covariance matrix \( R_k \) which describes the statistics of the actual noise \( \nu_k \). The constant variances for the actual measurement noises associated with all available measurement devices are stored in the vector AVARM. Subroutine AVARM selects the appropriate elements from this vector to construct the measurement noise covariance matrix \( R_k \).
**MENOS Flow Chart**

**ENTER**

**Is \( R_k \) or \( R_\ell \) to be computed?**

- **YES**
  - Set \( R_k \) to the previously computed \( R_k \).
  - RETURN

- **NO**
  - Zero out \( R_k \) matrix.
  - RETURN

**\( R_k \)**

- Compute \( R_k \). See MENOS flow chart for details.
- RETURN

**\( R_\ell = R_k? \)**

- **YES**
  - RETURN

- **NO**
  - Compute the \( R_k \) matrix for 3 star-planet angle measurements.
  - RETURN

**\( MMC\theta DZ = ? \)**

- **10, 11, 12, 13**
  - A
  - Compute the \( R_k \) matrix for a range-rate measurement from the appropriate station.

- **1, 2, 3, 4, 5, 6, 7, 8**
  - MMC\thetaDE even?
    - **YES**
      - Compute the \( R_k \) matrix for a range and range-rate measurement from the appropriate station.
      - RETURN
    - **NO**
      - Compute the \( R_k \) matrix for an apparent planet diameter measurement.

**A**

**\( MMC\theta D? = ? \)**

- **11, 12, 13**
  - Compute the \( R_k \) matrix for the appropriate star-planet angle measurement.
  - RETURN
SUBROUTINE MULCON

PURPOSE: TO PROPAGATE A SET OF CARTESIAN COORDINATES ALONG A LUNAR MULTI-COONIC TRAJECTORY OVER A SPECIFIED TIME INTERVAL.

CALLING SEQUENCE: CALL MULCON(SEI, TLI, TF, DT, SHF)

ARGUMENTS:
- SEI(6) I INITIAL SPACECRAFT GEOCENTRIC STATE
- TLI I INITIAL INJECTION TIME (JD EPOCH 1950)
- TF I TIME INTERVAL OF PROPAGATION
- DT I STEP SIZE USED IN MULTICONIC PROPAGATION
- SHF(6) O FINAL SPACECRAFT SELENOCENTRIC STATE

SUBROUTINES SUPPORTED: MULTAR
SUBROUTINES REQUIRED: CPROP EPH EM ORB

LOCAL SYMBOLS:
- ALNHQTH CONVERTS KILOMETERS TO ASTRONOMICAL UNITS
- A PERTURBING ACCELERATION VECTOR OF THE MOON OVER THE ITERATION INTERVAL
- COSEF COSINE OF TRUE ANOMALY ON SELENOCENTRIC CONIC AT END OF ITERATION INTERVAL
- DF FINAL TIME USED IN ITERATION INTERVAL
- DI INITIAL TIME USED IN ITERATION INTERVAL
- EMN MAGNITUDE OF FIRST THREE ELEMENTS OF EM
- EM GEOCENTRIC ECLIPTIC STATE OF MOON
- IDONE STOPPING CONDITION INDICATOR
  - =0 PROPAGATION CONTINUES
  - =1 STOPPING CONDITION REACHED
- I INDEX
- TIM JULIAN DATE OF FINAL TIME ON THE ITERATION INTERVAL
- TM CONVERTS SECONDS TO DAYS
- W SPACECRAFT VELOCITY VECTOR WITH RESPECT TO EARTH AND/OR MOON BEFORE AND AFTER LUNAR PERTURBATIONS AT DI AND DF.
<table>
<thead>
<tr>
<th>XM</th>
<th>MAGNITUDE OF THE X VECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>GEOCENTRIC POSITION OF SPACECRAFT AT DI AND GEOCENTRIC POSITION OF MOON AT DF</td>
</tr>
<tr>
<td>Y</td>
<td>GEOCENTRIC VELOCITY OF SPACECRAFT AT DI AND GEOCENTRIC VELOCITY OF MOON AT DF</td>
</tr>
<tr>
<td>Z</td>
<td>SPACECRAFT POSITION VECTOR WITH RESPECT TO EARTH, OR MOON AT DI AND DF BEFORE AND AFTER LUNAR PERTURBATIONS</td>
</tr>
</tbody>
</table>

COMMON COMPUTED: NO

COMMON USED: EHU TMU TMG XP ZERO
The equations of motion of a spacecraft traveling under the influence of the earth and moon may be written

\[ \ddot{r}_E = -\frac{\mu_E r_E}{r_E^3} - \frac{\mu_M r_M}{r_M^3} - \frac{\mu_M r_{EM}}{r_{EM}^3} \]  

(1)

where \( r_E, r_M, r_{EM} \) are the position vectors of the spacecraft-to-earth, the spacecraft-to-moon, and the moon-to-earth respectively and \( \mu_E, \mu_M \) are the gravitational constants of the earth and moon respectively.

The multi-conic approximation of the solution to (1) proceeds as follows. Let \( \vec{r}_{E,k}, \vec{v}_{E,k} \) be the geocentric state at some time \( t_k \). This state is propagated by conic formulae to obtain an estimate of the geocentric state at time \( t_{k+1} = t_k + \Delta t \) given by \( \vec{r}_{E,k+1}, \vec{v}_{E,k+1} \)

To account for the third term perturbations, the state of the moon relative to the earth at the two timepoints is computed, denoted by \( (\vec{r}_{EM,k}, \vec{v}_{EM,k}) \) and \( (\vec{r}_{EM,k+1}, \vec{v}_{EM,k+1}) \). The average value of this acceleration is then determined from

\[ \vec{a} = -\frac{\mu_M}{2} \left[ \frac{(\vec{r}_{EM,k} + \vec{r}_{EM,k+1})}{3} \right] \]  

(2)

The corrected geocentric state is then given by:

\[ \vec{r}_{E,k+1}'' = \vec{r}_{E,k+1}' + \frac{\vec{a}}{2} (\Delta t)^2 \]  

\[ \vec{v}_{E,k+1}'' = \vec{v}_{E,k+1}' + \vec{a} \Delta t \]  

(3)

The effect of the direct lunar perturbations is then added. The state of the spacecraft relative to the moon is then computed:

\[ \vec{r}_{M,k+1} = \vec{r}_{E,k+1} - \vec{r}_{EM,k+1} \]  

\[ \vec{v}_{M,k+1} = \vec{v}_{E,k+1} - \vec{v}_{EM,k+1} \]  

(4)
This state is then propagated linearly backwards in time over the time interval $\Delta t$ to obtain

$$\begin{align*}
\vec{r}_{N,k} &= \vec{r}_{N,k+1} - \vec{v}_{N,k+1} \Delta t \\
\vec{v}_{N,k} &= \vec{v}_{N,k+1}
\end{align*}$$  \hspace{1cm} (5)$$

This state is now propagated forward in a selenocentric conic to obtain a final state relative to the moon ($\vec{r}_{N,k+1}$, $\vec{v}_{N,k+1}$). The geocentric state of the spacecraft at time $t_{k+1}$ after considering all terms of (1) is then given by

$$\begin{align*}
\vec{r}_{E,k+1} &= \vec{r}_{N,k+1} + \vec{R}_{EN,k+1} \\
\vec{v}_{E,k+1} &= \vec{v}_{E,k} + \vec{v}_{EN,k+1}
\end{align*}$$  \hspace{1cm} (6)$$

This completes one cycle of the multi-conic propagation.

The multi-conic propagation proceeds until an input final time is reached or until the selenocentric conic passes through pericynthion.

SUBROUTINE  MULTAR

PURPOSE: TO CALCULATE THE TRANSLUNAR INJECTION CONDITIONS FROM
TARGETED PATCHED-CONIC CONDITIONS AND CALLS VMP TO
PERFORM THE NOMINAL TRAJECTORY NEEDED BY ITERAT.

CALLING SEQUENCE: CALL MULTAR

SUBROUTINES SUPPORTED: LUNA

SUBROUTINES REQUIRED: MULCON CAREL ELCAR EPHEM IMPACT
MATIN ORB PECEQ TIME

LOCAL SYMBOLS: AE  SEMI-MAJOR AXIS OF THE EARTH-ECLIPTIC,
TARGETED PATCHED-CONIC TRAJECTORY

ATRN  NOMINAL VALUES OF THE TARGET VARIABLES

ATAR  DESIRED VALUES OF THE TARGET VARIABLES

ATOL  TOLERANCES OF TARGET VARIABLES

BCOR  MAXIMUM STEPS ALLOWED IN ITERATIVE
CORRECTION OF CONTROL VARIABLES

BJ  ZERO TRUE ANOMALY USED TO DEFINE PERIGEE
OF THE TARGETED PATCHED-CONIC TRAJECTORY

BSTEP  MULTI-COMIC STEP SIZE (HOURS)

CHI  SENSITIVITY MATRIX RELATING PERTURBATIONS
IN CONTROL VARIABLES TO CHANGES IN TARGET
VARIABLES

DELP  VALUE USED TO PERTURB TLI FOR CONSTRUCTION
OF CHI

DELM  NOMINAL TIME FOR PROPAGATION

DELT  INTEGRATION TIME TO BE USED, AND TIME
ACTUALLY USED, IN THE MULTI-COMIC
PROPAGATION

DELV  VALUE USED TO PERTURB VELOCITY COMPONENTS
OF AT FOR CONSTRUCTION OF CHI

DV  CORRECTION ACTUALLY ADDED TO CONTROL
VARIABLES

ECEQP  TRANSFORMATION MATRIX FROM EARTH ECLIPTIC
TO LUNAR EQUATORIAL

ECEEQ  TRANSFORMATION MATRIX FROM EARTH ECLIPTIC
TO LUNAR EQUATORIAL
TO EARTH EQUATORIAL
EE  ECCENTRICITY OF THE EARTH-ECLIPTIC, TARGETED PATCHED-COMIC TRAJECTORY
ERR  ITERATE ERRORS IN TARGET CONDITIONS
FAC  INTERMEDIATE VARIABLE USED TO CHECK FOR MAXIMUM STEP
FNAE  INTERMEDIATE VARIABLE USED TO COMPUTE PHIA, PHIB, PHIC
HTT  HYPERBOLIC TIME TO LUNAR PERIAPSIS (DAYS)
IDATE  CALENDAR DATE OF INJECTION
IST  INDICATOR FOR CONTROL VARIABLE PERTURBATION
IT  ITERATIONS COUNTER
I  INDEX
JERTM  INDEX OF EARTH IN F-ARRAY
JMOON  INDEX OF MOON IN F-ARRAY
J  INDEX
K  INDEX
MITS  MAXIMUM NUMBER OF ITERATIONSALLOWED
INDEX  SAME AS JERTM
PERMN  MINIMUM PERTURBATION OF CONTROL VARIABLES FOR CONSTRUCTION OF CHI
PERMAX  MAXIMUM PERTURBATION OF CONTROL VARIABLES FOR CONSTRUCTION OF CHI
PERT  PERTURBATION VALUES USED TO CONSTRUCT CHI
PHIA  TRANSFORMATION MATRIX FROM RTN TO EC SYSTEM AT TLI
PHIB  TRANSFORMATION MATRIX FROM RTN TO EC SYSTEM AT PERTURBED TLI
PHIC  PRODUCT OF PHIB AND TRANSPOSE OF PHIA
PPE  DUMMY VARIABLE FOR CALL TO CAREL
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSI</td>
<td>TARGET MATRIX (INVERSE OF CHI) RELATING PERTURBATIONS IN TARGET VARIABLES TO CHANGES IN CONTROL VARIABLES</td>
</tr>
<tr>
<td>PTAR</td>
<td>PERTURBED TARGET VALUES</td>
</tr>
<tr>
<td>PV</td>
<td>PREDICTED CORRECTIONS TO CONTROL VARIABLES</td>
</tr>
<tr>
<td>QQE</td>
<td>DUMMY VARIABLE FOR CALL TO CAREL</td>
</tr>
<tr>
<td>REMAG</td>
<td>DUMMY VARIABLE FOR CALL TO ELCAR</td>
</tr>
<tr>
<td>REPET</td>
<td>MINIMUM ALLOWABLE INJECTION TIME DIFFERERENCE IN KTH AND K+2 ITERATIONS TO AVOID REPETITION-TRAP CORRECTION</td>
</tr>
<tr>
<td>RM</td>
<td>MAGNITUDE OF THE SCN POSITION VECTOR</td>
</tr>
<tr>
<td>RSE</td>
<td>INJECTION STATE IN EARTH EQUATORIAL SYSTEM AT TLI</td>
</tr>
<tr>
<td>RS</td>
<td>ROTATED INJECTION STATE FOR TIME DIFFERENTIAL</td>
</tr>
<tr>
<td>RT</td>
<td>INJECTION STATE USED IN PERTURBED MULTICONIC PROPAGATIONS</td>
</tr>
<tr>
<td>SCN</td>
<td>FINAL STATE IN 'UNAR ECLIPTIC SYSTEM ON THE MULTI-CONIC</td>
</tr>
<tr>
<td>SEC</td>
<td>SECONDS OF CALENDAR DATE OF TLI</td>
</tr>
<tr>
<td>STEP</td>
<td>MULTI-CONIC STEP SIZE (SECONDS)</td>
</tr>
<tr>
<td>STLII</td>
<td>ORIGINAL VALUE OF TLI, RESTORED FOR SUCESSIVE ITERATIONS</td>
</tr>
<tr>
<td>TAE</td>
<td>TRUE ANOMALY OF EARTH-ECLIPTIC TARGETED PATCHED-CONIC TRAJECTORY</td>
</tr>
<tr>
<td>TBR</td>
<td>DUMMY VARIABLE FOR CALL TO IMPACT</td>
</tr>
<tr>
<td>TBT</td>
<td>DUMMY VARIABLE FOR CALL TO IMPACT</td>
</tr>
<tr>
<td>TFP</td>
<td>TIME OF FLIGHT FROM PERIGEE OF THE E*TH-ECLIPTIC, TARGETED PATCHED-CONIC TRAJECTORY</td>
</tr>
<tr>
<td>TIMM1</td>
<td>INJECTION DATE ON K-1 ITERATION</td>
</tr>
<tr>
<td>TIMM2</td>
<td>INJECTION DATE ON K-2 ITERATION</td>
</tr>
<tr>
<td>COMMON COMPUTED/USED</td>
<td>NO</td>
</tr>
<tr>
<td>----------------------</td>
<td>----</td>
</tr>
<tr>
<td>COMMON COMPUTED</td>
<td>SQORD</td>
</tr>
<tr>
<td>COMMON USED</td>
<td>ALNGTH</td>
</tr>
<tr>
<td></td>
<td>KUR</td>
</tr>
<tr>
<td></td>
<td>SHA</td>
</tr>
<tr>
<td></td>
<td>TM</td>
</tr>
</tbody>
</table>
MULTAR Analysis

Let the earth equatorial state of the probe at the LSI as computed from the patched conic targeting be denoted $\mathbf{T}_{LS}$, $\mathbf{V}_{LS}$. Subroutine CAREL is called to compute the conic elements and conic time from perigee $\Delta t$ based on the geocentric conic. The time of injection is then computed as

$$t_{TIL} = t_{SL} - \Delta t \quad (1)$$

The position and velocity of the probe at $t_{TIL}$ is given by the state along the conic at perigee (true anomaly of zero) and determined by ELCAR to be $\mathbf{T}_{TIL}$, $\mathbf{V}_{TIL}$. If $\Phi_{ECEQ}$ is the transformation matrix from the EC (earth ecliptic) to the EQ (earth equatorial) system, then the patched conic injection state in EC coordinates is

$$\mathbf{T}_{I} = \Phi_{ECEQ}^T \mathbf{T}_{TIL}$$
$$\mathbf{V}_{I} = \Phi_{ECEQ}^T \mathbf{V}_{TIL} \quad (2)$$

Since the earth is revolving about the E-M barycenter in time, the EC injection state must be rotated if an earlier or later injection time is to be used. The necessary rotation matrix may be easily computed through the introduction of the R-T-U coordinate system. Let the state of the earth at some time $t_k$ in EC (barycentric ecliptic) coordinates be denoted $\mathbf{R}_k$, $\mathbf{V}_k$. Construct the $\mathbf{R}$-$\mathbf{T}$-$\mathbf{U}$ system at that point as

$$\mathbf{R}_k = \frac{\mathbf{R}_k}{R_k} \quad \mathbf{T}_k = \frac{\mathbf{R}_k \times \mathbf{V}_k}{|\mathbf{R}_k \times \mathbf{V}_k|} \quad \mathbf{U}_k = \mathbf{R}_k \times \mathbf{T}_k \quad (3)$$

The transformation matrix from the $\mathbf{R}_k$-$\mathbf{T}_k$-$\mathbf{U}_k$ system to the ecliptic system to the ecliptic system is then given by

$$\Phi_k = \left[ \begin{array}{c|c|c} \mathbf{R}_k & \mathbf{T}_k & \mathbf{U}_k \end{array} \right] \quad (4)$$

At a time $t_{k+1}$ the state of the earth in EC coordinates is given by $\mathbf{R}_{k+1}$, $\mathbf{V}_{k+1}$ and the transformation from the $\mathbf{R}_{k+1}$-$\mathbf{T}_{k+1}$-$\mathbf{U}_{k+1}$ system to ecliptic coordinates is given by $\Phi_{k+1}$ in accordance with (4). Injection
states at times \( t_k \) and \( t_{k+1} \) will be called "equivalent" if they are identical when expressed in the pertinent \( \hat{\mathbf{X}}-\hat{\mathbf{Y}}-\hat{\mathbf{Z}} \) system. Therefore if \( (\tilde{\mathbf{r}}_k, \tilde{\mathbf{v}}_k) \) is the injection state in EC coordinates at time \( t_k \), the equivalent state in EC coordinates at \( t_{k+1} \) is given by

\[
\begin{bmatrix}
\tilde{r}_{k+1} \\
\tilde{v}_{k+1}
\end{bmatrix} =
\begin{bmatrix}
\Psi_{k+1,k} & 0 \\
0 & \Psi_{k+1,k}
\end{bmatrix}
\begin{bmatrix}
\tilde{r}_k \\
\tilde{v}_k
\end{bmatrix}
\]

(5)

where the rotation matrix \( \Psi \) is defined by

\[
\Psi_{k+1,k} = \Phi_{k+1} \Phi_k^T
\]

(6)

The targeting algorithm used by MULTAR may now be described. Let the injection state in EC coordinates on the \( k \)-th iteration be denoted \( (t_k, \tilde{r}_k, \tilde{v}_k) \). This state is propagated forward using the multi-conic propagator MULTIN to determine a final state \( \tilde{r}_M, \tilde{v}_M \) to the moon in ecliptic coordinates. IMPACT is then called to compute the \( B \cdot T_k, B \cdot R_k \) and \( t_{CA,k} \) actually achieved on the trajectory and the target values of \( B^* \cdot T_k, B^* \cdot R_k \) required to satisfy the \( i_{CA} \) and \( r_{CA} \) constraints. The semi-major axis \( a_k \) of the \( k \)-th iterate is computed from the conic formula

\[
a = r_M \left( 2 - \frac{r_M v_M^2}{\mu_M} \right)^{-1}
\]

(7)

Errors in the four target conditions

\[
\Delta \mathbf{T} = \begin{bmatrix}
\Delta a \\
\Delta B \cdot T \\
\Delta B \cdot R \\
\Delta t_{CA}
\end{bmatrix} = \begin{bmatrix}
a - a^* \\
B \cdot T_k - B^* \cdot T_k \\
B \cdot R_k - B^* \cdot R_k \\
t_{CA,k} - t_{CA}
\end{bmatrix}
\]

(8)

if the error in each parameter is less than the allowable tolerance, the process stops.
If convergence has not been achieved a Newton-Raphson iteration is entered. The four controls are $\vec{v}_{x_k}, \vec{v}_{y_k}, \vec{v}_{z_k}$, and $t_{k}$. For the velocity components $\vec{v}_x, \vec{v}_y, \vec{v}_z$, a perturbation $\Delta v$ is added to the pertinent component while the rest of the injection state is held constant before propagating with the multi-conic. For the time perturbation, the rotation matrix $\Psi_{\theta}$ corresponding to the perturbed time $t_{k} + \Delta t$ (6) is first computed. The injection state used in the perturbed propagation for time is then $[t_{k} + \Delta t, \Psi_{\theta} \vec{v}_{k}, \Psi_{\theta} \vec{v}_{k}]$.

A sensitivity matrix is computed using the results of the numerical differencing:

$$
X = \begin{bmatrix}
\frac{\Delta a_x}{\Delta v_x} & \frac{\Delta a_y}{\Delta v_y} & \cdots \\
\frac{\Delta B \vec{v}_x}{\Delta v_x} \\
\frac{\Delta B \vec{v}_y}{\Delta v_y} \\
\frac{\Delta B \vec{v}_z}{\Delta v_z} \\
\frac{\Delta \bar{t}_{CA_k}}{\Delta v_x} & \frac{\Delta \bar{t}_{CA_k}}{\Delta t}
\end{bmatrix}
$$

(9)

where in the term $\frac{\Delta a_{\beta}}{\Delta \beta}$, $\Delta a_{\beta}$ is the change in the $\alpha$ target parameter produced by the variation of the $\beta$ control component and $\Delta \beta$ is the change in the $\beta$ control component. The $k+1$ iterate controls are then given by

$$
\Delta c = \begin{bmatrix}
\delta v_x \\
\delta v_y \\
\delta v_z \\
\delta t
\end{bmatrix} = X^{-1} \Delta \tau
$$

(10)

The $k+1$ injection state is then computed by first determining the injection state after rotation due to the change in injection time and then adding the injection velocity corrections

$$
t_{k+1} = t_k + \delta t
$$

$$
\vec{v}_{k+1} = \Psi_{\theta} \vec{v}_k + \delta \vec{v}
$$

(11)

The iteration process is repeated until tolerable errors are met. The converged injection state is then integrated in the virtual mass trajectory.
SUBROUTINE MUND

PURPOSE: TO COMPUTE THE AUGMENTED PORTION OF THE STATE TRANSITION MATRIX WHEN THE GRAVITATIONAL CONSTANT OF THE SUN OR OF THE TARGET PLANET HAS BEEN AUGMENTED TO THE BASIC STATE VECTOR.

CALLING SEQUENCE: CALL MUND(RI,RF,POSS)

ARGUMENTS:
RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL
RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL
POSS I DISTANCE OF THE VEHICLE FROM THE TARGET PLANET AT THE INITIAL TIME

SUBROUTINES SUPPORTED: PSIM
SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS:
IC COUNTER FOR VARIABLES AUGMENTED TO STATE VECTOR
IPR TEMPORARY STORAGE FOR IPRINT
RPER ALTERED POSITION AND VELOCITY OF VEHICLE AT FINAL TIME
SAVE TEMPORARY STORAGE LOCATION FOR GRAVITATIONAL CONSTANTS OF SUN AND TARGET PLANET

COMMON COMPUTED/USED: IPRINT PMASS
COMMON COMPUTED: TXU TXXS
COMMON USED: ALNGTH DELMUP DELMUS I AUGDC I AUGIN
I AUG NTMC NTP SPHERE TM
MUND Analysis

The nonlinear equations of motion of the spacecraft can be written symbolically as

\[ \ddot{x} = \mathbf{f}(x, \mu, t) \]  

(1)

where \( \mathbf{x} \) is the spacecraft position/velocity state and \( \mathbf{\mu} \) is a vector composed of the gravitational constants of the Sun and the target planet.

Suppose we wish to use numerical differencing to compute those columns of \( \mathbf{\Theta}_{xx} \) and \( \mathbf{\Theta}_{xx} \) associated with gravitational constant biases included in the augmented state vector over the time interval \([t_{k-1}, t_k]\). Let \( \mathbf{\theta}_j(t_k, t_{k-1}) \) represent the column associated with the j-th gravitational constant bias.

We assume we have available the nominal states \( \mathbf{x}^*(t_{k-1}) \) and \( \mathbf{x}^*(t_k) \), which, of course, were obtained by numerically solving equation (1) using nominal \( \mu \). To obtain \( \mathbf{\theta}_j(t_k, t_{k-1}) \) we increment the j-th gravitational constant bias by the pertinent numerical differencing factor \( \Delta \mu_j \) and numerically integrate equation (1) over the interval \([t_{k-1}, t_k]\) to obtain the new spacecraft state \( \mathbf{x}_j(t_k) \), where the j-subscript on the spacecraft state indicates that it was obtained by incrementing the j-th gravitational constant bias. Then

\[ \mathbf{\theta}_j(t_k, t_{k-1}) = \frac{\mathbf{x}_j(t_k) - \mathbf{x}^*(t_k)}{\Delta \mu_j} \]

(2)
MUND Flow Chart

ENTER

Is $\delta \mu_s$ in $\tilde{x}_s$ or $\tilde{u}$?

NO $\rightarrow$ A

YES

Increment $\mu_s$ after saving its original value.

Call NTM to compute the state at $t_k$ resulting from the incrementation of $\mu_s$ at $t_{k-1}$.

Reset $\mu_s$ to its original value.

$\tilde{x}_s$

Is $\delta \mu_s$ in $\tilde{x}_s$ or $\tilde{u}$?

$\tilde{u}$

Compute the appropriate column of $\theta_{xx_s}$ using equation (2).

Compute the appropriate column of $\theta_{xu}$ using equation (2).

A
Is $\delta \mu_p$ in $\bar{x}_s$ or $\bar{u}$?

Is spacecraft within six times the target planet sphere of influence?

Yes

Increment $\mu_p$ after saving its original value.

Call NIM to compute the state at $t_x$ resulting from the incrementation of $\mu_p$ at $t_{k-1}$.

Reset $\mu_p$ to its original value.

$\bar{x}_s$

$\bar{u}$

Is $\delta \mu_p$ in $\bar{x}_s$ or $\bar{u}$?

Compute the appropriate column of $Q_{x}$ using equation (2).

Compute the appropriate column of $Q_{x}$ using equation (2).

RETURN
SUBROUTINE NAVM

PURPOSE: TO PROPAGATE COVARIANCE MATRIX PARTITIONS P, CXXS, CXU, CXV, PS, CXSU, CXSV FROM THE TIME OF THE LAST MEASUREMENT OR EVENT TO THE PRESENT TIME USING A CONSIDER RECURSIVE ALGORITHM.

CALLING SEQUENCE: CALL NAVM(NR,ICODE)

ARGUMENT: ICODE I INTERNAL CODE WHICH DETERMINES IF A MEASUREMENT IS BEING PROCESSED

NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX

SUBROUTINES SUPPORTED: SIMULL SETEVS GUISIM PRE5.M ERRANN SETEVN GUIDM PRED

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS AJ MEASUREMENT RESIDUAL COVARIANCE MATRIX AND ITS INVERSE

AKW INTERMEDIATE ARRAY

DUM INTERMEDIATE VECTOR

PSAVE INTERMEDIATE ARRAY

SUM INTERMEDIATE VARIABLE

SW INTERMEDIATE ARRAY

COMMON COMPUTED/USED: AK CXXSUP CXSU CXSVP CXSV CXUP CXU CXVP CXV CXXSP CXXS PP PSP PS P

COMMON USED: AL AM G H NDIM1 NDIM2 NDIM3 ONE PHI Q R TXU TXXS UD VO ZERO

366
NAVH Analysis

The augmented deviation state vector is defined as
\[ \bar{x}^A = [x, \dot{x}_s, \mathcal{U}, \mathcal{V}]^T \]

where
- \( x \) = position and velocity state (dimension 6)
- \( \dot{x}_s \) = solve-for parameter state (dimension \( n_1 \))
- \( \mathcal{U} \) = dynamic consider parameter state (dimension \( n_2 \))
- \( \mathcal{V} \) = measurement consider parameter state (dimension \( n_3 \))

The linearized equations of motion have form
\[ \begin{align*}
\dot{x} &= f_x \bar{x} + f_{x_\xi} \dot{x}_s + f_{x_\mathcal{U}} \mathcal{U} \\
\dot{x}_s &= 0 \\
\dot{\mathcal{U}} &= 0 \\
\dot{\mathcal{V}} &= 0
\end{align*} \]

and solution
\[ \begin{align*}
\bar{x}_{k+1} &= \Phi(k+1,k)\bar{x}_k + \Theta_{xx}(k+1,k)\dot{x}_s_k + \Theta_{x\mathcal{U}}(k+1,k)\mathcal{U}_k + \mathcal{U}_k \\
\dot{x}_{s,k+1} &= \dot{x}_{s,k} \\
\mathcal{U}_{k+1} &= \mathcal{U}_k \\
\mathcal{V}_{k+1} &= \mathcal{V}_k
\end{align*} \]

where dynamic noise \( \mathcal{U}_k \) has been added to the solution of \( \bar{x}_{k+1} \). This solution can be written in augmented form
\[ \bar{x}^A_{k+1} = \Phi^A(k+1,k) \bar{x}^A_k + \mathcal{U}^A_k \]

where the augmented state transition matrix \( \Phi^A(k+1,k) \) is defined as
Henceforth state transition matrix partitions will be written without stating the associated interval of time, which will always be assumed to be \([k, k+1]\).

The measurement deviation vector \(\bar{y}_k\) (dimension \(m\)) is related to the augmented deviation state vector through the equation

\[
\bar{y}_k = H_k^A \hat{x}_k + \bar{\eta}_k
\]

where the augmented observation matrix is defined as

\[
H_k^A = 
\begin{bmatrix}
H_k & M_k & C_k & L_k
\end{bmatrix}
\]

and \(\bar{\eta}_k\) is measurement noise.

The augmented state covariance matrix \(P_k^A\) can be written in terms of its partitions as

\[
P_k^A = 
\begin{bmatrix}
P_k & C_{xx_k} & C_{xu_k} & C_{xv_k} \\
C_{xx_k}^T & P_{\bar{u}_k} & C_{x\bar{u}_k} & C_{x\bar{v}_k} \\
C_{xu_k}^T & C_{x\bar{u}_k}^T & U_0 & C_{uv_k} \\
C_{xv_k}^T & C_{x\bar{v}_k}^T & C_{uv_k}^T & V_0
\end{bmatrix}
\]
Prediction and filtering equations for the partitions appearing in the previous equation will be written below. Equations need not be written for the consider parameter covariances $U_0$ and $V_0$ since these do not change with time. Also, $C_{uv}$ will be set to zero because of the assumption that no cross-correlation exists between dynamic and measurement consider parameters. In the equations below $Q$ and $R$ represent the covariances of the dynamic and measurement noises, respectively, defined previously. A minus superscript on covariance partitions indicates the covariance partition immediately prior to processing a measurement; a plus superscript, immediately after processing a measurement. If $IC\&DE$ indicates that a measurement is not to be processed, the filtering equations are bypassed. To improve numerical accuracy and avoid non-positive definite covariance matrices, $P^-$, $P^+_s$, $P^+_x$, $P^+_a$, $J$, and $J^{-1}$ are always symmetrized after their computation.

Prediction equations:

$$P_{k+1}^- = (\Phi P_k^- + \Theta_{sx} C_{xx}^+ - \Theta_{xu} C_{xu}^T) \Phi^T + C_{xx}^- + Q_k$$
$$C_{xx}^-_{k+1} = \Phi C_{xx}^-_k + \Theta_{sx} P^-_k + \Theta_{xu} C_{xu}^- u_k$$
$$P^+_{k+1} = P^+_k$$
$$C_{xx}^+_{k+1} = \Phi C_{xx}^+ k + \Theta_{sx} C_{xu}^+ + \Theta_{xu} u_0$$
$$C_{xu}^-_{k+1} = C_{xu}^- k$$
$$C_{xv}^-_{k+1} = \Phi C_{xv}^- k + \Theta_{sx} C_{xv}^- v_k$$
$$C_{xv}^+_{k+1} = C_{xv}^+ k$$
Filtering equations:

\[ J_{k+1} = H_{k+1} A_{k+1} + M_{k+1} B_{k+1} + G_{k+1} D_{k+1} + L_{k+1} E_{k+1} + R_{k+1} \]

where

\[ A_{k+1} = P_{k+1}^{-} H_{k+1}^{T} + C_{xx}^{*-} M_{k+1} + C_{xx}^{*-} G_{k+1}^{T} + C_{xx}^{*-} L_{k+1}^{T} \]

\[ B_{k+1} = P_{k+1}^{-} H_{k+1}^{T} + C_{xx}^{*-} H_{k+1}^{T} + C_{xx}^{*-} G_{k+1}^{T} + C_{xx}^{*-} L_{k+1}^{T} \]

\[ D_{k+1} = C_{xu}^{*-} H_{k+1}^{T} + C_{xu}^{*-} M_{k+1} + U_{o} G_{k+1}^{T} \]

\[ E_{k+1} = C_{xv}^{*-} H_{k+1}^{T} + C_{xv}^{*-} M_{k+1} + V_{o} L_{k+1}^{T} \]

The Kalman gain matrices for both position/velocity state and solve-for parameters are given by

\[ K_{k+1} = A_{k+1} J_{k+1}^{-1} \]

\[ S_{k+1} = R_{k+1} J_{k+1}^{-1} \]

The covariance partitions immediately after processing a measurement are given by

\[ P_{k+1}^{+} = P_{k+1}^{-} - K_{k+1} A_{k+1}^{T} \]

\[ C_{xx}^{+} = C_{xx}^{-} - K_{k+1} B_{k+1}^{T} \]

\[ P_{s}^{+} = P_{s}^{-} - E_{k+1} B_{k+1}^{T} \]

\[ C_{xu}^{+} = C_{xu}^{-} - K_{k+1} D_{k+1}^{T} \]

\[ C_{xv}^{+} = C_{xv}^{-} - K_{k+1} D_{k+1}^{T} \]
\[
C^+_{x,x} = C^-_{x,x} - S_{k+1} D^T_{k+1} \\
C^+_{x,v} = C^-_{x,v} - X_{k+1} E^T_{k+1} \\
C^+_{x,v} = C^-_{x,v} - S_{k+1} E^T_{k+1}
\]

Compute covariance matrix partitions at $t_{k+1}$ using prediction equations:

\[ P_{k+1}^{-} C_x \quad C_u \quad C_{xv} \]

\[ s_{k+1}^{-} \quad x_u \quad x_{v} \quad x_{v} \]

Symmetrize $P_{k+1}^{-}$

Is a measurement to be processed?

NO

Update covariance matrix partitions at $t_{k+1}^{+}$ by equating them to the previously computed covariance matrix partition.

YES

Compute measurement residual covariance matrix $J_{k+1}$ and symmetrize.

Compute Kalman gain matrices $K_{k+1}$ and $S_{k+1}$.

Update covariance matrix partitions at $t_{k+1}^{+}$ using filtering equations.

Symmetrize $P_{k+1}^{+}$ and $S_{k+1}$.

RETURN
SUBROUTINE NDTH

PURPOSE: TO COMPUTE THE UNAUGMENTED PORTION OF THE STATE TRANSITION MATRIX USING THE NUMERICAL DIFFERENCE TECHNIQUE.

CALLING SEQUENCE: CALL NDTH(RI,RF)

ARGUMENTS:
RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL

RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: NTM

LOCAL SYMBOLS:
F1 TEMPORARY STORAGE FOR FACP
F2 TEMPORARY STORAGE FOR FACV
IPR INTERMEDIATE STORAGE FOR IPRINT
RP POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
SAVE TEMPORARY STORAGE FOR ACC
T ALTERED POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
U ALTERED POSITION AND VELOCITY OF VEHICLE AT FINAL TIME

COMMON COMPUTED/USED: ACC FACP FACV IPRINT

COMMON COMPUTED: PHI

COMMON USED: ACCMD DELTM MDACC ONE ZERO
NDIM Analysis

The nonlinear equations of motion of the spacecraft can be written symbolically as

\[ \dot{\bar{x}} = \bar{f}(\bar{x}, t) \]  

where \( \bar{x} \) is the spacecraft position/velocity state.

Suppose we wish to use numerical differencing to compute the state transition matrix \( \Phi(t_k, t_{k-1}) \). Let \( \Phi_j(t_k, t_{k-1}) \) represent the j-th column of \( \Phi_j(t_k, t_{k-1}) \). We assume we have available the nominal states \( \bar{x}^*(t_{k-1}) \) and \( \bar{x}^*(t_k) \). To obtain \( \Phi_j(t_k, t_{k-1}) \) we increment the j-th element of \( \bar{x}^*(t_{k-1}) \) by the numerical differencing factor \( \Delta x_j \) and numerically integrate equation (1) over the time interval \([t_{k-1}, t_k]\) to obtain the new spacecraft state \( \bar{x}^*_j(t_k) \). The j-subscript indicates \( \bar{x}^*_j(t_k) \) was obtained by incrementing the j-th element of \( \bar{x}^*(t_{k-1}) \).

Then

\[ \Phi_j(t_k, t_{k-1}) \]  

\[ \frac{\bar{x}^*_j(t_k) - \bar{x}^*(t_k)}{\Delta x_j} \]  

\[ j = 1, 2, \ldots, 6 \]
MDTM Flow Chart

ENTER

Is $\Delta t > 1 \times 10^{-8}$?

NO

YES

Set $\Phi = 1$.

RETURN

Select numerical differencing accuracy.

Call NTM to compute the state at the end of the time interval $[t_{k-1}, t_k]$.

Set column index $N$ to 1.

Save nominal values of numerical differencing factors.

Multiply nominal numerical differencing factors by 10 if $\Delta t < 10$. Divide factors by 10 if $\Delta t > 100$.

Increment the $N$-th element of the state at $t_{k-1}$ by the appropriate numerical differencing factor.

Call NTM to compute the state at $t_k$. 

A

B
Compute the N-th column of $\Phi$ using equation (2).

Increment column index N.

$N > 6$

YES

Reset saved value.

RETURN
SUBROUTINE NEWPGE

PURPOSE: PRINTS APPROPRIATE HEADING AT THE TOP OF EACH PAGE WHEN PRINTOUT OF TRAJECTORY INFORMATION IS DESIRED

CALLING SEQUENCE: CALL NEWPGE

SUBROUTINES SUPPORTED: INPUTZ PRINT SPACE

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: IPG

COMMON COMPUTED: LINCT

COMMON USED: KL
PROGRAM   NOMINAL

PURPOSE: TO CONTROL THE ENTIRE GENERATION OF A NOMINAL TRAJECTORY FROM INJECTION TARGETING THROUGH MIDCOURSE CORRECTIONS AND ORBIT INSERTION.

CALLING SEQUENCE: NONE (MAIN PROGRAM)

SUBROUTINES SUPPORTED: NONE (MAIN PROGRAM)

SUBROUTINES REQUIRED: GIDANS PRELIM TRJTRY

COMMON COMPUTED: IPRE

COMMON USED: KWIT
NORMAL Analysis

NORMAL is the executive program controlling the entire generation of a nominal trajectory from injection targeting through midcourse corrections and orbit insertion.

NORMAL begins by calling PRELIM for the preliminary work including initialization of variables, reading of the input data, and computation of zero iterate values of initial time, position, and velocity if required.

NORMAL then calls TRJTRY. TRJTRY first determines the time of the next guidance event. It then integrates and records the nominal trajectory to that time. TRJTRY then returns control to NORMAL.

NORMAL next calls GIDANS. GIDANS processes the computation and execution of the current guidance event. NORMAL then reenters its basic cycle by calling TRJTRY to propagate the corrected trajectory to its next guidance event.

Two flags are used by NORMAL. The flag IPRE is initialized at zero in NORMAL. During the processing of the first data case PRELIM sets it to unity. PRELIM uses IPRE to determine whether to preset constants to internally stored values or leave them at their previous values before reading the next data case.

The second flag KQIT determines whether the current case should be continued or terminated according to the flag value zero or unity respectively. Termination is indicated when a fatal error occurs during trajectory propagation or guidance event computation or when the desired end time is reached.
**NORMAL Flow Chart**

1. **ENTER**
2. **IPRE = 0**
3. **A**
   - **Call PRELIM**
4. **B**
   - **Call TRJTRY**
   - **KWIT=?**
     - **KWIT=1**
     - **=0**
     - **Call GIDANS**
   - **=0**
5. **KWIT=1**
   - **=0**
   - **A**
   - **B**
SUBROUTINE NONINS

PURPOSE: TO DETERMINE THE TIME AND CORRECTION V... FOR FOR AN
INSERTION FROM AN APPROXIM HYPERBOLA INTO A SPECIFIED
PLANE AND AS NEAR AS POSSIBLE TO A PRESCRIBED CLOSED
ORBIT.

CALLING SEQUENCE: CALL NONINS(GM,X,Z,DA,DE,DWTP,DI,DN,TMEX,VEL,
IEX)

ARGUMENTS: GM I GRAVITATIONAL CONSTANT
X(3) I POSITION VECTOR AT DECISION
Z(3) I VELOCITY VECTOR AT DECISION
DA I DESIRED SEMIMAJOR AXIS
DE I DESIRED ECCENTRICITY
DWTP I DESIRED ARGUMENT OF PRRIAPSIS
DWTP I DESIRED ARGUMENT OF PERIAPSIS
DI I DESIRED INCLINATION
DN I DESIRED LONGITUDE OF ASCENDING NODE
TMEX O TIME FROM DECISION TO EXECUTION (SECONDS)
VEL(3) O INSERTION VELOCITY VECTOR
IEX O EXECUTION CODE
=0 EXECUTABLE SOLUTION DETERMINED
=1 NO EXECUTABLE SOLUTION FOUND

SUBROUTINES SUPPORTED: INSERS
SUBROUTINES REQUIRED: CARCL ELCAR
LOCAL SYMBOLS: AH HYPERBOLIC SEMIMAJOR AXIS
ANG TRUE ANOMALY OF HYPERBOLIC ASYMPOTOTE
ARC2 360.
ARC 180.
A SEMIMAJOR AXIS OF MODIFIED ELLIPSE
CEI COSINE OF DI
CEN COSINE OF DN
CMI  COSINE OF MI
CHN  COSINE OF MN
CTAE COSINE OF ETA
CTASY COSINE OF TASY
CMTXE COSINE OF MTXE
CMTXM COSINE OF MTXM
DELV VELOCITY CORRECTIONS OF CANDIDATE SOLUTION
DRA  DESIRED APOAPSIS RADIUS
DRP  DESIRED PERIAPSIS RADIUS
DTA  DUMMY VARIABLE FOR OUTPUT
DVM  MAGNITUDES OF CANDIDATE CORRECTIONS
DV   MAGNITUDES OF CANDIDATE CORRECTIONS
EH   HYPERBOLIC ECCENTRICITY
ERRMAX SCALAR ERROR ASSIGNED TO IMPOSSIBLE SOLUTION
ERR  ARRAY OF SCALAR ERRORS OF SOLUTIONS
ETAX TRUE ANOMALIES AT INTERSECTION POINTS ON ELLIPSE
E    ECCENTRICITY OF MODIFIED ELLIPSE
HI   HYPERBOLIC INCLINATION
HN   HYPERBOLIC LONGITUDE OF ASCENDING NODE
HRP  HYPERBOLIC PERIAPSIS RADIUS
MTAX TRUE ANOMALIES AT INTERSECTION POINTS ON HYPERBOLA
MTA  CANDIDATE HYPERBOLIC TRUE ANOMALY AT INTERSECTION
MININ INDEX OF OPTIMAL INTERSECTION POINT
MIN  INDEX OF OPTIMAL SOLUTION (POINT AND MOD)
NCPOS FLAG INDICATING WHETHER ANGLE BETWEEN NODE
AND INTERSECTION IS GREATER OR LESS THAN
180
NWOLS NUMBER OF SOLUTIONS
NT1 INDEX OF FIRST SOLUTION
NT2 INDEX OF LAST SOLUTION
PM HYPERBOLIC SEMILATUS RECTUM
PI THE MATHEMATICAL CONSTANT PI
PP UNIT VECTOR TOWARD PERIAPSISS
QQ UNIT VECTOR IN ORBITAL PLANE NORMAL TO PP
RAD DEGREE TO RADIANT FACTOR
RA APOAPSISS RADIUS
RHYP HYPERBOLIC CANDIDATE RADII TO INTERSECTION
RMAG RADIUS TO INTERSECTION POINT
RM RADIUS AT DECISION
RP PERIAPSISS RADIUS
RX RADIUS TO INTERSECTION POINT ON ELLIPSE
RI RADIUS VECTOR TO HYPERBOLA AT INTERSECTION
R RADIUS VECTOR TO ELLIPSE AT INTERSECTION
SEI SINE OF DI
SIN SINE OF DN
SGHZ SIGN OF DECLINATION OF INTERSECTION POINT
SHI SINE OF H1
OSH SINE OF OH
STA TRUE ANOMALY AT DECISION
TAE ARRAY OF ELLIPTIC TRUE ANOMALIES
TASY TRUE ANOMALY OF ASYMPTOTE
TAXE TRUE ANOMALY AT INTERSECTION POINT ON
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{ELLIPSE}$</td>
<td>TRUE ANOMALY AT INTERSECTION POINT ON HYPERBOLA</td>
</tr>
<tr>
<td>$\text{TAXH}$</td>
<td>TRUE ANOMALY</td>
</tr>
<tr>
<td>$\text{TDEC}$</td>
<td>TIME FROM PERIAPSIS AT DECISION</td>
</tr>
<tr>
<td>$\text{TEXC}$</td>
<td>TIME FROM PERIAPSIS AT EXECUTION ON HYPERBOLA</td>
</tr>
<tr>
<td>$\text{TEX}$</td>
<td>ARRAY OF CANDIDATE TIMES FROM DECISION TO EXECUTION</td>
</tr>
<tr>
<td>$\text{TTF}$</td>
<td>TIME FROM PERIAPSIS AT DECISION ON ELLIPSE</td>
</tr>
<tr>
<td>$\text{VM}$</td>
<td>SPEED AT DECISION</td>
</tr>
<tr>
<td>$\text{V1}$</td>
<td>VELOCITY VECTOR ON HYPERBOLA AT EXECUTION</td>
</tr>
<tr>
<td>$\text{V}$</td>
<td>VELOCITY VECTOR ON ELLIPSE AT EXECUTION</td>
</tr>
<tr>
<td>$\text{ME}$</td>
<td>ARGUMENT OF PERIAPSIS ON ELLIPSE</td>
</tr>
<tr>
<td>$\text{MNM}$</td>
<td>ARGUMENT OF PERIAPSIS ON HYPERBOLA</td>
</tr>
<tr>
<td>$\text{NTXE}$</td>
<td>ANGLE BETWEEN ASCENDING NODE AND INTERSECTION POINT ON ELLIPSE</td>
</tr>
<tr>
<td>$\text{NTXH}$</td>
<td>ANGLE BETWEEN ASCENDING NODE AND INTERSECTION POINT ON HYPERBOLA</td>
</tr>
<tr>
<td>$\text{WN}$</td>
<td>UNIT NORMAL TO ORBITAL PLANE</td>
</tr>
<tr>
<td>$\text{W}$</td>
<td>ARGUMENT OF PERIAPSIS</td>
</tr>
<tr>
<td>$\text{XINT}$</td>
<td>X COMPONENT OF INTERSECTION POINT</td>
</tr>
<tr>
<td>$\text{YINT}$</td>
<td>Y COMPONENT OF INTERSECTION POINT</td>
</tr>
<tr>
<td>$\text{ZINT}$</td>
<td>Z COMPONENT OF INTERSECTION POINT</td>
</tr>
</tbody>
</table>
NOMINS Analysis

NOMINS determines the time and correction vector for an impulsive insertion from an approach hyperbola into a specified plane and as near as possible to a prescribed closed orbit. The approach hyperbola is specified by giving the planetocentric equatorial state \( T, \mathbf{V} \) at the time of decision \( t_d \). The final orbit is defined by giving its desired orbital elements \((a_e, e_e, i_e, \omega_e, \Omega_e)\) again in planetocentric equatorial coordinates.

Subroutine CABLE is first called to convert the hyperbolic state at decision \( T, \mathbf{V} \) into Keplerian conic elements \((e_H, \omega_H, \Omega_H, \Omega_H, t_H)\) where \( t_H \) is the time from periapsis at decision (negative on the approach ray).

The points of intersection of the approach orbital plane and the desired orbital plane are then determined. The elements defining the two planes are therefore given by \( i_H, \omega_H, \Omega_H, \Omega_H, t_H \). Let \( \mathbf{A} \) denote the unit vector toward the ascending node of an orbit and \( \mathbf{B} \) denote the in-plane normal to \( \mathbf{A} \) in the direction of motion. Then

\[
\mathbf{A} = (\cos \Omega, \sin \Omega, 0) \tag{1}
\]

\[
\mathbf{B} = (-\sin \Omega \cos i, \cos \Omega \cos i, \sin i) \tag{2}
\]

Hence the normal to the orbital plane \( \mathbf{C} \) is given by \( \mathbf{C} = \mathbf{A} \times \mathbf{B} \) or

\[
\mathbf{C} = (\sin \Omega \sin i, -\cos \Omega \sin i, \cos i) \tag{3}
\]

The direction of the line of intersection of the two planes is therefore determined by \( \mathbf{Y} = \mathbf{C}_H \times \mathbf{C}_E \) or

\[
\mathbf{Y} = (\cos \omega_H \sin i_E \cos \Omega_E - \sin \omega_H \cos i_E \cos \Omega_H, \cos \omega_H \sin i_E \sin \Omega_E - \sin \omega_H \cos i_E \sin \Omega_H, \sin \omega_H \sin i_E (\cos \Omega_H \sin \Omega_E - \sin \Omega_H \cos \Omega_E)) \tag{4}
\]

Then the unit vector along the line of intersection toward the northern hemisphere is given by

\[
\mathbf{X} = \frac{\mathbf{Y}}{\mathbf{Y} \cdot \mathbf{X}} \tag{5}
\]

Therefore the true anomaly \( f_H \) along the hyperbola at the northern intersection point is given by

\[
\cos (\omega_H + f_H) = \mathbf{X} \cdot \mathbf{A}_H \tag{6}
\]
The true anomaly on the hyperbola at the southern point is therefore \( f_H = 180^\circ + \beta \). Note that there exists a region of true anomalies lying between the incoming and outgoing asymptotes for which the hyperbola is not defined. Similar equations define the true anomaly on the ellipse at the two points of intersection. Note that this implies that the modified ellipse will have the same \( \omega \) as the desired ellipse.

For the intersection true anomaly \( f_{HX} \), the radius magnitude on the hyperbola may be determined

\[
\frac{r_1}{1 + e_H \cos f_H} = a_H(1 - e_H^2)
\]

To permit an impulsive insertion, \( a_E \) and \( e_E \) must be modified to satisfy

\[
\frac{r_1}{1 + e_E \cos f_E} = a_E(1 - e_E^2)
\]

There are three candidate modifications examined to determine a "best" one:

1. Vary \( r_a \) while holding \( r_p \) constant
2. Vary \( r_p \) while holding \( r_a \) constant
3. Vary \( a \) while holding \( e \) constant

"Best" is defined below in terms of a weighted scalar function of the changes in \( r_a \) and \( r_p \).

Rewriting (8) in terms of \( r_a \) and \( r_p \) (using \( a = \frac{r_a + r_p}{2} \), \( e = \frac{r_a - r_p}{r_a r_p} \)) yields the useful relation

\[
r_a(1 + \cos f_E) + r_p(1 - \cos f_E) = \frac{2r_a r_p}{r_1}
\]

Equation (9) may be solved for \( r_a \) as

\[
r_a = \frac{r_p r_a(1 - \cos f_E)}{2 r_a - r_p (1 + \cos f_E)}
\]

This yields the \( r_a \) which defines the modified orbit holding \( r_p \) at its desired value. The semi-major axis and eccentricity are then computed from

\[
a = \frac{r_a + r_p}{2}, \quad e = \frac{r_a - r_p}{r_a + r_p}
\]
Similarly (9) may be solved for $r_p$ as

$$r_p = \frac{r_1 r_a (1 + \cos f)}{2 r_a - r_1 (1 - \cos f)}$$

(11)

This determines the modification is $r_p$ required to achieve an intersecting ellipse having the desired $r_a$.

Finally (8) may be solved trivially for the $a_E$ required to produce intersection for the desired eccentricity.

$$a_E = \frac{r_1 (1 + e_E \cos f_p)}{(1 - e_E^2)}$$

(12)

An error is assigned to each of the candidate solutions as

$$E_i = w_i \left[ |\Delta r_a| + |\Delta r_p| \right]$$

where $\Delta r_a$, $\Delta r_p$ are the errors between the desired and modified values of $r_a$ and $r_p$. The weighting factor $w_i$ is assigned rather arbitrarily. Currently the weighting factor is $w_i = w_{i1} w_{21}$ where

$$w_{i1} = 1$$
$$w_{i1} = 1$$
$$w_{i1} = 1$$

if the true anomaly is on the incoming ray
2 if the true anomaly is on the outgoing ray
1 if option 1
2 if option 2
3 if option 3

Thus a solution on the incoming asymptote is preferred over one on the outgoing asymptote and one subsequent trim is preferred over two subsequent trims.

Having determined the elements of an intersecting orbit the insertion parameters are easily computed. The velocity on the hyperbola at the intersection point may be computed from ELCAR as $\vec{v}_H$. The velocity on the ellipse following the insertion is computed by calling ELCAR with the modified elliptical element to get $\vec{v}_E$. The impulsive $\Delta \vec{v}$ is then given by

$$\Delta \vec{v} = \vec{v}_E - \vec{v}_H$$

The time interval from the decision to the execution is given by the hyperbolic time from the initial point to the relevant intersection point.
Compute hyperbolic elements at time of decision (CAREL) and record.

Compute true anomaly on hyperbolic at two points of intersection with desired orbital plane $f_{h1}, f_{h2}$.

Eliminate any solution occurring before time of decision or in impossible region between two asymptotes and set indices NT1, NT2, NSOLS accordingly.

$I = NT1$

Compute cartesian state on hyperbola at $f_{h1}$ (ELCAR) and record.

Modify $r_a$ of desired ellipse to obtain intersection with hyperbola and compute cartesian state (ELCAR). Compute $\Delta V_1$ and $\epsilon_1 = 2I \cdot |\Delta r_a|$.

Modify $r_p$ of desired ellipse to obtain intersection with hyperbola and compute cartesian state (ELCAR). Compute $\Delta V_2$ and $\epsilon_2 = I \cdot |\Delta r_p|$.

Modify "a" of desired ellipse to obtain intersection with hyperbola and compute cartesian state (ELCAR). Compute $\Delta V_3$ and $\epsilon_3 = 3I \left( |\Delta r_a + \Delta r_p| \right)$.
Choose index $\text{MIN}$ of minimum $\epsilon_k$.

- $\epsilon_{\text{MIN}} > 5 \times 10^{25}$: $\text{IEX} = 1$, RETURN
- $\epsilon_{\text{MIN}} \leq 5 \times 10^{25}$:
  - $\Delta v = \Delta v_{\text{MIN}}$
  - $\Delta t = \Delta t_{\text{MIN}}$

RETURN
SUBROUTINE NONLIN

PURPOSE: TO CONTROL EXECUTION OF NON-LINEAR GUIDANCE EVENTS.

CALLING SEQUENCE: CALL NONLIN

SUBROUTINES SUPPORTED: GUISIM GUIDM

SUBROUTINES REQUIRED: CAREL ELCAR GIDANS

LOCAL SYMBOLS:  
AA  ARGUMENT FOR SUBROUTINE CAREL
DI  JULIAN DATE OF EVENT
EE  ARGUMENT FOR SUBROUTINE CAREL
ISNPR  SAVE INPR VALUE
ISPRNT  SAVE IPRINT VALUE
KEY  INTERMEDIATE VARIABLE IN SETTING UP TARGET ARRAY
KICL2  SAVE ICL2 VALUE
KICL  SAVE ICL VALUE
KISPH  SAVE ISPH VALUE
KISP2  SAVE ISP2 VALUE
ODELT  SAVE ORIGINAL 2ELTP VALUE
OSPH  SAVE ORIGINAL SPHERE OF INFLUENCE OF TARGET PLANET
PP  ARGUMENT RETURNED FROM CAREL
QQ  ARGUMENT RETURNED FROM CAREL
RMAG  ARGUMENT FOR SUBROUTINE ELCAR
TAA  TRUE ANOMALY
TFFP  TIME OF FLIGHT FROM PERIAPSIS
TFP  TIME FROM PERIAPSIS
TRTINE  TRAJECTORY TIME OF THE GUIDANCE EVENT
VMAG  ARGUMENT FOR SUBROUTINE ELCAR
WW  ARGUMENT FOR SUBROUTINE CAREL
**COMMON COMPUTED/USED**

<table>
<thead>
<tr>
<th>DELTP</th>
<th>DELV</th>
<th>DSI</th>
<th>DT</th>
<th>IBAOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICL2</td>
<td>ICL</td>
<td>INPR</td>
<td>IPRINT</td>
<td>ISPM</td>
</tr>
<tr>
<td>ISP2</td>
<td>KLP</td>
<td>KMXQ</td>
<td>KTAR</td>
<td>KTIM</td>
</tr>
<tr>
<td>KTP</td>
<td>KWIT</td>
<td>MAT</td>
<td>MAXB</td>
<td>MDL</td>
</tr>
<tr>
<td>NODT</td>
<td>NTP</td>
<td>RSI</td>
<td>SPHERE</td>
<td>TAR</td>
</tr>
<tr>
<td>TGT3</td>
<td>TIN</td>
<td>TMU</td>
<td>TOL</td>
<td>VSI</td>
</tr>
<tr>
<td>XDC</td>
<td>XDELV</td>
<td>XRC</td>
<td>ZDAT</td>
<td></td>
</tr>
</tbody>
</table>

**COMMON COMPUTED**

<table>
<thead>
<tr>
<th>ACKT</th>
<th>AC</th>
<th>BDR</th>
<th>BDT</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG</td>
<td>DVMAX</td>
<td>D1</td>
<td>ISTART</td>
<td>IZERO</td>
</tr>
<tr>
<td>KTYT</td>
<td>KUR</td>
<td>LVLS</td>
<td>NLP</td>
<td>NOGYD</td>
</tr>
<tr>
<td>NPAR</td>
<td>PERV</td>
<td>RC</td>
<td>RIN</td>
<td>SPHFAC</td>
</tr>
<tr>
<td>TIMG</td>
<td>TRIM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMMON USED**

<table>
<thead>
<tr>
<th>ACX</th>
<th>ALNCHT</th>
<th>DATEJ</th>
<th>DELTAV</th>
<th>IX</th>
</tr>
</thead>
<tbody>
<tr>
<td>JX</td>
<td>L:LP</td>
<td>LKTAR</td>
<td>LKTP</td>
<td>LLVLS</td>
</tr>
<tr>
<td>LNPAR</td>
<td>PMASS</td>
<td>TM</td>
<td>T3</td>
<td>XAC</td>
</tr>
<tr>
<td>XBDR</td>
<td>XBDT</td>
<td>XDSI</td>
<td>XDVMAX</td>
<td>XFA</td>
</tr>
<tr>
<td>XIN</td>
<td>XPERV</td>
<td>XRSI</td>
<td>XTAR</td>
<td>XTO</td>
</tr>
<tr>
<td>XWSI</td>
<td>ZERO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NONLIN Analysis

NONLIN is the interface subroutine between the non-linear guidance subroutines of NONNAL and subroutine GUIM of ERRAN and subroutine GUISIM of SIMUL. NONLIN selects the necessary data from the ERRAN and SIMUL common blocks and stores into the common blocks of NONNAL the information needed to compute and/or execute the $\Delta V$ required in order to meet specified target conditions.

The most important task performed by NONLIN is the selection of the desired guidance scheme. The variable IX is tested and control is transformed according to the following:

$$IX = 1, \text{ retargeting to specified target parameters}$$
$$= 2, \text{ orbit insertion to specified orbit}$$
$$= 3, \text{ $\Delta V$ execution by a series of specified pulses}$$

For each type of event, NONLIN then sets up values controlling the type of guidance event (KTyp), implementation code (KXXQ), and execution model code (KDL). For retargeting only, NONLIN stores the remaining values needed for $\Delta V$ calculation and prints the zero iterate conditions.

NONNAL calls GIDANS to perform the guidance event and restores parameters necessary for the basic cycles of ERRAN and SIMUL. For retargeting only, NONNAL then stores the conditions at sphere of influence and closest approach of the target planet which were calculated by subroutine TARGET.
Save the basic cycle parameters and set up the general guidance parameters.

Set up remaining target parameters and print zero iterate conditions.

CALL GIDANS

Restore basic cycle trajectory parameters.

Was target time at closest approach?

Store S.O.I. conditions.

Calculate C.A. conditions.

Store C.A. conditions.

RETURN
SUBROUTINE NTH

PURPOSE: CONTROL COMPUTATION OF NOMINAL TRAJECTORY IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL NTH(RI,RF,NTMC,ICODE)

ARGUMENTS:
- ICODE: I INTERNAL CODE THAT DETERMINES WHICH TRAJECTORY IS BEING RUN AND WHAT INFORMATION IS DESIRED
- NTMC: I NOMINAL TRAJECTORY MODULE CODE THAT DETERMINES WHICH TYPE OF TRAJECTORY PROGRAM IS TO BE USED (NOTE: ONLY THE VIRTUAL MASS TECHNIQUE IS SUPPLIED WITH THIS PROGRAM. HOWEVER, WITH LITTLE EFFORT ANY TRAJECTORY PROGRAM MAY BE ADDED AS AN EXTRA OPTION.)
- RF: O POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: ERRANN MUNO NDM PLNO PSIM

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS:
- D1: JULIAN DATE, EPOCH JAN. 0, 1900, OF INITIAL TRAJECTORY TIME
- RMP: DISTANCE OF VEHICLE FROM TARGET PLANET AT SPHERE OF INFLUENCE OR CLOSEST APPROACH
- VMM: MAGNITUDE OF THE VELOCITY VECTOR

COMMON COMPUTED/USED:
- BDTSI1, BDTSI2, BDTSI3, BDRSI1, BDRSI2, BDRSI3, BOS, BOSI1, BOSI2, BOSI3, ICA1, ICA2, ICA3, ICL, ISI, ISI1, ISI2, ISI3, ISPO, ISPH, RCA1, RCA2, RCA3, RSAI, RSAS, RSHA, SSHA, TAI, TCI, TCA1, TCA2, TCA3, TSAI, TSAS, TSHA, VSI, VSI1, VSI2, VSI3

COMMON USED:
- ACC, BDR, BDT, B, DATEJ, DC, DELTM, DSI, IPROB, ISP2, IIR, NQE, RG, RSI, TRT
**Subroutine NIM** is used to generate the (most recent) targeted nominal trajectory in the error analysis mode. Subroutine NIM is equivalent to a subroutine NIMS from which all loops associated with ICODE = -3, -2, 2, 3 have been removed. For this reason no further analysis and no flow chart will be presented for subroutine NIM. Refer to subroutine NIMS.
SUBROUTINE NTMS

PURPOSE: CONTROL COMPUTATION OF TARGETED NOMINAL, MOST RECENT
        NOMINAL, AND ACTUAL TRAJECTORIES IN THE SIMULATION
        PROGRAM

CALLING SEQUENCE: CALL NTMS(RI,RF,NTMC,ICODE)

ARGUMENTS:

ICODE  I  INTERNAL CODE THAT DETERMINES WHICH
        TRAJECTORY IS BEING RUN AND WHAT
        INFORMATION IS DESIRED

NTMC  I  NOMINAL TRAJECTORY MODULE CODE THAT
        DETERMINES WHICH TYPE OF TRAJECTORY PROGRAM
        IS TO BE USED (NOTE ONLY THE VIRTUAL MASS
        TECHNIQUE IS SUPPLIED WITH THIS PROGRAM.
        HOWEVER, WITH LITTLE EFFORT ANY TRAJECTORY
        PROGRAM MAY BE ADDED AS AN EXTRA OPTION.)

RF  O  POSITION AND VELOCITY OF THE VEHICLE AT THE
     END OF THE TIME INTERVAL

RI  I  POSITION AND VELOCITY OF THE VEHICLE AT THE
     BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: SIMULL MUND NDTM PLMD PSIM
                       SETEVS GUISS VARSIM PREDIM

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS:

ACCS  INTERMEDIATE STORAGE FOR ACCURACY

D1  JULIAN DATE, EPOCH JAN., 1900, OF INITIAL
    TRAJECTORY TIME

K1  INDEX FOR SEMIMAJOR AXIS ELEMENT

K2  INDEX FOR ECCENTRICITY ELEMENT

K3  INDEX FOR INCLINATION ELEMENT

K4  INDEX FOR ASCENDING NODE ELEMENT

K5  INDEX FOR PERIAPSIS ELEMENT

K6  INDEX FOR MEAN ANOMALY ELEMENT

NBODS  INTERMEDIATE STORAGE FOR NBOD

NBS  INTERMEDIATE STORAGE FOR NB ARRAY

RMP  DISTANCE OF VEHICLE FROM TARGET PLANET
<table>
<thead>
<tr>
<th>SAVE10</th>
<th>INTERMEDIATE STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE1</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE2</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE3</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE4</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE5</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE6</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE7</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
<tr>
<td>SAVE8</td>
<td>INTERMEDIATE STORAGE</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED:***

<table>
<thead>
<tr>
<th>ACC</th>
<th>BDRI1</th>
<th>BDRI2</th>
<th>BDRI3</th>
<th>BDRI1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDRI2</td>
<td>BDRI3</td>
<td>BSI1</td>
<td>BSI2</td>
<td>BSI3</td>
</tr>
<tr>
<td>CN</td>
<td>EMN</td>
<td>ICA1</td>
<td>ICA2</td>
<td>ICA3</td>
</tr>
<tr>
<td>ICL</td>
<td>ISOI1</td>
<td>ISOI2</td>
<td>ISOI3</td>
<td>ISPM</td>
</tr>
<tr>
<td>MBOD</td>
<td>MB</td>
<td>PMASS</td>
<td>RCA1</td>
<td>RCA2</td>
</tr>
<tr>
<td>RCA3</td>
<td>RSOI1</td>
<td>RSOI2</td>
<td>RSOI3</td>
<td>SJNR</td>
</tr>
<tr>
<td>ST</td>
<td>TCA1</td>
<td>TCA2</td>
<td>TCA3</td>
<td>TSOI1</td>
</tr>
<tr>
<td>TSOI2</td>
<td>TSOI3</td>
<td>VSOI1</td>
<td>VSOI2</td>
<td>VSOI3</td>
</tr>
</tbody>
</table>

**COMMON USED:***

<table>
<thead>
<tr>
<th>ACC1</th>
<th>ALMTH</th>
<th>BDR</th>
<th>BD'T</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAB</td>
<td>DATEJ</td>
<td>DC</td>
<td>DEB</td>
<td>DELTM</td>
</tr>
<tr>
<td>DIB</td>
<td>OMAB</td>
<td>DMUPB</td>
<td>DNUM</td>
<td>DNOB</td>
</tr>
<tr>
<td>OSI</td>
<td>OWB</td>
<td>IPROB</td>
<td>ISP2</td>
<td>ITR</td>
</tr>
<tr>
<td>MBODI</td>
<td>MB1</td>
<td>NGE</td>
<td>NGE</td>
<td>NTP</td>
</tr>
<tr>
<td>RC</td>
<td>RSI</td>
<td>TM</td>
<td>TRTM1</td>
<td>VSI</td>
</tr>
</tbody>
</table>

**VMN**

MAGNITUDE OF THE VELOCITY VECTOR
NTMS Analysis

Subroutine NTMS is used to generate any of the three trajectories required in the simulation mode -- the (most recent) targeted nominal trajectory, the most recent nominal trajectory, and the actual trajectory.

The input variable ICODE is used to distinguish between these trajectories. It is unimportant to the virtual mass technique which trajectory is being computed. However, it is important to keep them separated so that the proper codes are set that check for approaching the sphere of influence of the target planet and reaching closest approach. It is also important to keep separate the conditions at which these occur for each trajectory. The following list describes ICODE completely.

ICODE = 3, NTMS will check to see if the sphere of influence and/or closest approach has been reached on the actual trajectory. If not, VMP will check for these conditions and on encountering either, NTMS places the conditions in special storage locations so they will be saved for future reference.

ICODE = 2, NTMS performs the same operations as described above for the most recent nominal trajectory.

ICODE = 1, NTMS again checks for sphere of influence and closest approach as above for the targeted nominal trajectory.

ICODE = 0, the only important information in this situation is the state vector at the end of the time interval. Therefore, NTMS does not check to see if closest approach or sphere of influence is encountered. This might occur in numerical differencing, for example.

ICODE = -1, it is important to know if sphere of influence or closest approach is reached on the targeted nominal trajectory. However, it is not desired that the information be stored for future use. This situation occurs in the guidance event.

ICODE = -2, the same comments may be made as if ICODE = -1, except this is on the most recent nominal trajectory.

ICODE = -3, again, this value of ICODE is treated the same as is ICODE = -1, for the actual trajectory.

Physical constants, planetary ephemerides, and other information relating to the dynamic model are the same for the targeted and most recent nominal trajectories. This is not true for the actual trajectory. There may be biases in the target planet ephemerides and the gravitational constants of the Sun and target planet. The numerical accuracy and the number of celestial bodies employed in the generation of the actual trajectory may also differ.
Ephemeris biases are specified as biases in orbital elements $a$, $e$, $i$, $\Omega$, $\omega$, and $M$. However, within the program are stored the ephemeris constants of $a$, $e$, $i$, $\Omega$, $\tilde{\omega}$, and $M$ for the planets and $a$, $e$, $i$, $\Omega$, $\tilde{\omega}$, and $L$ for the moon, where

$$\tilde{\omega} = \omega + \Omega$$

and

$$L = \lambda + \omega + \Omega.$$  

Incrementation of $\tilde{\omega}$ and $L$ requires addition of biases in $\Omega$, $\omega$, and $M$ as indicated by the above equations.
Compute Julian date associated with trajectory time $t_{k+1}$.

Is actual trajectory to be computed?

**YES**
Add actual dynamic biases to nominal dynamic parameters. Set actual trajectory numerical accuracy.

**NO**

Trajectory code NTC = ?

- patched conic
- 1
- virtual mass
- 2

Write: Patched conic trajectory not available.

IC$\geq$DE ≠ 0 ?

**NO**
IEPH = 1
ICL = 1
35

**YES**

IC$\geq$DE < 0 ?

**YES**
IEPH = 0
ICL = 0
25

**NO**

IC$\geq$DE = ?

1 targeted nominal
2 most recent nominal
3 actual trajectory

50
Has targeted nominal pierced sphere of influence?

- NO
  - ISPH = 0
  - ISPH = 1

- YES
  - HAS greatest nominal pierced sphere of influence

Has most recent nominal pierced sphere of influence?

- NO
  - ISPH = 0
  - ISPH = 1

- YES
  - HAS actual trajectory pierced sphere of influence

Has actual trajectory pierced sphere of influence?

- NO
  - ISPH = 0
  - ISPH = 1

- YES
  - HAS closest approached closest approach

Has targeted nominal encountered closest approach?

- NO
  - ICL = 0
  - ICL = 1

- YES
  - HAS most recent nominal encountered closest approach

Has most recent nominal encountered closest approach?

- NO
  - ICL = 0
  - ICL = 1

- YES
  - HAS actual trajectory encountered closest approach

Has actual trajectory encountered closest approach?

- NO
  - ICL = 0
  - ICL = 1

- YES
  - CALL VMP to compute the specified trajectory at $t_k$

ICODE $\leq 0$?

- YES
  - GO to 50

- NO
  - ICODE = 1
    - GO to 36
  - ICODE = 2
    - GO to 41
  - ICODE = 3
    - GO to 45
Did most recent nominal pierce sphere of influence?

NO

NO

YES

Did targeted nominal pierce sphere of influence?

YES

Was sphere of influence previously pierced?

NO

ISO11 = 1

Store sphere of influence conditions for the targeted nominal trajectory. Write out sphere of influence conditions.

YES

Was closest approach previously encountered?

NO

Store closest approach conditions for the targeted nominal trajectory.

YES

Did targeted nominal encounter closest approach?

YES

ICA1 = 1

Write out closest approach conditions.

NO

Is most recent nominal identical to target nominal?

YES

41

NO

50

YES

Was closest approach previously encountered?

NO

Store closest approach conditions for the most recent nominal trajectory.

YES

Did targeted nominal encounter closest approach?

YES

ICA2 = 1

Write out closest approach conditions.
43

NO

Did actual trajectory pierce sphere of influence?

YES

Was sphere of influence previously pierced?

NO

YES

Was closest approach previously encountered?

NO

Store closest approach conditions for the actual trajectory.

Did actual trajectory encounter closest approach?

YES

NO

I(913 = 1

Write out closest approach conditions.

30

Was actual trajectory computed?

YES

NO

Reset dynamic parameters and trajectory numerical accuracy to their nominal values.

RETURN
SUBROUTINE ORB

PURPOSE: TO COMPUTE THE ORBITAL ELEMENTS -- INCLINATION, LONGITUDE OF ASCENDING NODE, LONGITUDE OF PERIHElION, ECCENTRICITY, AND LENGTH OF SEMIMAJOR AXIS -- FOR A SPECIFIED PLANET AT A GIVEN TIME.

CALLING SEQUENCE: CALL ORB(IP,D)

ARGUMENTS:  D I JULIAN DATE, EPOCH 1900, OF THE TIME AT WHICH THE ELEMENTS ARE TO BE CALCULATED

IP I CODE NUMBER OF PLANET
=1 SUN
=2 MERCURY
=3 VENUS
=4 EARTH
=5 MARS
=6 JUPITER
=7 SATURN
=8 URANUS
=10 PLUTO
=11 MOON

SUBROUTINES SUPPORTED: DATA DATAS PCTM PRINT3 PRINT4 PSIM TRAKM TRAKS TRAPAR VMP GUIDM GUID GUSSIM GUISS PRNTS3 HELIO LAUNCH LUNTAR MULCON MULTAR PRNTS4 TRAPAR

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: FN1 STATEMENT FUNCTION DEFINING A THIRD ORDER POLYNOMIAL
FN2 STATEMENT FUNCTION DEFINING A FIRST ORDER POLYNOMIAL
ITEMP INTERMEDIATE VARIABLE
PI2 TWICE THE MATHEMATICAL CONSTANT PI

COMMON COMPUTED/USED: ELMNT T
COMMON USED: CN EMN SMJR ST TWOPI
ORB Analysis

ORB determines the mean orbital elements for any gravitational body at a specified time.

The elements used are semi-major axis $a$, eccentricity $e$, inclination $i$, longitude of the ascending node $\Omega$, and longitude of perihelion $\Omega$. These elements are referenced to heliocentric ecliptic for the planets or geocentric ecliptic for the moon.

The mean elements are computed from time expansions as follows. Let $\sigma$ be any of the elements. Then the value of $\sigma$ at any time $t$ is given by

$$\sigma(t) = \sigma_0 + \sigma_1 t + \sigma_2 t^2 + \sigma_3 t^3$$

where the constants $\sigma_k$ are stored by BILKDAT. These constants are stored into the arrays CN, ST, and E2N for inner planets, outer planets, and the moon respectively. The definitions of these arrays and the values stored are provided in the analysis of the previous subroutine ELMNT. The element value as computed from the above equation is then returned in the ELMNT array according to the gravitational body code $k$ as

<table>
<thead>
<tr>
<th>ELMNT(8k-15)</th>
<th>$k$</th>
<th>$k = 1$ Sun</th>
<th>$k = 1$ Saturn</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMNT(8k-14)</td>
<td>$\Omega$</td>
<td>2 Mercury</td>
<td>8 Uranus</td>
</tr>
<tr>
<td>ELMNT(8k-13)</td>
<td>$\omega$</td>
<td>3 Venus</td>
<td>9 Neptune</td>
</tr>
<tr>
<td>ELMNT(8k-12)</td>
<td>$e$</td>
<td>4 Earth</td>
<td>10 Pluto</td>
</tr>
<tr>
<td>ELMNT(8k-10)</td>
<td>$a$</td>
<td>5 Mars</td>
<td>11 Moon</td>
</tr>
<tr>
<td>ELMNT(8k-9)</td>
<td>$\omega$</td>
<td>6 Jupiter</td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE PARTL

PURPOSE
COMPUTE PARTIALS OF B DOT T AND B DOT R WITH RESPECT TO
SPACECRAFT POSITION AND VELOCITY

CALLING SEQUENCE:
CALL PARTL(R,V,B,BDT,BDR,PBT,PBR)

ARGUMENTS
B 0 IMPACT PLANE PARAMETER
BDR 0 B DOT R
BDT 0 B DOT T
PBR 0 PARTIAL OF B DOT R WITH RESPECT TO R AND V
PBT 0 PARTIAL OF B DOT T WITH RESPECT TO R AND V
R 1 POSITION OF VEHICLE RELATIVE TO PLANET
V 1 VELOCITY OF VEHICLE RELATIVE TO PLANET

SUBROUTINES SUPPORTED:
GUSS  GUID

LOCAL SYMBOLS:
H3  INTERMEDIATE VARIABLE
RU  INTERMEDIATE VARIABLE
S  MAGNITUDE OF VELOCITY
U  INTERMEDIATE VARIABLE
U2  SQUARE OF U
U2PV2  INTERMEDIATE VARIABLE
UV  INTERMEDIATE VARIABLE
UV3  CUBE OF UV
V2  SQUARE OF MAGNITUDE OF VELOCITY

COMMON USED:
ZERO
PARTL Analysis

PARTL is responsible for the computation of the partials of $B\cdot T$ and $B\cdot R$ with respect to the cartesian components of position and velocity.

Let the state of the spacecraft with respect to the target body at intersection with its sphere of influence be denoted

$$
\vec{r} = [x, y, z]^T \quad r = \sqrt{y^2 + z^2} \\
\vec{v} = [\dot{x}, \dot{y}, \dot{z}]^T \quad v = \sqrt{\dot{y}^2 + \dot{z}^2}
$$ (1)

Introduce the approach asymptote $\hat{S}$ and approximate it by the direction of $\vec{v}$

$$
\hat{S} = \frac{\vec{v}}{v}
$$ (3)

The B-plane is the plane normal to $\hat{S}$ containing the center of the target body. Any vector $\vec{B}$ within the B-plane must satisfy therefore

$$
\hat{S} \cdot \vec{B} = 0
$$ (4)

The impact parameter vector $\vec{B}$ is determined by the intersection of the B-plane and the incoming asymptote. The incoming asymptote is given parametrically by

$$
\vec{B} = \vec{r} + \vec{v} t
$$ (5)

The time at which the asymptote intersects the B-plane may be determined by applying the B-plane condition (4)

$$
\hat{S} \cdot \vec{r} + \hat{S} \cdot \vec{v} t = 0
$$

$$
t = -\frac{\vec{r} \cdot \vec{v}}{v^2}
$$ (6)

Therefore the B-vector is given by

$$
\vec{B} = \vec{r} - \frac{\vec{r} \cdot \vec{v}}{v^2} \vec{v}
$$

$$
\vec{B} = [x - \alpha \dot{x}, y - \alpha \dot{y}, z - \alpha \dot{z}]^T
$$ (7)

where

$$
\alpha = \frac{\vec{r} \cdot \vec{v}}{v^2} = \frac{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}
$$
Now assuming that the $T$ axis is to lie in the $x$-$y$ reference plane and the $B$-plane, it is defined as

$$ T = \begin{bmatrix} \hat{S} \times \hat{R} \\ \hat{S} \times \hat{T} \end{bmatrix} $$

where $u^2 = x^2 + y^2$. The $\hat{R}$ axis is defined by

$$ \hat{R} = \hat{S} \times \hat{T} $$

Now combining (7), (8), (9) $B \cdot T$ and $B \cdot R$ may be computed in terms of the state components

$$ B \cdot T = \frac{1}{u} (x'y - xy) $$

$$ B \cdot R = \frac{1}{uv} \begin{bmatrix} (x' + y'y)z - u^2 z \end{bmatrix} $$

where $u^2 = x^2 + y^2$, $v^2 = u^2 + z^2$.

The partials may now be computed from differentiation of the above equations.

$$ \frac{\partial B \cdot T}{\partial x} = \frac{y}{u} \quad \frac{\partial B \cdot R}{\partial x} = \frac{z}{uv} $$

$$ \frac{\partial B \cdot T}{\partial y} = \frac{-x}{u} \quad \frac{\partial B \cdot R}{\partial y} = \frac{-y}{uv} $$

$$ \frac{\partial B \cdot T}{\partial z} = 0 \quad \frac{\partial B \cdot R}{\partial z} = \frac{-u}{v} $$

$$ \frac{\partial B \cdot T}{\partial x} = \frac{1}{u^3} (x' + y'y) \quad \frac{\partial B \cdot R}{\partial x} = \frac{1}{u^3 v^3} \left[ u^2 (v^2 x - z'z) - x' (u^2 + v^2) (x' + y'y) \right] $$

$$ \frac{\partial B \cdot T}{\partial y} = \frac{1}{u^3} (x' + y'y) \quad \frac{\partial B \cdot R}{\partial y} = \frac{1}{u^3 v^3} \left[ u^2 (v^2 y - y'z) - y' (u^2 + v^2) (x' + z') \right] $$

$$ \frac{\partial B \cdot T}{\partial z} = 0 \quad \frac{\partial B \cdot R}{\partial z} = \frac{1}{u^3 v^3} (x' + y'y - z') $$
SUBROUTINE PCTM

PURPOSE: CONTROL COMPUTATION OF STATE TRANSITION MATRIX USING THE
ANALYTICAL PATCHED CONIC TECHNIQUE

CALLING SEQUENCE: CALL PCTM(RI)

ARGUMENTS: RI I POSITION AND VELOCITY OF THE VEHICLE AT THE
BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: CONC2 EPHEN ORB

LOCAL SYMBOLS: D JULIAN DATE, EPOCH JAN.0, 1900, OF INITIAL
TIME
DELT LENGTH OF TIME INCREMENT IN PROPER UNITS
DUM TEMPORARY STORAGE FOR STATE TRANSITION
MATRIX
GMS GRAVITATIONAL CONSTANT OF GOVERNING BODY
IP CODE OF PLANET
RM DISTANCE FROM SPECIFIED PLANET
RS POSITION OF VEHICLE RELATIVE TO SPECIFIED
PLANET
VS VELOCITY OF VEHICLE RELATIVE TO SPECIFIED
PLANET

COMMON COMPUTED/USED: XP

COMMON COMPUTED: NO PHI

COMMON USED: ALNGTH DATEJ DELTM F IBARY
NBOD MB PMASS SPHERE TN
TRTM1
Subroutine PCTM does not actually compute the state transition matrix \( \Phi(t_k, t_{k-1}) \) itself; this is accomplished by calling CPCSIC2 from within PCTM. The primary function of PCTM is to determine the dominant body at time \( t_{k-1} \) to be used in the computation of \( \Phi(t_k, t_{k-1}) \) by means of the analytical patched conic technique.

On interplanetary trajectories we compute the distance separating the spacecraft from each of the celestial bodies included in the analysis. If the distance between the spacecraft and the \( i \)-th body is less than or equal to 1.1 times the sphere of influence of the \( i \)-th body, the \( i \)-th body is selected as the dominant body. Otherwise, the Sun is selected as the dominant body.

On luna trajectories we compute the distance separating the spacecraft from the Moon. If this distance is less than or equal to 1.1 times the sphere of influence of the Moon, the moon is selected as the dominant body. If not, the Earth is selected as the dominant body.
FCM Flow Chart

ENTER

Compute Julian date at $t_{k-1}$

Is reference coordinate system heliocentric or barycentric ecliptic?

Yes: bary -

No: helio -

I = 1

Compute position and velocity of I-th body. Compute position, velocity, and position magnitude of spacecraft relative to I-th body.

I = I + 1

Is spacecraft within 1.1 times the sphere of influence of the I-th body?

Yes: YES

No: NO

Have all celestial bodies been treated?

Yes: B

Select Sun as dominant body.

Select I-th body as dominant body.

Compute gravitational constant of dominant body.

Call ORB1C2 to compute $\phi$ analytically over the time interval $[t_{k-1}, t_k]$.

RETURN
Compute position and velocity of I-th body. Compute position, velocity, and position magnitude of spacecraft relative to I-th body.

I-th body = Earth?

Yes

Store position and velocity of spacecraft relative to Earth.

Target body = Moon?

No

Is spacecraft within 1.1 times the sphere of influence of the I-th body?

No

Is target body = Moon and is I-th body = Moon?

Yes

Select I-th body as dominant body.

Yes

Select Earth as dominant body.

I = I + 1

No

Have all bodies been treated?

Yes

B

D

C

A
SUBROUTINE PECEQ

PURPOSE: TO COMPUTE THE MATRIX DEFINING THE TRANSFORMATION FROM PLANET CENTERED ECLIPTIC COORDINATES TO PLANET CENTERED EQUATORIAL COORDINATES AS A FUNCTION OF THE PARTICULAR PLANET AND TIME.

CALLING SEQUENCE: CALL PECEQ(NP,D,ECEQ)

ARGUMENT DESCRIPTION
NP I CODE OF PLANET
D I JULIAN DATE, EPOCH 1900, OF REFERENCE TIME
ECEQ(3,3) I TRANSFORMATION MATRIX FROM ECLIPTIC TO EQUATORIAL COORDINATES

SUBROUTINES SUPPORTED: TARGET HELIO LAUNCH LUNITAR MULTAR INSERS TRAPAR VMP DATAS GUSIM DATA GUIDM EXECUTE

SUBROUTINES REQUIRED: EULMX

LOCAL SYMBOLS: DD JULIAN TIME (EPOCH 1900) DIVIDED BY 10000.
DGTR CONVERSION FACTOR FROM DEGREES TO RADIANS
ECOP TRANSFORMATION MATRIX FROM ECLIPTIC TO ORBITAL PLANE
OPEQ TRANSFORMATION MATRIX FROM ORBITAL TO EQUATORIAL PLANE
T YEARS OF JULIAN TIME (EPOCH 1900)
XI THE INCLINATION OF THE ORBITAL PLANE TO ECLIPTIC PLANE
XIQ INCLINATION OF PLANET EQUATOR TO ORBITAL PLANE
XL THE LONGITUDE OF THE ASCENDING NODE OF THE ORBITAL PLANE TO THE ECLIPTIC PLANE
XLQ THE LONGITUDE OF THE ASCENDING NODE OF THE PLANET EQUATOR TO THE ORBITAL PLANE

COMMON USED: EMN EM2 EM4 EM5 EM6 EM7 EM8 ZERO
**FECEQ Analysis**

FECEQ is responsible for the computation of the transformation matrix from ecliptic to planet equatorial coordinates for any planet (or moon) at any time.

![Diagram of Geometric Relationships](image)

**Figure 1. Geometry of Ecliptic, Orbital, and Equatorial Planes**

Figure 1 illustrates the geometry of the problem. In that figure the following definitions hold:

- **XYZ**  
  The ecliptic coordinate axes
- **X₀Y₀Z₀**  
  The orbital plane coordinate axes
- **X₟Y₟Z₟**  
  The planet equatorial coordinate axes
- **i**  
  Inclination of orbital plane to ecliptic plane
- **Ω**  
  Right ascension of orbital plane to ecliptic plane
- **δ**  
  Inclination of planet equator to orbital plane
- **θ**  
  Right ascension of planet equatorial to orbital plane

Time expansions for the mean values of the angles are evaluated at the specific time. Then the construction of the transformation matrix proceeds in two steps:
In the first step the transformation from the ecliptic to orbital plane coordinates is made. This is done by rotating about the z-axis through an angle $\Omega$ and then about the resulting x-axis through an angle $\iota$:

$$T_{ECOP} = (\Omega \text{ about } 3, \iota \text{ about } 1)$$

The transformation from the orbital plane to the equatorial coordinates is accomplished in a similar fashion:

$$T_{ECOP} = (\theta \text{ about } -3, \delta \text{ about } -1)$$

The rotation matrix (or Euler matrix) for each of these transformations is computed by subroutine EULMX. The transformation from ecliptic to equatorial is now given as the product of these matrices

$$T_{ECOE} = T_{OPEQ} T_{ECOP}$$
**SUBROUTINE** PERHEL

**PURPOSE:** TO PROPAGATE A HELIOCENTRIC TRAJECTORY CONSIDERING THE PERTURBATIONS PRODUCED BY BOTH THE LAUNCH AND TARGET BODIES.

**CALLING SEQUENCE:** CALL PERHEL(GM, HSI, HLTI, HLTF, DELT, HSF)

**ARGUMENTS:**
- **GM(3):** I GRAVITATIONAL CONSTANTS OF SUN, LAUNCH AND TARGET PLANETS
- **HSI(6):** I HELIOCENTRIC ECLIPTIC SPACECRAFT STATE (INITIAL)
- **HLTI(2,3):** I INITIAL HELIOCENTRIC STATES OF LAUNCH AND TARGET BODIES
- **HLTF(2,3):** I FINAL HELIOCENTRIC STATES OF LAUNCH AND TARGET BODIES
- **DELT:** I TIME INTERVAL OF PROPAGATION
- **HSF(6):** O HELIOCENTRIC ECLIPTIC SPACECRAFT STATE (FINAL)

**SUBROUTINES SUPPORTED:** PULCOV PULSEX

**SUBROUTINES REQUIRED:** BATCON

**LOCAL SYMBOLS:**
- **CON:** INTERMEDIATE VARIABLE
- **DEL:** RF-RI
- **PER:** PERTURBATION IN FINAL STATE
- **PSF:** SPACECRAFT POSITION RELATIVE TO PLANET (FINAL)
- **PSI:** SPACECRAFT POSITION RELATIVE TO PLANET (INITIAL)
- **RAV:** AVERAGE OF RI AND RF
- **RA:** INTERMEDIATE VARIABLE
- **RF:** MAGNITUDE OF PSF
- **RM:** INTERMEDIATE VARIABLE
- **RI:** MAGNITUDE OF PSI

---

**PERHEL-A**
PIIItUPir reponrible for propagating a heliocentric trajectory considering the perturbations produced by both the launch and target bodies. The equations of motion of a body moving under the influence of the sun while perturbed by a smaller mass are

\[
\ddot{\mathbf{r}} = - \frac{\mu_0 \mathbf{r}}{r^3} - \frac{\mu (\mathbf{r} - \mathbf{r}_m)}{|\mathbf{r} - \mathbf{r}_m|^3} - \frac{\mu \mathbf{r}_m}{r_m^3} \tag{1}
\]

where \( \mathbf{r} \) is the vector radius from the sun to the spacecraft
\( \mathbf{r}_m \) is the vector radius from the sun to the perturbative mass
\( \mu_0, \mu \) are the gravitational constants of the sun and mass respectively.

Assuming that the indirect term is small, attention may be directed to the first two terms only. Suppose that \( (\mathbf{r}_0(t), \mathbf{v}_0(t)) \) satisfy

\[
\begin{align*}
\dot{\mathbf{r}}_o &= \mathbf{v}_o \\
\dot{\mathbf{v}}_o &= \frac{\mu_0 \mathbf{r}_o}{r_o^3}
\end{align*} \tag{2}
\]

Then \( (r_0(t), v_0(t)) \) are given by the familiar equations of conic motion.

A first order corrected solution necessary to account for the direct term force must then satisfy

\[
\begin{align*}
\dot{\mathbf{r}} &= \dot{\mathbf{r}}_o + \dot{\mathbf{r}} \ 	ext{and} \\
\dot{\mathbf{v}} &= \dot{\mathbf{v}}_o + \dot{\mathbf{v}} - \frac{\mu_0 \mathbf{r}_o}{r_o^3} - \mu \frac{(\mathbf{r}_o - \mathbf{r}_m)}{|\mathbf{r}_o - \mathbf{r}_m|^3}
\end{align*} \tag{3}
\]

Applying the conditions (2) leads to the equations defining the corrections

\[
\begin{align*}
\dot{\mathbf{r}} &= \dot{\mathbf{v}} \\
\dot{\mathbf{v}} &= - \frac{\mu \mathbf{r}}{r^3}
\end{align*} \tag{4}
\]

where \( \mathbf{r} = \mathbf{r}_o(t) - \mathbf{r}_m(t) \) is the position vector of the spacecraft with respect to the perturbing mass.

One further assumption enables one to solve in closed form the perturbations produced by the third mass. Generally \( \mathbf{R}(t) \) and \( \mathbf{R}(t) \) are nearly linear functions of time. Therefore suppose that the initial and final values of these variables are known to be \( \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_1, \mathbf{R}_2 \) over the interval \( \Delta t \).
Introduce the definitions
\[ \Delta \vec{R} = \vec{R}_2 - \vec{R}_1 \]
\[ \Delta R = R_2 - R_1 \quad \text{(not } |\Delta \vec{R}|) \]
\[ \langle \vec{R} \rangle = \frac{1}{2} (\vec{R}_1 + \vec{R}_2) \]
\[ \Delta \vec{R} = \frac{\vec{R}_2 - \vec{R}_1}{R_2 - R_1} \] \hspace{1cm} (5)

Then the equation defining the velocity perturbation would be
\[ \frac{\Delta \vec{v}}{\Delta t} = -\mu \frac{\vec{a} + \vec{b} \rho}{(c + d \Delta t)^3} \quad \vec{a} = \vec{R}_1 \quad c = R_1 \]
\[ \vec{b} = \frac{\Delta \vec{R}}{\Delta t} \quad d = \frac{\Delta R}{\Delta t} \] \hspace{1cm} (6)

It is more convenient however to transform from time \( t \) to position magnitude \( \rho \) as the independent variable. This may be done since the position magnitude is assumed to be linear in time with \( \rho = \frac{\Delta R}{\Delta t} \).

According to the assumptions, the position vector \( \vec{R} \) is a linear function of \( \rho \) also
\[ \vec{R} = \vec{A} + \vec{B} \rho \] \hspace{1cm} (7)

Since \( \vec{R}(\rho_1) = \vec{R}_1 \) and \( \vec{R}(\rho_2) = \vec{R}_2 \), the constants are
\[ \vec{A} = \vec{R}_1 - \frac{\Delta \vec{R}}{\Delta R} R_1 = -\frac{R_1 R_2}{\Delta R} \hat{\Delta R} \]
\[ \vec{B} = \frac{\Delta \vec{R}}{\Delta R} \]

In terms of \( \rho \) the equations defining the perturbations may be written (with primes indicating differentiation with respect to \( \rho \))
\[ \vec{A}' = \frac{\Delta E}{\Delta R} \vec{A} \quad \vec{B}' = -\frac{\mu \Delta \rho}{\Delta R} \vec{A} + \frac{\vec{B}}{\rho} \] \hspace{1cm} (9)

These equations are easily integrated to determine the perturbations caused as the spacecraft moves from \( \vec{R}_1 \) to \( \vec{R}_2 \) relative to the perturbative body.
FREHEL calls HATCON for the generation of the uncorrected heliocentric conic, computes the initial and final positions of the spacecraft relative to each of the launch and target planets, and computes the perturbations based on equations (10) and (11) above.

\[ \Delta v = \mu \frac{\Delta t}{\Delta R} \int_{R_1}^{R_2} \frac{\vec{A} + \vec{B} \rho}{\rho^3} \, d\rho \]

\[ = \mu \frac{\Delta t}{\Delta R} \left[ \frac{\vec{A}}{2} \left( \frac{1}{\rho^2} - \frac{1}{R_1^2} \right) + \vec{B} \left( \frac{1}{\rho} - \frac{1}{R_1} \right) \right] \]

\[ = \mu \frac{\Delta t}{R_1 R_2 \Delta R} \left[ \langle \rho \rangle \Delta R - \Delta \vec{R} \right], \quad \rho = R_2 \]  

(10)

\[ \Delta r = \mu \frac{\Delta t^2}{\Delta R^2} \int_{R_1}^{R_2} \left[ \frac{\vec{A}}{2} \left( \frac{1}{\rho^2} - \frac{1}{R_1^2} \right) + \vec{B} \left( \frac{1}{\rho} - \frac{1}{R_1} \right) \right] \, d\rho \]

\[ = \mu \frac{\Delta t^2}{\Delta R} \left[ \frac{1}{2} \frac{\Delta R}{\rho^2} + \frac{\Delta R}{\Delta R^2} \left( \ln \left( \frac{R_2}{R_1} \right) - \frac{\Delta R}{R_1} \right) \right] \]  

(11)
PERHEL Flow Chart

ENTER

Call BATCON to propagate state over \( \Delta t \) as heliocentric conic

\[ IB = IB + 1 \]

Compute initial and final relative position of S/C to body IB

Compute perturbations of IB body

\[ IB = ? \]

Add perturbations of launch and target bodies to conic final state

RETURN
SUBROUTINE PLND

PURPOSE: TO COMPUTE COLUMNS OF THE STATE TRANSITION MATRIX
PARTITIONS TXXS AND TXU ASSOCIATED WITH TARGET PLANET
EPHEMERIS BIASES INCLUDED IN THE AUGMENTED STATE VECTOR
BY A NUMERICAL DIFFERENCING TECHNIQUE.

CALLING SEQUENCE: CALL PLND(RI,RF)

ARGUMENTS:
RF I POSITION AND VELOCITY OF THE VEHICLE AT THE END OF THE TIME INTERVAL
RI I POSITION AND VELOCITY OF THE VEHICLE AT THE BEGINNING OF THE TIME INTERVAL

SUBROUTINES SUPPORTED: PSIM

SUBROUTINES REQUIRED: NTH

LOCAL SYMBOLS:
DEL TEMPORARY STORAGE FOR TARGET PLANET SEMI-MAJOR AXIS FACTOR USED IN NUMERICAL DIFFERENCING
IC COUNTER FOR VARIABLES AUGMENTED TO STATE VECTOR
IEMN VECTOR OF INDICES FOR ORBITAL ELEMENTS OF THE MOON
IEND FLAG FOR VARIABLES AUGMENTED TO STATE VECTOR
IPR TEMPORARY STORAGE FOR IPRINT
NEW VECTOR OF INDICES FOR ORBITAL ELEMENTS OF INNER AND OUTER PLANETS
RPER ALTERED FINAL POSITION AND VELOCITY OF VEHICLE
SAVE1 TEMPORARY STORAGE FOR CONSTANS OF AUGMENTED ELEMENTS OF TARGET PLANET
SAVE2 SAME COMMENTS AS SAVE1

COMMON COMPUTED/USED:
CN EMN IPRINT SMJR ST
COMMON COMPUTED:
TXX TXXS
COMMON USED:
ALNGTH DELAXS DELX IAUGDC I AUGIN
IAUG NTMC NTP
FLND Analysis

The nonlinear equations of motion of the spacecraft can be written symbolically as

$$\dot{x} = f(x, \varphi(t), t)$$  \hspace{2cm} (1)

where $\dot{x}$ is the spacecraft position/velocity state and $\varphi(t)$ is a vector composed of the 6 orbital elements $a$, $e$, $i$, $\Omega$, $\omega$, and $M$ of the target planet. The motion of the spacecraft is, of course, dependent on the positions of other celestial bodies, but this dependency need not be explicitly stated for the purposes of this analysis.

Suppose we wish to use numerical differencing to compute those columns of $\varphi_{xx}$ and $\varphi_{xu}$ associated with target planet ephemeris biases included in the augmented state vector over the time interval $[t_{k-1}, t_k]$. Let $\varphi_j(t_k, t_{k-1})$ represent the column associated with the j-th ephemeris bias. We assume we have available the nominal states $\varphi^*(t_{k-1})$ and $\varphi^*(t_k)$, which, of course, were obtained by numerically solving equation (1) using nominal $\varphi(t)$. To obtain $\varphi_j(t_k, t_{k-1})$ we increment the j-th orbital element by the pertinent numerical incrementing factor $\Delta e_j$, and numerically integrate equation (1) over the interval $[t_{k-1}, t_k]$ to obtain the new spacecraft state $\varphi_j(t_k)$, where the j-subscript on the spacecraft state indicates that it was obtained by incrementing the j-th orbital element.

Then

$$\varphi_j(t_k, t_{k-1}) = \frac{\varphi_j(t_k) - \varphi^*(t_k)}{\Delta e_j}.$$  \hspace{2cm} (2)

 Ephemeris biases are defined as biases of the basic set of orbital elements $a$, $e$, $i$, $\Omega$, $\omega$, and $M$. However, within the program are stored the ephemeris constants of $a$, $e$, $i$, $\Omega$, $\omega$, and $M$ for the planets and $a$, $e$, $i$, $\Omega$, $\omega$, and $L$ for the moon. Thus, in order to increment certain of the basic elements we must increment certain combinations of the stored ephemeris constants.

The elements $\omega$ and $M$ are related to the longitude of perihelion $\Omega$ and the mean longitude $L$ as follows:

$$\omega = \Omega - \Omega$$

$$M = L - \Omega$$

Thus, to increment $\Omega$ by $\Delta \Omega$ without changing the other 5 basic elements requires that we also increment $\omega$ by $\Delta \Omega$ for the case of a planet, and
both $\Omega$ and $L$ by $\Delta \varpi$ for the case of the moon. To increment $\omega$ by $\Delta \omega$ we simply increment $\omega$ by $\Delta \omega$ for a planet, while for the moon we must increment both $\Omega$ and $L$ by $\Delta \omega$. To increment $\kappa$ by $\Delta \kappa$ for the moon we simply increment $L$ by $\Delta \kappa$.

In the PLNO flow chart we employ the following definition:

\[
p_j = \begin{cases} 
a, & j = 1 \\
e, & j = 2 \\
i, & j = 3 \\
\Omega, & j = 4 \\
\omega, & j = 5 \\
\kappa, & j = 6 \\
\end{cases}
\]
Flmd Flow Chart

**ENTER**

j = 1

A

j = j + 1

NO

Is \( \delta p_j \) in \( \mathbb{Z} \) or \( \mathbb{U} \)?

YES

j = 1?

YES

B

NO

Is the target planet the Moon?

YES

C

NO

Is the target planet beyond Mars?

NO

Increment the appropriate element of the CM array after saving its original value.

Increment the appropriate element of the ST array after saving its original value.

Call NIM to compute the state at \( t_k \) resulting from the incrementation of the \( j \)-th orbital element at \( t_{k-1} \).

Call NIM to compute the state at \( t \) resulting from the incrementation of the \( j \)-th orbital element at \( t_{k-1} \).

Reset the appropriate element of the CM array to its original value.

Reset the appropriate element of the ST array to its original value.

D
bpute the appropriate column of $\Theta_{xu}$ using equation (2).

Is $\delta p_j$ in $\hat{z}$ or $\hat{c}$?

\[ \text{Compute the appropriate column of } \Theta_{xu} \text{ using equation (2).} \]

\[ \text{Is } \delta p_j \text{ in } \hat{z} \text{ or } \hat{c}? \]

\[ \text{Compute the appropriate column of } \Theta_{xu} \text{ using equation (2).} \]

\[ \text{Increment the appropriate element of the } \text{EMN array after saving its original value.} \]

\[ \text{Increment the appropriate element of the } \text{EMN array after saving its original value.} \]

\[ \text{Call NDM to compute the state at } t_k \text{ resulting from the incrementation of the semi-major axis at } t_{k-1}. \]

\[ \text{Call NDM to compute the state at } t_k \text{ resulting from the incrementation of the j-th orbital element of the Moon at } t_{k-1}. \]

\[ \text{Reset the appropriate element of the EMN array to its original value.} \]

\[ \text{Reset the appropriate element of the EMN array to its original value.} \]
SUBROUTINE POICOM

PURPOSE COMPUTE PROBABILITY OF IMPACT

CALLING SEQUENCE: CALL POICOM(XXXX,DET)

ARGUMENTS

<table>
<thead>
<tr>
<th>XXXX</th>
<th>AIMPOINT IN THE IMPACT PLANE VECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>DETERMINANT OF LAMBDAMATRIX</td>
</tr>
</tbody>
</table>

SUBROUTINES SUPPORTED: BIAIM

SUBROUTINES REQUIRED: MATIN

LOCAL SYMBOLS:

<table>
<thead>
<tr>
<th>PMQM</th>
<th>P+ MQM TRANSPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>SUM</td>
<td>INTERMEDIATE VARIABLE</td>
</tr>
<tr>
<td>W</td>
<td>ADA* PM* ADA TRANSPOSE</td>
</tr>
</tbody>
</table>

COMMON COMPUTED/USED:

<table>
<thead>
<tr>
<th>IEND</th>
<th>POI</th>
<th>PSTAR</th>
<th>XLM</th>
</tr>
</thead>
</table>

COMMON USED:

<table>
<thead>
<tr>
<th>ADA</th>
<th>A</th>
<th>CR</th>
<th>EXEC</th>
<th>IIGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE</td>
<td>PI</td>
<td>PP</td>
<td>TWO</td>
<td>XLAM</td>
</tr>
<tr>
<td>ZERO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
POICOM Analysis

Subroutine POICOM computes the target condition covariance $W_j^+$ after a guidance correction, the projection of $W_j^+$ into the impact plane, and the probability of impact of the spacecraft with the target planet.

The target condition covariance matrix $W_j^+$ is defined as

$$W_j^+ = \eta_j (P_{k_j}^- + Q_j M^T) \eta_j^T$$

where $\eta_j$ is the variation matrix for the appropriate guidance policy, $P_{k_j}^-$ is the knowledge covariance prior to the guidance correction, $Q_j$ is the execution error covariance, and $M$ is defined as the following $6 \times 3$ matrix:

$$M = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

Before the probability of impact can be computed, it is necessary to compute the projection $\Lambda_j$ of $W_j^+$ into the impact plane. The covariance $\Lambda_j$ is computed as follows for each of the three available midcourse guidance policies.

a. Fixed-time-of-arrival:

$$\Lambda_j = A W_j^+ A^T$$

where transformation $A$ is defined in the subroutine BIALM analysis.

b. Two-variable B-plane:

$$\Lambda_j = W_j^+$$

c. Three-variable B-plane:

$$\Lambda_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} W_j^+ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
Assuming the probability density function associated with $\mathbf{A}_j$ is Gaussian and nearly constant over the target planet capture area permits us to compute the probability of impact using the equation

$$\text{POI} = \pi R_c^2 p$$

where $R_c$ is the target planet capture radius and $p$ is the Gaussian probability density function evaluated at the target planet center and given by

$$p = \frac{1}{2\pi|\mathbf{A}_j|^{1/2}} \exp \left[ -\frac{1}{2} \mathbf{\mu}^T \mathbf{A}_j^{-1} \mathbf{\mu} \right]$$

where $\mathbf{\mu}$ is the aimpoint in the impact plane.
Compute

\[ P_{k_j}^{-1} + \tilde{N}_{j} M^T \]

Compute target condition covariance matrix

\[ W_j^+ \]

Compute and write out \( \Lambda_j \) covariance matrix.

Call MATIN to compute \( \Lambda_j^{-1} \).

Compute

\[ \mu^T \Lambda_j^{-1} \mu \]

Compute det \( (\Lambda_j) \).

Evaluate Gaussian probability density function \( p \) at target planet center.

Compute and write out the approximate probability of impact.

RETURN
PROGRAM PRED

PURPOSE CONTROL EXECUTION OF A PREDICTION EVENT IN THE ERROR ANALYSIS PROGRAM

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED: CORREL DYNO HTELS JACOBI NAVM NTM PSIM STMPR

LOCAL SYMBOLS CXSU1 STORAGE FOR CXSU COVARIANCE ARRAY
               CXSV1 STORAGE FOR CXSV COVARIANCE ARRAY
               CXU1 STORAGE FOR CXU COVARIANCE ARRAY
               CXV1 STORAGE FOR CXV COVARIANCE ARRAY
               CXXS1 STORAGE FOR CXXS COVARIANCE ARRAY
               DUM2 ARRAY OF EIGENVECTORS
               DUM3 ARRAY OF EIGENVALUES
               DUM B DOT T AND B DOT R COVARIANCE MATRIX
               EGVCT ARRAY OF EIGENVECTORS
               EGVL ARRAY OF EIGENVALUES
               ICODE INTERNAL CONTROL FLAG
               IPR STORAGE FOR IPRINT
               OUT ARRAY OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENTS
               PEIG MATRIX WHOSE HYPERELIPSOID IS TO BE COMPUTED
               PS1 STORAGE FOR PS COVARIANCE ARRAY
               P1 STORAGE FOR P COVARIANCE ARRAY
               RF NOMINAL SPACECRAFT STATE AT TIME TPT
               TPT TIME TO WHICH PREDICTION IS TO BE MADE
               VEIG INTERMEDIATE VECTOR

COMMON COMPUTED/USED: CXSU CXSV CXU CXV CXXS
                       IPRINT NPE PS P
<table>
<thead>
<tr>
<th>COMMON COMPUTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON USED</td>
</tr>
<tr>
<td>DELTM TRTM1 XI</td>
</tr>
<tr>
<td>EM FOP FOV IEG IHYP1</td>
</tr>
<tr>
<td>ISTMC NDIM1 NOIM2 NOIM3 AGE</td>
</tr>
<tr>
<td>NTMC ONE Q IPT2 TSOII</td>
</tr>
<tr>
<td>UG V0 XF</td>
</tr>
</tbody>
</table>
Subroutine FRED executes a prediction event in the error analysis mode. Subroutine FRED differs from subroutine PRESIM in two respects. First, the propagated knowledge covariance partitions are based on the (most recent) targeted nominal, rather than on the most recent nominal as in PRESIM. And second, estimated position/velocity deviations are not propagated in FRED since estimates are processed only in the simulation mode, and not in the error analysis mode. See subroutine PRESIM for further analytical details. A flow chart for FRED is not presented here since it is but a subset of the PRESIM flow chart.
SUBROUTINE PRELIM

PURPOSE TO PERFORM THE PRELIMINARY WORK ASSOCIATED WITH THE
NORMAL PROGRAM INCLUDING THE READING OF THE INPUT DATA,
INITIALIZATION OF CONSTANTS, AND THE COMPUTATION OF A
ZERO ITERATE IF REQUIRED

CALLING SEQUENCE: CALL PRELIM

SUBROUTINES SUPPORTED: NOMINAL

SUBROUTINES REQUIRED: CPWMS TIME ZERIT

LOCAL SYMBOLS:

OF JULIAN DATE CORRESPONDING TO KALF ARRAY
DI JULIAN DATE CORRESPONDING TO KALI ARRAY
GS ARRAY OF VALUES OF SECONDS CORRESPONDING
TO KALG ARRAY
I INDEX
J INDEX
KALF CALENDAR DATE OF FINAL TRAJECTORY TIME
KALG ARRAY OF CALENDAR DATES OF GUIDANCE EVENTS
KALI CALENDAR DATE OF INITIAL TRAJECTORY TIME
KALT ARRAY OF CALENDAR DATES OF TARGET TIMES
KEY LOCAL VARIABLE USED TO COMPLETE
INFORMATION IN THE ARRAY
SF SECONDS OF FINAL TRAJECTORY TIME
SI SECONDS OF INITIAL TRAJECTORY TIME
TS SECONDS OF TARGET TIMES CORRESPONDING TO
KALT ARRAY

COMMON COMPUTED/USED:

AC ALMGTH DG D1 FI
IBADS IBARY ICOORD IFINT IPRE
ISTRAT IZERO KGYD KMXQ KOAST
KTIM KTYP LTARG LVLS MAT
MAXB MDL NBOD NB NCPR
MOGYD MOIT NPRAR ONE PERV
PHILS PSI1 PSI2 RIN RPRAT
RP SIGNAL SPHFACT SSS TAR
THEDOT THELS TING TIM1 TIM2
TIN TMPR TM ZDAT
<table>
<thead>
<tr>
<th>COMMON COMPUTED</th>
<th>DINTG</th>
<th>EIGHT</th>
<th>FIVE</th>
<th>FOUR</th>
<th>HALF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IEPHEM</td>
<td>IPRINT</td>
<td>KSICA</td>
<td>KUR</td>
<td>NBODYI</td>
</tr>
<tr>
<td></td>
<td>NINETY</td>
<td>RAO</td>
<td>TEN</td>
<td>THREE</td>
<td>TMU</td>
</tr>
<tr>
<td></td>
<td>TRTH</td>
<td>TWO</td>
<td>ZERO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMON USED</td>
<td>ACKT</td>
<td>DELV</td>
<td>DT</td>
<td>DVMAX</td>
<td>IBAST</td>
</tr>
<tr>
<td></td>
<td>KTAR</td>
<td>LEVELS</td>
<td>MAXBAD</td>
<td>NITS</td>
<td>NLP</td>
</tr>
<tr>
<td></td>
<td>NTP</td>
<td>PHI</td>
<td>PMASS</td>
<td>TMS</td>
<td>TOL</td>
</tr>
</tbody>
</table>
PRELIM Analysis

PRELIM is responsible for the preliminary work required by NOMINAL including the initialization of variables, the reading of input, and the computation of zero iterate values for initial time, position, and velocity if necessary.

On the first call to PRELIM, PRELIM presets constants to be used on the entire series of runs. These constants include the double precision numbers and the launch profile parameters. On subsequent calls these variables are not reset.

PRELIM then preset constants for individual runs. These constants presently include most of the guidance event parameters. The user may easily change the two sets of constants for his particular needs.

PRELIM then accepts the input data. It reads data in the NAMELIST format.

Target times must be read in as calendar dates. PRELIM next converts these to Julian date referenced 1900 and stores the converted values in the TAR array.

If the flag XZERO is nonzero, ZERIT is called for the computation of the zero iterate values of initial time, position, and velocity. ZERIT in turn calls HELIO for interplanetary trajectories and LUNA for lunar trajectories.

PRELIM then converts guidance event times referenced to initial time to calendar data and converts times read in as calendar dates to times referenced to the initial time. When the latter is done, it sets KTM to acknowledge that conversion.

Finally PRELIM records all pertinent data.
PRELIM Flow Chart

ENTER

IF IPRE = ?

0

Preset constants for series of runs

Preset constants for current run.

Read input data.

Convert calendar dates of target times to J.D. and store in TAR array.

IF IZERO = ?

0

Call ZERIT

Convert guidance times ref'd to initial time to calendar date and vice versa.

Record data.

RETURN
SUBROUTINE PREPUL

PURPOSE: TO PERFORM THE PRELIMINARY COMPUTATIONS REQUIRED FOR THE PULSING ARC MODEL.

CALLING SEQUENCE: CALL PREPUL(RIN, DELTAV, D1)

ARGUMENTS:
- RIN(6)  I INERTIAL STATE OF SPACECRAFT AT NOMINAL TIME OF CORRECTION
- DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED
- D1    I JULIAN DATE OF NOMINAL TIME OF CORRECTION

SUBROUTINES SUPPORTED: EXCUT E EXCUTS

SUBROUTINES REQUIRED: TIME

LOCAL SYMBOLS:
- A  SEMIMAJOR AXIS
- C  INTERMEDIATE VARIABLE IN F AND G SERIES
- OB JULIAN DATE AT BEGINNING OF PULSING ARC
- DELVM MAGNITUDE OF TOTAL IMPULSIVE CORRECTION
- D1E JULIAN DATE AT END OF PULSING ARC
- DVFM MAGNITUDE OF FINAL PULSE OF SEQUENCE
- DVIM MAGNITUDE OF TYPICAL PULSE OF SEQUENCE
- D  INTERMEDIATE VARIABLE IN F AND G SERIES
- G  GRAVITATIONAL CONSTANT OF BODY UNDER CONSIDERATION
- ID CALENDAR DATE OF CRITICAL TIMES FOR OUTPUT
- MAXP MAXIMUM NUMBER OF PULSES ALLOWED
- NDX ARRAY OF CODES OF LAUNCH AND TARGET BODIES
- NX INDEX OF GIVEN PLANET COORDINATES IN F-ARRAY
- RD TIME DERIVATIVE OF RADIUS MAGNITUDE OF PLANET
- RR MAGNITUDE OF RADIUS
- SD SECONDS OF CRITICAL TIMES FOR OUTPUT
<table>
<thead>
<tr>
<th>VV</th>
<th>SPEED OF PLANET</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON COMPUTED/USED1</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>GS</td>
</tr>
<tr>
<td>COMMON USED1</td>
<td>ALNGTM</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>NTP</td>
</tr>
<tr>
<td></td>
<td>TM</td>
</tr>
</tbody>
</table>
PREFUL Analysis

PREFUL is responsible for performing the preliminary computations required for the pulsing arc model.

PREFUL first determines the nominal pulsing arc. Let the following definitions be made:

- \( T \) magnitude of pulsing engine thrust
- \( m \) nominal mass of spacecraft
- \( \Delta t \) duration of single pulse
- \( \Delta t_i \) time interval between pulses
- \( \Delta v \) total velocity increment to be added

The velocity increment imparted by a single pulse is

\[
\Delta v_i = \frac{T \Delta t}{m}
\]  

The number of pulses required is then

\[
N_p = \left\lceil \frac{\Delta v}{\Delta v_i} \right\rceil + 1
\]  

where \( \lceil \cdot \rceil \) denotes the greatest integer function. The magnitude of the final pulse must be set to

\[
\Delta v_f = \Delta v - (N_p - 1) \Delta v_i
\]

The vector nominal pulse and final pulse are therefore given by

\[
\vec{\Delta v}_i = \frac{\Delta v}{\Delta v_i}
\]
\[
\vec{\Delta v}_f = \frac{\Delta v}{\Delta v}
\]

The duration of the pulsing arc is then given by

\[
\Delta T = (N_p - 1) \Delta t_i
\]

Later computations require time histories of the position vectors of the launch and target bodies. An efficient means of obtaining this involves the \( f \) and \( g \) series. Given the state \( \vec{r}_o, \vec{v}_o \) of body moving in a conic section about a central body of gravitational constant \( \mu \), the position vector as a function of \( t \) measured from the initial time is given by

\[
\vec{r}(t) = f(t) \vec{r}_o + g(t) \vec{v}_o
\]
where

\[ f(t) = \sum_{k=0}^{n} f_k t^k \quad g(t) = \sum_{k=1}^{n} g_k t^k \]  \hspace{1cm} (7)

The constants \( f_k \), \( g_k \) are computed in PREPUL as

\[
\begin{align*}
  f_0 &= 1 \\
  f_1 &= 0 \\
  f_2 &= -\frac{\mu}{r_o^3} \\
  f_3 &= \frac{\mu \dot{r}_o}{2r_o} \\
  f_4 &= \frac{\mu^2}{24r_o^6} \left( 4 - 15 \frac{r_o \dot{r}_o^2}{\mu} - 3 \frac{\dot{r}_o}{a} \right) \\
  f_5 &= -\frac{\mu^2 \ddot{r}_o}{8r_o^7} \left( 4 - \frac{7r_o \dot{r}_o^2}{\mu} - 3 \frac{\dot{r}_o}{a} \right) \\
  f_6 &= \frac{\mu^3}{720r_o^9} \left[ -90 + 114 \frac{r_o \dot{r}_o^2}{\mu} + 840 \frac{r_o \dot{r}_o^2}{\mu a} - 630 \frac{r_o \dot{r}_o^2}{\mu a} - 450 \left( \frac{r_o \dot{r}_o^2}{\mu} \right)^2 - 45 \frac{r_o \dot{r}_o^2}{a^2} \right] \\
  g_1 &= 1 \\
  g_2 &= 0 \\
  g_3 &= \frac{1}{3} f_2 \\
  g_4 &= \frac{1}{2} f_3 \\
  g_5 &= \frac{3}{5} f_4 - \frac{1}{15} f_2^2 \\
  g_6 &= \frac{2}{3} f_5 - \frac{1}{6} f_3 f_2
\end{align*}
\]

Compute pulsing arc data $N_p, \Delta T, \Delta \vec{v}_f, \Delta \vec{v}_f$

Compute current state $\vec{r}_0, \vec{v}_0$ of launch and target bodies

Compute $f$ and $g$ series for launch and target bodies

RETURN
PROGRAM PRESIM

PURPOSE CONTROL EXECUTION OF A PREDICTION EVENT IN THE SIMULATION PROGRAM

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL DYNOS HYELS JACOBI NAVM NTMS PSIM STMPR

LOCAL SYMBOLS

CXSU1 STORAGE FOR CXSU COVARIANCE ARRAY
CXSV1 STORAGE FOR CXSV COVARIANCE ARRAY
CXU1 STORAGE FOR CXU COVARIANCE ARRAY
CXV1 STORAGE FOR CXV COVARIANCE ARRAY
CXXS1 STORAGE FOR CXXS COVARIANCE ARRAY
DM2 ARRAY OF EIGENVECTORS
DM3 ARRAY OF EIGENVALUES
DM DOT T AND DOT R COVARIANCE MATRIX
EGVCT ARRAY OF EIGENVECTORS
EGVL ARRAY OF EIGENVALUES
IPR STORAGE FOR IPRINT
OUT ARRAY OF STANDARD DEVIATIONS AND CORRELATION COEFFICIENTS
PEIG MATRIX WHOSE HYPERELLIPSOID IS TO BE COMPUTED
PS1 STORAGE FOR PS COVARIANCE ARRAY
P1 STORAGE FOR P COVARIANCE ARRAY
RF1 MOST RECENT NOMINAL SPACECRAFT STATE AT TIME TP12
TPT TIME TO WHICH PREDICTION IS TO BE MADE
VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED: CXSU CXSV CXU CXV CXXS ICODE IPRINT MPE PS P RII
**PRESIM Analysis**

Subroutine PRESIM executes a prediction event in the simulation mode. In a prediction event the knowledge covariance partitions and the estimated position/velocity deviations from the most recent nominal trajectory are propagated forward to $t_p$, the time to which the prediction is to be made. The knowledge covariance partitions are propagated using the prediction equations found in the NAVH Analysis section. The estimate is propagated using the equation

$$\delta \tilde{x}_p = \Phi(t_p, t_j) \delta \tilde{x}_j + \Theta_{\tilde{x}x}(t_p, t_j) \delta \tilde{x}_{\tilde{x}}$$

where $\Phi$ and $\Theta_{\tilde{x}x}$ are the state transition matrix partitions over the time interval $[t_j, t_p]$.

The position and velocity partitions of the propagated knowledge covariance are diagonalized at time $t_p$ and the eigenvalues, eigenvectors, and hyper-ellipsoids are computed. If a guidance event has occurred previously, so that the matrix $M$ relating B-T and B-R variations to position and velocity deviations at sphere of influence is available, and if $t_p$ is within one day of the time $t_{SI}$ at which the sphere of influence is encountered, then the B-T and B-R covariance matrix is also computed.
Write out actual dynamic noise.
Write out estimated and actual deviations from most recent nominal at prediction event time \( t_j \).

Increment prediction event counter and obtain time \( t_p \) to which prediction is to be made.

Save all knowledge covariance partitions at \( t_j \).

Call NTNS to compute the most recent nominal trajectory at time \( t_p \).
Call PSIM to compute the state transition matrix partitions over the time interval \( [t_j, t_p] \).

Call DYNAS to compute the dynamic noise covariance matrix for the interval \( [t_j, t_p] \).
Write out the state transition matrix partitions and the dynamic noise covariance matrix.

Call NAVM to propagate knowledge covariance partitions forward to time \( t_p \). Write out the knowledge correlation matrix partitions and standard deviations at time \( t_p \).
Compute the predicted position/velocity deviation at time \( t_p \). Write out these deviations and the most recent nominal trajectory at \( t_p \).

Call JACOBI and HYELS to compute the eigenvalues, eigenvectors, and hyper-ellipsoids of the position and velocity partitions of the knowledge covariance matrix at \( t_p \). Write out these results.

Have any guidance events occurred and is \( t_p \) within one day of \( t_\text{SI} \)?

- NO
- YES

Compute the correlation matrix and standard deviations of \( B\cdot T \) and \( B\cdot R \) at \( t_\text{SI} \). Compute the associated eigenvalues and eigenvectors. Write out these results.

Reset targeted and most recent nominal states in preparation for next cycle. Restore saved values of all knowledge covariance partitions at time \( t_j \).

RETURN
SUBROUTINE PRINT

PURPOSE: TO PRINT THE VIRTUAL MASS INFORMATION SPECIFIED.

CALLING SEQUENCE: CALL PRINT

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: TIME TRAPAR NEWPGE SPACE

LOCAL SYMBOLS: D INTERMEDIATE VARIABLE USED FOR PRINTOUT
IDAY DAY OF CALENDAR DATE OF CURRENT TIME
IMHR HOUR OF CALENDAR DATE OF CURRENT TIME
INCMNT CURRENT TOTAL INCREMENTS FOR PRINTOUT
IP CODE OF I-TH PLANET FOR PRINTOUT PURPOSES
IYR YEAR OF CALENDAR DATE OF CURRENT TIME
MIN MINUTES OF CALENDAR DATE OF CURRENT TIME
MO MONTH OF CALENDAR DATE OF CURRENT TIME
RP RADIUS OF I-TH PLANET RELATIVE TO INERTIAL FRAME
RS RADIUS OF VEHICLE RELATIVE TO INERTIAL FRAME
RV RADIUS OF VIRTUAL MASS RELATIVE TO INERTIAL FRAME
SEC SECONDS OF CALENDAR DATE OF CURRENT TIME
TMP POSITION AND VELOCITY OF VIRTUAL MASS RELATIVE TO PLANETS
VMR MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO VIRTUAL MASS
VP MAGNITUDE OF VELOCITY OF I-TH PLANET FOR PRINTOUT PURPOSES
VS MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO INERTIAL FRAME
VSP MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO I-TH PLANET FOR PRINTOUT PURPOSES
VW MAGNITUDE OF VELOCITY OF VIRTUAL MASS
<table>
<thead>
<tr>
<th>COMMON COMPUTED/USED:</th>
<th>F</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON USED:</td>
<td>INCMNT</td>
<td>IPRT</td>
</tr>
<tr>
<td></td>
<td>PLANET</td>
<td>ZERO</td>
</tr>
</tbody>
</table>
SUBROUTINE PRINT3

PURPOSE: TO PRINT THE PERTINENT INFORMATION AT THE END OF EACH MEASUREMENT.

CALLING SEQUENCE: CALL PRINT3(MMCODE, NR)

ARGUMENTS:

MMCODE I MEASUREMENT CODE

NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX M

SUBROUTINES SUPPORTED: ERRAN

SUBROUTINES REQUIRED: CORREL EPHEM ORB SYMPR TRAPAR

LOCAL SYMBOLS:

D INTERMEDIATE DATE
J1 STATION NUMBER
D3 JULIAN DATE OF INITIAL TIME
D4 JULIAN DATE OF FINAL TIME
IDAY CALENDAR DAY OF FINAL TIME
IHR CALENDAR HOUR OF FINAL TIME
IMIN CALENDAR MINUTE OF FINAL TIME
IMO CALENDAR MONTH OF FINAL TIME
ITEMP INTERMEDIATE VARIABLE
IYR CALENDAR YEAR OF FINAL TIME
LDAY CALENDAR DAY OF INITIAL TIME
LHR CALENDAR HOUR OF INITIAL TIME
LMIN CALENDAR MINUTES OF INITIAL TIME
LMO CALENDAR MONTH OF INITIAL TIME
LYR CALENDAR YEAR OF INITIAL TIME
M NUMBER OF MEASUREMENT
RME GEOCENTRIC RADIUS OF VEHICLE
RMP DISTANCE OF VEHICLE FROM TARGET PLANET
SECI CALENDAR SECONDS OF FINAL TIME
<table>
<thead>
<tr>
<th>SECL</th>
<th>CALENDAR SECONDS OF INITIAL TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRTM2</td>
<td>TRAJECTORY TIME AT END OF INTERVAL</td>
</tr>
<tr>
<td>VME</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH</td>
</tr>
<tr>
<td>VMP</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET</td>
</tr>
</tbody>
</table>

### COMMON COMPUTED/USED:

<table>
<thead>
<tr>
<th>NO</th>
<th>RE</th>
<th>RTP</th>
<th>XP</th>
</tr>
</thead>
</table>

### COMMON USED:

- AK
- CXSU
- CVVP
- DELM
- IPROB
- NDIM1
- PSP
- S
- XF
- XV
SUBROUTINE PRINT4

PURPOSE: THIS SUBROUTINE PRINTS RELEVANT DATA AT THE END OF EACH MEASUREMENT IN THE SIMULATION MODE

CALLING SEQUENCE: CALL PRINT4(MMCODE,NR)

ARGUMENTS: MMCODE I MEASUREMENT CODE
            NR I NUMBER OF ROWS IN THE OBSERVATION MATRIX

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL EPHEN ORB STMPR SUB1
                      TRAPAR

LOCAL SYMBOLS: ADON ACTUAL STATE DEVIATION FROM TARGETED NOMINALTrajectory
                AODI ACTUAL ORBIT ESTIMATION ERROR
                D INTERMEDIATE DATE
                EDON ESTIMATED STATE DEVIATION FROM TARGETED NOMINAL Trajectory
                IA STATION NUMBER
                IB STAR/PLANET ANGLE NUMBER
                M MEASUREMENT NUMBER
                ROW ARRAY OF CORRELATION COEFFICIENTS
                SQP VECTOR OF STANDARD DEVIATIONS
                TRIM2 TRAJECTORY TIME AT END OF INTERVAL
                XP1 POSITION AND VELOCITY OF EARTH AT TRIM1
                XP2 POSITION AND VELOCITY OF EARTH AT TRIM2
                XE1 POSITION AND VELOCITY OF EARTH AT TRIM1
                XE2 POSITION AND VELOCITY OF EARTH AT TRIM2

COMMON COMPUTED/ USED: NO

COMMON USED: ADEVXS ADEVX AK ALIGNED AL
              AN APONIS AR AY GXSUP
<table>
<thead>
<tr>
<th>CXSU</th>
<th>CXVP</th>
<th>CXSV</th>
<th>CXUP</th>
<th>CXU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXVP</td>
<td>CXV</td>
<td>CXSP</td>
<td>CXS</td>
<td>DATEJ</td>
</tr>
<tr>
<td>DELTM</td>
<td>EDEVXS</td>
<td>EDEVX</td>
<td>EY</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>HPHR</td>
<td>H</td>
<td>IBARY</td>
<td>IPR08</td>
</tr>
<tr>
<td>IPRT</td>
<td>MGMTR</td>
<td>MB00</td>
<td>NB</td>
<td>NDM1</td>
</tr>
<tr>
<td>NDIM2</td>
<td>NDIM3</td>
<td>NTP</td>
<td>PP</td>
<td>PSP</td>
</tr>
<tr>
<td>PS</td>
<td>P</td>
<td>Q</td>
<td>RES</td>
<td>R</td>
</tr>
<tr>
<td>S</td>
<td>TM</td>
<td>TRTM1</td>
<td>UQ</td>
<td>V0</td>
</tr>
<tr>
<td>M</td>
<td>XF1</td>
<td>XF</td>
<td>XI1</td>
<td>XI</td>
</tr>
<tr>
<td>XLAB</td>
<td>XP</td>
<td>XSL</td>
<td>XU</td>
<td>XV</td>
</tr>
<tr>
<td>ZF</td>
<td>ZI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM PRNTS3

PURPOSE: TO PRINT A SUMMARY OF THE ERROR ANALYSIS MODE

SUBROUTINES SUPPORTED: ERROR

SUBROUTINES REQUIRED: CORREL TIME

LOCAL SYMBOLS:

D0  MOLLERITH LABEL INITIAL
D9  MOLLERITH LABEL FINAL
D1  JULIAN DATE, EPOCH JAN. 0, 1900, OF INITIAL TIME
D2  JULIAN DATE, EPOCH JAN. 0, 1900, OF FINAL TIME
D3  JULIAN DATE OF INITIAL TIME
D4  JULIAN DATE OF FINAL TIME
F  FUNCTION= SQUARE ROOT OF SUM OF 3 SQUARES
IDAY  CALENDAR DAY OF FINAL TIME
IHR  CALENDAR HOUR OF FINAL TIME
IMIN  CALENDAR MINUTES OF FINAL TIME
IMO  CALENDAR MONTH OF FINAL TIME
IYR  CALENDAR YEAR OF FINAL TIME
LHAY  CALENDAR DAY OF INITIAL TIME
LHR  CALENDAR HOUR OF INITIAL TIME
LMIN  CALENDAR MINUTES OF INITIAL TIME
LMO  CALENDAR MONTH OF INITIAL TIME
LYR  CALibAND YEAR OF INITIAL TIME
RI  POSITION AND VELOCITY OF VEHICLE AT INITIAL TIME
RMF  HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME
RMI  HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEC1</td>
<td>Calendar seconds of final time</td>
</tr>
<tr>
<td>SEC1L</td>
<td>Calendar seconds of initial time</td>
</tr>
<tr>
<td>TRTM2</td>
<td>Trajectory time at end of trajectory</td>
</tr>
<tr>
<td>VE</td>
<td>Position and velocity of vehicle relative to Earth at final time</td>
</tr>
<tr>
<td>VMF</td>
<td>Magnitude of velocity of vehicle at final time</td>
</tr>
<tr>
<td>VMI</td>
<td>Magnitude of velocity of vehicle at initial time</td>
</tr>
<tr>
<td>VT</td>
<td>Position and velocity of vehicle relative to target planet at final time</td>
</tr>
</tbody>
</table>

**Common Computed/Used**

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
</tr>
<tr>
<td>DSI</td>
</tr>
</tbody>
</table>

**Common Computed**

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRTM1</td>
</tr>
</tbody>
</table>

**Common Used**

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCMD</td>
</tr>
<tr>
<td>ACC</td>
</tr>
<tr>
<td>ALNGTH</td>
</tr>
<tr>
<td>BDRSI1</td>
</tr>
<tr>
<td>BDTSI1</td>
</tr>
<tr>
<td>BSI1</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>CXSUB</td>
</tr>
<tr>
<td>CXSU</td>
</tr>
<tr>
<td>CXSVB</td>
</tr>
<tr>
<td>CXV</td>
</tr>
<tr>
<td>CXVSB</td>
</tr>
<tr>
<td>CXXS</td>
</tr>
<tr>
<td>DATEJ</td>
</tr>
<tr>
<td>DELTH</td>
</tr>
<tr>
<td>DELX</td>
</tr>
<tr>
<td>DNTM</td>
</tr>
<tr>
<td>DTMAY</td>
</tr>
<tr>
<td>FACP</td>
</tr>
<tr>
<td>FACV</td>
</tr>
<tr>
<td>FATH</td>
</tr>
<tr>
<td>IAGIM</td>
</tr>
<tr>
<td>IDMF</td>
</tr>
<tr>
<td>IEPHEM</td>
</tr>
<tr>
<td>IMNF</td>
</tr>
<tr>
<td>IPROB</td>
</tr>
<tr>
<td>ISPM</td>
</tr>
<tr>
<td>ISTMC</td>
</tr>
<tr>
<td>ISTM1</td>
</tr>
<tr>
<td>MNCN</td>
</tr>
<tr>
<td>MNNAME</td>
</tr>
<tr>
<td>NDACC</td>
</tr>
<tr>
<td>NDIM1</td>
</tr>
<tr>
<td>NDIM2</td>
</tr>
<tr>
<td>NDIM3</td>
</tr>
<tr>
<td>NEV1</td>
</tr>
<tr>
<td>NEV2</td>
</tr>
<tr>
<td>NEV3</td>
</tr>
<tr>
<td>NEV</td>
</tr>
<tr>
<td>NMN</td>
</tr>
<tr>
<td>NST</td>
</tr>
<tr>
<td>NTMC</td>
</tr>
<tr>
<td>NTP</td>
</tr>
<tr>
<td>PB</td>
</tr>
<tr>
<td>PLANET</td>
</tr>
<tr>
<td>PSB</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>RCA1</td>
</tr>
<tr>
<td>RE</td>
</tr>
<tr>
<td>RSO11</td>
</tr>
<tr>
<td>RTP</td>
</tr>
<tr>
<td>SAL</td>
</tr>
<tr>
<td>SIGALP</td>
</tr>
<tr>
<td>SIGBET</td>
</tr>
<tr>
<td>SIGPRO</td>
</tr>
<tr>
<td>SIGRES</td>
</tr>
<tr>
<td>SLAT</td>
</tr>
<tr>
<td>SLOM</td>
</tr>
<tr>
<td>TCA1</td>
</tr>
<tr>
<td>TM</td>
</tr>
<tr>
<td>TRTM1</td>
</tr>
<tr>
<td>TSO1I</td>
</tr>
<tr>
<td>UST</td>
</tr>
<tr>
<td>UO</td>
</tr>
<tr>
<td>VSO1I</td>
</tr>
<tr>
<td>VST</td>
</tr>
<tr>
<td>VO</td>
</tr>
<tr>
<td>WST</td>
</tr>
<tr>
<td>XB</td>
</tr>
<tr>
<td>XNM</td>
</tr>
</tbody>
</table>
PROGRAM PRNTS4

PURPOSE: TO PRINT OUT A SUMMARY OF THE SIMULATION MODE

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: CORREL EPHEN ORB TIME NOMINAL

LOCAL SYMBOLS: ADON ACTUAL STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY AT FINAL TIME

AODI ACTUAL ORBIT ESTIMATION ERROR AT FINAL TIME

BLANK BLANK HOLLERITH CHARACTER

D1 JULIAN DATE, EPOCH JAN. 0, 1900, OF INITIAL TIME

D2 JULIAN DATE, EPOCH JAN. 0, 1900, OF FINAL TIME

D3 JULIAN DATE OF INITIAL TIME

D4 JULIAN DATE OF FINAL TIME

EDON ESTIMATED STATE DEVIATION FROM TARGETED NOMINAL TRAJECTORY

IDAY CALENDAR DAY OF FINAL TIME

IMH CALENDAR HOUR OF FINAL TIME

INMIN CALENDAR MINUTES OF FINAL TIME

IMO CALENDAR MONTH OF FINAL TIME

IYR CALENDAR YEAR OF FINAL TIME

IDAY CALENDAR DAY OF INITIAL TIME

LHR CALENDAR HOURS OF INITIAL TIME

LMIN CALENDAR MINUTES OF INITIAL TIME

LMO CALENDAR MONTH OF INITIAL TIME

LYR CALENDAR YEAR OF INITIAL TIME

RE1 POSITION AND VELOCITY OF VEHICLE RELATIVE TO EARTH ON TARGETED NOMINAL
RE2  POSITION AND VELOCITY OF VEHICLE RELATIVE TO TO EARTH ON MOST RECENT NOMINAL
RE3  POSITION AND VELOCITY OF VEHICLE RELATIVE TO TO EARTH ON ACTUAL TRAJECTORY
RME1 GEOCENTRIC RADIUS OF VEHICLE ON TARGETED NOMINAL AT FINAL TIME
RME2 GEOCENTRIC RADIUS OF VEHICLE ON MOST RECENT NOMINAL AT FINAL TIME
RME3 GEOCENTRIC RADIUS OF VEHICLE ON ACTUAL TRAJECTORY AT FINAL TIME
RME GEOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RMP1 DISTANCE OF VEHICLE FROM TARGET PLANET ON TARGETED NOMINAL AT FINAL TIME
RMP2 DISTANCE OF VEHICLE FROM TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME
RMP3 DISTANCE OF VEHICLE FROM TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME
RMP DISTANCE OF VEHICLE FROM TARGET PLANET AT INITIAL TIME
RMS1 HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON TARGETED NOMINAL
RMS2 HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL
RMS3 HELIOCENTRIC RADIUS OF VEHICLE AT FINAL TIME ON ACTUAL TRAJECTORY
RMS HELIOCENTRIC RADIUS OF VEHICLE AT INITIAL TIME
RP1 STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON TARGETED NOMINAL
RP2 STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON MOST RECENT NOMINAL
RP3 STATE OF VEHICLE RELATIVE TO TARGET PLANET AT FINAL TIME ON ACTUAL TRAJECTORY
SECI CALENDAR SECONDS AT FINAL TIME
<table>
<thead>
<tr>
<th>SECL</th>
<th>CALENDAR SECONDS AT INITIAL TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VME</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH AT INITIAL TIME</td>
</tr>
<tr>
<td>VME1</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON TARGETED NOMINAL AT FINAL TIME</td>
</tr>
<tr>
<td>VME2</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON MOST RECENT NOMINAL AT FINAL TIME</td>
</tr>
<tr>
<td>VME3</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO EARTH ON ACTUAL TRAJECTORY AT FINAL TIME</td>
</tr>
<tr>
<td>VMP</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET AT INITIAL TIME</td>
</tr>
<tr>
<td>VMP1</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON TARGETED NOMINAL AT FINAL TIME</td>
</tr>
<tr>
<td>VMP2</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON MOST RECENT NOMINAL AT FINAL TIME</td>
</tr>
<tr>
<td>VMP3</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE RELATIVE TO TARGET PLANET ON ACTUAL TRAJECTORY AT FINAL TIME</td>
</tr>
<tr>
<td>VMS</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE AT INITIAL TIME</td>
</tr>
<tr>
<td>VMS1</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON TARGETED NOMINAL</td>
</tr>
<tr>
<td>VMS2</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON MOST RECENT NOMINAL TRAJECTORY</td>
</tr>
<tr>
<td>VMS3</td>
<td>MAGNITUDE OF VELOCITY OF VEHICLE AT FINAL TIME ON ACTUAL TRAJECTORY</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED:** NO RE RTP XP ZI

**COMMON USED:**

AALP ABET ACCAD ACC1 ACC ADEVX ADEYX ADEYX ALNGTH APRO ARES AVARM BDRSI1 BDRSI2 BDRSI3 BDTSI1 BDTSI2 BDTSI3 BS11 BS12
BSI3  CXSU  CXSUb  CXSV  CXSVB
CXSV  CXUB  CXU  CXVB  CXV
CXXSB  CXXS  DAB  DATEJ  DEB
DELHUP  DELAUS  DELX  DIB  DMAB
DMUPB  DMUSB  DNCN  DNOB  DTMAX
DN8  DENVX  DENVX  FAC  FAC
FNTH  F  H  IAMNF  IAUGIN
IBARY  IDNF  IMNF  IPROB  ISOI1
ISO12  ISO13  ISTMC  ISTM1  MNCH
MNNAME  NBOD1  NBOD  NB1  NB
NDACC  NDIM1  NDIM2  NDIM3  NEV1
NEV2  NEV3  NEV5  NEV  NMN
NTMC  NTP  PB  PLANET  PSB
PS  P  RCA1  RCA2  RCA3
RSOI1  RSOI2  RSOI3  SAL  SIGALP
SIGBET  SIGPRO  SIGRES  SLAT  SLOH
TCAI  TCA2  TCA3  TM  TRTMB
TSOI1  TSOI2  TSOI3  TTIM1  TTIM2
UNMAC  UST  UO  VSOI1  VSOI2
VSOI3  VST  VQ  WST  XB
XF1  XF  XLAB  XMM  ZF
### SUBROUTINE PSIM

**Purpose**: To compute the state transition matrix partitions \( \Phi \), \( \theta \), and \( \tau \) over an arbitrary interval of time \((t_k, t_{k+1})\).

**Calling Sequence**: CALL PSIM\( (RI, RF, ISC) \)

**Arguments**:
- **ISC** (I): Code specifying which technique is to be used to compute the state transition matrix partition \( \Phi \)
- **RF** (I): Position and velocity of the vehicle at the end of the time interval
- **RI** (I): Position and velocity of the vehicle at the beginning of the time interval

**Subroutines Supported**: SIMU, SETEV, BIAIM, GUISIM, GUSS, PR, MSIM, ERMANN, SETEYN, GUID, GUID

**Subroutines Required**: CASCA, CONC, EPHEM, MUND, NDTM, ORB, PCTM, PLNO

**Local Symbols**:
- **D**: Intermediate Julian date
- **DELT**: Time interval in correct units
- **DUM**: Temporary storage for state transition matrix
- **IANS**: Variable used in examining IAUG, IAUG, IAIN
- **POSS**: Distance of the vehicle from the target planet at initial time
- **RS**: Position of vehicle relative to governing body at initial time
- **THSP**: Constant equal to six times the sphere of influence of target planet
- **VEC**: Position and velocity of vehicle relative to target planet at initial time
- **VS**: Velocity of vehicle relative to governing body at initial time

**Common Computed/Used**:
- \( \Phi \), \( \tau \), \( \theta \)
- **XP**: No

**Common Computed**:
- \( \Phi \), \( \tau \), \( \theta \)
<table>
<thead>
<tr>
<th>COMMON USED1</th>
<th>ALNGTH</th>
<th>DATEJ</th>
<th>DELTM</th>
<th>DTMAX</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAUGDC</td>
<td>IAUG</td>
<td>IBARY</td>
<td>ISTM1</td>
<td>NBOD</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>NOIM1</td>
<td>NOIM2</td>
<td>NTP</td>
<td>RVS</td>
<td></td>
</tr>
<tr>
<td>SPHERE</td>
<td>TM</td>
<td>TRTM1</td>
<td>VMU</td>
<td>ZERO</td>
<td></td>
</tr>
</tbody>
</table>
The first part of the subroutine deals solely with the computation of $\Phi(k+1,k)$ by one of three techniques: analytical patched conic, analytical virtual mass, or numerical differencing. If an analytical technique is selected for computing $\Phi(k+1,k)$ over an interval of time greater than the maximum time interval for which the analytical technique is considered valid, we compute $\Phi(k+1,k)$ using numerical differencing or by cascading Danby matricants.

The remaining partitions, $\Theta_{xx_8}$ and $\Theta_{xu}$, are always computed by numerical differencing. Columns in these partitions associated with target planet gravitational constant or orbital elements are computed only if the spacecraft is within six times the sphere of influence of the target planet at $t_{k+1}$. Otherwise, these columns are set to zero.
Zero out partitions $\phi(6x6)$, $\Theta_{xx}(6xn_1)$, and $\Theta_{xu}(6xn_2)$. 

Is $\Delta t \leq \Delta t_{max}$? 

- NO: $\text{ISC} = \text{yes}$ 
- YES: $\text{ISC} = \text{no}$ 

If $\text{ISC} = \text{yes}$, call CASCAD to compute $\phi(6x6)$ using cascaded Danby matrices. 

If $\text{ISC} = \text{no}$, call PCTM to compute $\phi(6x6)$ using the analytical patched conic technique. 

- NO: $n_2 \neq 0$? 
- YES: $n_1 = 0$? 

If $n_1 = 0$, call CONCT to compute $\phi(6x6)$ using the analytical virtual mass technique. 

If $n_2 \neq 0$, call NDTM to compute $\phi(6x6)$ using the numerical differencing technique. 

Are there any dynamic parameters in $x_u$? 

RETURN
Compute six times the sphere of influence of the target planet and the Julian date at \( t_{k+1} \).

Compute the position components of the target planet in the reference coordinate system at \( t_{k+1} \).

Compute the distance between the spacecraft and the target planet at \( t_{k+1} \).

\[ \text{Is } \mu_s \text{ or } \mu_p \text{ in } \vec{x}_s \text{ or } \vec{u} ? \]

\[ \text{YES} \]

Call MUND to compute appropriate columns of \( \Theta_{xx_s} \) and \( \Theta_{xu} \).

\[ \text{Are any target planet orbital elements in } \vec{x}_s \text{ or } \vec{u} ? \]

\[ \text{NO} \]

\[ \text{RETURN} \]

\[ \text{YES} \]

\[ \text{Is spacecraft within six times the sphere of influence of the target planet?} \]

\[ \text{NO} \]

\[ \text{RETURN} \]

\[ \text{YES} \]

Call PLND to compute appropriate columns of \( \Theta_{xx_s} \) and \( \Theta_{xu} \). Maintain appropriate columns of \( \Theta_{xx_s} \) and \( \Theta_{xu} \) zero.

\[ \text{RETURN} \]
SUBROUTINE PULCOV

PURPOSE COMPUTE EFFECTIVE EXECUTION ERROR COVARIANCE MATRIX FOR A VELOCITY CORRECTION MODELED AS AN IMPULSE SERIES

CALLING SEQUENCE: CALL PULCOV(RIN, DELTAV, TH, QK)

ARGUMENTS:  RIN(6)  I INERTIAL STATE OF SPACECRAFT AT NOMINAL TIME OF CORRECTION
            DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED
            TH  I TIME UNITS PER DAY
            QK(6,6)  O DEVIATION MATRIX RESULTING FROM EXECUTION ERRORS

SUBROUTINES SUPPORTED: EXCUT EXCUTS

SUBROUTINES REQUIRED: PERHEL QCMP

LOCAL SYMBOLS:  DELR  PERTURBATION IN POSITION
                DELV  PERTURBATION IN VELOCITY
                DVFH  MAGNITUDE OF FINAL PULSE
                DVIH  MAGNITUDE OF TYPICAL PULSE
                FSER  F-SERIES CONSTANT FOR PLANET
                GSER  G-SERIES CONSTANT FOR PLANET
                HLTF  STATES OF LAUNCH AND TARGET BODIES AT END OF PROPAGATION INTERVAL
                PERT  CURRENT PERTURBATION
                PHI  STATE TRANSITION MATRIX OVER TYPICAL INTERVAL
                QQ  DEVIATION MATRIX DURING PROPAGATION THROUGH PULSES
                Q  TYPICAL VELOCITY EXECUTION ERROR COVARIANCE
                RF  NOMINAL INERTIAL STATE OF SPACECRAFT AT END OF TYPICAL INTERVAL
                RPF  PERTURBED INERTIAL STATE OF SPACECRAFT AT END OF TYPICAL INTERVAL
                R  INERTIAL STATE OF SPACECRAFT AT BEGINNING
OF TYPICAL INTERVAL

<table>
<thead>
<tr>
<th>T1</th>
<th>TIME INTERVAL BETWEEN PULSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>T1**2</td>
</tr>
<tr>
<td>T3</td>
<td>T1**3</td>
</tr>
<tr>
<td>T4</td>
<td>T1**4</td>
</tr>
<tr>
<td>T5</td>
<td>T1**5</td>
</tr>
<tr>
<td>T6</td>
<td>T1**6</td>
</tr>
</tbody>
</table>

COMMON USED:

<table>
<thead>
<tr>
<th>OTI</th>
<th>DVF</th>
<th>DVI</th>
<th>FS</th>
<th>GG</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td>NPUL</td>
<td>ONE</td>
<td>PSIGA</td>
<td>PSIGB</td>
</tr>
<tr>
<td>PSIGK</td>
<td>PSIGS</td>
<td>RK</td>
<td>TWO</td>
<td>VK</td>
</tr>
<tr>
<td>ZERO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FULCO Analysis

FULCO processes the control covariance through the pulsing arc to determine a measure of the probabilistic deviation of the corrected trajectory from the desired trajectory resulting from execution errors.

The pulsing arc itself is computed in PREPUL. It consists of \( N_{p} - 1 \) pulses \( \vec{\Delta v}_{1} \) and a final pulse \( \vec{\Delta v}_{f} \) satisfying

\[
(N_{p} - 1) \vec{\Delta v}_{1} + \vec{\Delta v}_{f} = \vec{\Delta v}
\]

where \( \vec{\Delta v} \) is the equivalent single impulse. The pulses are separated by a time interval \( \Delta t_{i} \). The duration of the entire sequence of pulses is given by \( \Delta T = (N_{p} - 1) \Delta t_{1} \).

FULCO must compute the execution error matrices \( Q, Q_{f} \) corresponding to the nominal pulse \( \vec{\Delta v}_{1} \) and the final pulse \( \vec{\Delta v}_{f} \) respectively. The error model for the engine is defined by the input specifications

\[
\begin{align*}
\sigma_{k}^{2} &= \text{proportionality error} \\
\sigma_{k}^{2} &= \text{resolution error} \\
\sigma_{\alpha}^{2} &= \text{first pointing error} \\
\sigma_{\beta}^{2} &= \text{second pointing error}
\end{align*}
\]

The execution error matrix measuring the probabilistic deviation of the actual velocity increment from the desired velocity increment is computed by QCMF.

The exact equations defining the propagation of the covariance matrix are recursive in nature. If \( P_{k}^{+} \) is the control covariance immediately after the \( k^{th} \) pulse, the covariance will propagate to the time of the next pulse \( t_{k+1} \) by the formula

\[
P_{k+1}^{-} = \Phi_{k+1,k} P_{k}^{+} \Phi_{k+1,k}^{T}
\]

where \( \Phi_{k+1,k} \) is the 6x6 state transition matrix relating perturbations at \( t_{k+1} \) to perturbations at \( t_{k} \). Adding the pulse at \( t_{k+1} \) expands the covariance by
\[ \begin{align*}
\mathbf{P}^{+}_{k+1} &= \mathbf{P}^{-}_{k+1} + \begin{bmatrix}
0 & 0 \\
-1 & -1 \\
0 & 1
\end{bmatrix} Q
\end{align*} \]

where \( Q \) is set equal either to the nominal or final form of \( Q \).

To start the process the control covariance following the first pulse is given by

\[ \mathbf{P}^{+} = \begin{bmatrix}
0 & 0 \\
-1 & -1 \\
0 & 1
\end{bmatrix} Q \]

For efficiency one simplification is made in the process. Instead of recomputing the state transition matrix over each interval, the value of that matrix is held constant at the value corresponding to the "average interval". To explain this, let the state of the spacecraft at the time \( t_0 \) of the impulsive \( \Delta \mathbf{V} \) computation be denoted \( \mathbf{r}_0, \mathbf{v}_0 \). Then the "average interval" is defined to be the perturbed heliocentric trajectory (PERHEL) resulting from the propagation of the state \(( \mathbf{r}_0, \mathbf{v}_0 + \frac{1}{2} \Delta \mathbf{V} )\) over the interval \(( t_0, t_0 + \Delta t_i )\).

The constant state transition matrix \( \Phi \) is computed by numerical differencing. The initial state \(( \mathbf{r}_0, \mathbf{v}_0 + \frac{1}{2} \Delta \mathbf{V} )\) is first propagated over the \( \Delta t_i \) time interval (using PERHEL) resulting in the state \(( \mathbf{r}_f, \mathbf{v}_f )\). Then the \( x \)-component of initial position is perturbed by \( \Delta x \), leading to a final state of \(( \mathbf{r}_f^{+}, \mathbf{v}_f^{+} )\) upon propagation. The first column of the matrix is then computed by

\[ \Phi_1 = \begin{bmatrix}
\mathbf{r}_f^{+} - \mathbf{r}_f \\
\frac{\mathbf{r}_f^{+} - \mathbf{r}_f}{\Delta x} \\
\mathbf{v}_f^{+} - \mathbf{v}_f \\
\frac{\mathbf{v}_f^{+} - \mathbf{v}_f}{\Delta x}
\end{bmatrix} \]

The other columns of \( \Phi \) are computed by similar computations using the remaining components of position and velocity \((y, z, \dot{x}, \dot{y}, \dot{z})\).
PULCOV Flow Chart

ENTER

Compute final positions of launch and target bodies at $t_0 + \Delta t_1$

Compute nominal state transition matrix $\Phi$

Compute nominal execution error matrix $Q$

Propagate control covariance to next pulse

\[ P_{k+1}^+ = \Phi P_k^+ \Phi^T \]

Last pulse?

YES

REcompute execution error matrix for last pulse $Q$

NO

Update covariance by current execution error

\[ P_{k+1}^- = P_{k+1}^+ + \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix} \]

Last Pulse?

YES

RETURN

NO

A
SUBROUTINE PULSEX

PURPOSE: TO CONTROL EXECUTION OF THE PULSING ARC MODEL.

CALLING SEQUENCE: CALL PULSEX(RIN, DELTAV, RE, TM, IRE)

ARGUMENTS:
- RIN(6) I INERTIAL STATE OF SPACECRAFT AT NOMINAL TIME OF CORRECTION
- DELTAV(3) I TOTAL VELOCITY INCREMENT TO BE ADDED
- RE(6) O FINAL INERTIAL STATE OF SPACECRAFT (IRE)
- TM I TIME UNITS PER DAY
- IRE I FLAG DETERMINING FINAL STATE
  = 0 RETURN FINAL STATE AT END OF PULSE ARC
  = 1 RETURN FINAL STATE AT ARC MIDPOINT

SUBROUTINES SUPPORTED: EXECUTE EXCUITS EXCUT

SUBROUTINES REQUIRED: CAREL PERHEL

LOCAL SYMBOLS:
- A SEMIMAJOR AXIS
- DTS TIME INTERVAL IN TIME UNITS
- DT DUMMY VARIABLE FOR OUTPUT
- E ECCENTRICITY
- FSER F-SERIES CONSTANT
- GSER G-SERIES CONSTANT
- HLTF STATES OF LAUNCH AND TARGET BODIES AT END OF PROPAGATION INTERVAL
- HLTI STATES OF LAUNCH AND TARGET BODIES AT BEGINNING OF PROPAGATION INTERVAL
- IPUL PULSE COUNTER
- PP UNIT VECTOR TO PERIAPSIS
- QQ UNIT VECTOR IN ORBITAL PLANE NORMAL TO PP
- RB INERTIAL STATE OF SPACECRAFT AT BEGINNING OF PROPAGATION INTERVAL
- TA TRUE ANOMALY
- TFP TIME FROM PERIAPSIS
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Time from start of pulsing arc</td>
</tr>
<tr>
<td>T</td>
<td>Time interval from midpoint of arc</td>
</tr>
<tr>
<td>X(M)</td>
<td>Argument of periapsis</td>
</tr>
<tr>
<td>W</td>
<td>Normal to orbital plane</td>
</tr>
<tr>
<td>D(TI)</td>
<td>Longitude of ascending node</td>
</tr>
<tr>
<td>D(V)</td>
<td>Inclination</td>
</tr>
<tr>
<td>N(P)</td>
<td>Unit normal to orbital plane</td>
</tr>
<tr>
<td>G(K)</td>
<td>Longitude of node</td>
</tr>
<tr>
<td>F(Y)</td>
<td>Time from midpoint of arc</td>
</tr>
<tr>
<td>P(Y)</td>
<td>Time interval from midpoint of arc</td>
</tr>
</tbody>
</table>

**COMMON USED VARIABLES**

- \( T \) is the time from start of pulsing arc
- \( T_1 \) is the time interval from midpoint of arc
- \( X(M) \) is the argument of periapsis
- \( W \) is the normal to the orbital plane
- \( D(TI) \) is the longitude of ascending node
- \( D(V) \) is the inclination
- \( N(P) \) is the unit normal to the orbital plane
- \( G(K) \) is the longitude of node
- \( F(Y) \) is the time from midpoint of arc
- \( P(Y) \) is the time interval from midpoint of arc
PULSEX Analysis

PULSEX is responsible for the actual execution of the pulsing arc. Experiments have shown that adding an impulsive \( \Delta \vec{v} \) at time \( t_0 \) may be approximated quite closely by centering an equivalent sequence of smaller impulses about the nominal time \( t_0 \).

This equivalent sequence of thrusts is computed by PEPUL. It consists of \( N_p - 1 \) pulses \( \Delta \vec{v}_1 \) and a final pulse \( \Delta \vec{v}_f \) satisfying

\[
(N_p - 1) \Delta \vec{v}_1 + \Delta \vec{v}_f = \Delta \vec{v}
\]

The pulses are separated by a time interval \( \Delta t_i \). The duration of the entire sequence of pulses is given by \( \Delta T = (N_p - 1) \Delta t_i \).

For efficiency the perturbed heliocentric conic propagator PERHEL is used to propagate the trajectory between pulses. PERHEL requires the positions of the launch and target bodies at the beginning and end of each propagation interval. PEPUL stores the position and velocity of the launch and target bodies at the reference time \( t_0 : (\vec{r}_{LO}, \vec{v}_{LO}) \) and \( (\vec{r}_{TO}, \vec{v}_{TO}) \) and stores the constants of the \( f \) and \( g \) series for those states \( f_{lk}, g_{lk}, k=1,6 \). The position of the launch body at some time \( t \) relative to the reference time \( t_0 \) is then given by

\[
\vec{r}_L(t) = f_L(t) \vec{r}_{LO} + g_L(t) \vec{v}_{LO}
\]

where

\[
f_L(t) = \sum_{k=0}^{6} f_{lk} t^k
\]

\[
g_L(t) = \sum_{k=1}^{6} g_{lk} t^k
\]

with similar equations holding for the target body.

The procedure of PULSEX is straightforward. The positions of the launch and target bodies are computed at the time the pulsing arc should begin: \( t_B = t_0 - \Delta T/2 \). PERHEL is then called to propagate the spacecraft from \( t_0 \) backwards to \( t_B \). The actual pulsing arc cycle is now entered. The nominal velocity increment \( \Delta \vec{v}_1 \) is added to the current velocity impulsively

\[
\vec{v} = \vec{v} + \Delta \vec{v}_1
\]
and the resulting state \( \left( \vec{r}, \vec{v} \right) \) is propagated forward over the time interval \( \Delta t_i \) by PHEREL. Another pulse is added and the process repeated until \( p - 1 \) pulses have been added. Finally a pulse of \( \Delta \vec{v}_f \) is added.

Two options are now permitted. If \( IRE = 0 \), the final state is not altered (NOMINAL). If \( IRE = 1 \), the final state is propagated backwards back to \( t_0 \) for use in ERRAN and SIMUL.

Finally CAMEL is called to compute the conic elements of the final state. For comparison purposes, the impulsive \( \Delta \vec{v} \) is added to the state at \( t_0 \), propagated to the final time \( t_f = t_0 + \Delta T/2 \) by PHEREL, and those elements computed.
PULSEX Flow Chart

ENTER

Propagate launch and target bodies to start of pulsing arc \( \vec{T}_{Lf}, \vec{T}_{Tf} \)

Propagate state of S/C back to start of pulsing arc to get \( \vec{T}_{Sf}, \vec{V}_{Sf} \)

IF \( \text{IPUL} = \text{NPUL} \)

\( \text{IPUL} = \text{IPUL} + 1 \)

Update variables from previous step:

\[
\begin{align*}
  t &= t_k + \frac{\Delta T}{2} \\
  \vec{r}_{L1} &= \vec{r}_{Lf} \\
  \vec{r}_{T1} &= \vec{r}_{Tf} \\
  \vec{r}_{S1} &= \vec{r}_{Sf} \\
  \vec{V}_{S1} &= \vec{V}_{Sf} + \Delta \vec{V}_f \\
\end{align*}
\]

Compute final positions of launch and target bodies \( \vec{r}_{Lf}', \vec{r}_{Tf} \)

Propagate S/C forward to time of next pulse: \( \vec{r}_{Sf}', \vec{V}_{Sf}' \)

Propagate \( \vec{r}, \vec{v} \) to \( t_0 + \Delta t_i/2 : \vec{r}, \vec{v} \)

Compute conics els of \((\vec{r}_{Sf}', \vec{V}_{Sf}')\) and \((\vec{r}, \vec{v})\)

RETURN
SUBROUTINE QCOMP

PURPOSE: TO COMPUTE THE EXECUTION ERROR COVARIANCE MATRIX FOR A VELOCITY CORRECTION.

CALLING SEQUENCE: CALL QCOMP(V,EM,Q)

ARGUMENTS: V I VELOCITY CORRECTION
            Q O EXECUTION ERROR MATRIX
            EM I ERROR MODEL (SIGRES,SIGPRO,SIGALP,SIGBET)

SUBROUTINES SUPPORTED: BIAIM GUISIM PULCOV GUIDM

LOCAL SYMBOLS: AU SIGALP/U2
                BRK SIGPRO+ SIGRES/R2
                BU SIGBET/U2
                R2 U2+Z2
                U2 X2+Y2
                X2 V(1) SQUARED
                Y2 V(2) SQUARED
                Z2 V(3) SQUARED
QCQMP Analysis

Subroutine QCQMP computes the execution error covariance matrix $\tilde{Q}_j$ for a velocity correction $\Delta V = (\Delta V_x, \Delta V_y, \Delta V_z)$ occurring at time $t_j$. If the execution error is assumed to have form

$$\delta \Delta V \approx k \Delta V + s \frac{\Delta V}{\Delta V} + \delta \Delta V_{\text{pointing}}$$

where $k$ is the proportionality error and $s$ is the resolution error, then the elements of the $\tilde{Q}_j$ matrix are given by

$$\tilde{Q}_{11} = \Delta V_x \left[ \sigma_k^2 + \frac{\sigma_a^2}{\rho^2} \right] + \frac{\Delta V_x^2 \rho^2}{\mu^2} \delta \alpha + \frac{\Delta V_x \Delta V_z}{\mu^2} \delta \beta$$

$$\tilde{Q}_{12} = \tilde{Q}_{21} = \Delta V_x \Delta V_y \left[ \sigma_k^2 + \frac{\sigma_b^2}{\rho^2} - \frac{\rho^2 \sigma_k^2}{\mu^2} + \frac{\Delta V_y^2 \delta \beta}{\mu^2} \right]$$

$$\tilde{Q}_{13} = \tilde{Q}_{31} = \Delta V_x \Delta V_z \left[ \sigma_k^2 + \frac{\sigma_b^2}{\rho^2} - \sigma_\beta^2 \right]$$

$$\tilde{Q}_{22} = \Delta V_y \left[ \sigma_k^2 + \frac{\sigma_b^2}{\rho^2} \right] + \frac{\Delta V_x \Delta V_y}{\mu^2} \delta \alpha + \frac{\Delta V_y \Delta V_z}{\mu^2} \delta \beta$$

$$\tilde{Q}_{23} = \tilde{Q}_{32} = \Delta V_y \Delta V_z \left[ \sigma_k^2 + \frac{\sigma_b^2}{\rho^2} - \sigma_\beta^2 \right]$$

$$\tilde{Q}_{33} = \Delta V_z \left[ \sigma_k^2 + \frac{\sigma_b^2}{\rho^2} \right] + \mu^2 \sigma_\beta^2$$

where $\mu^2 = \Delta V_x^2 + \Delta V_y^2$, $\rho^2 = \mu^2 + \Delta V_z^2$ and $\sigma_k^2$, $\sigma_a^2$, $\sigma_b^2$, and $\sigma_\alpha^2$ are the variances associated with the resolution, proportionality, and two pointing errors, respectively.
PROGRAM QUASI

PURPOSE: PERFORM QUASI-LINEAR FILTERING EVENT IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL QUASI

SUBROUTINES SUPPORTED: SIMULL

LOCAL SYMBOLS:

COMMON COMPUTED/USED: ADEVX EDEVX NQE XF1

COMMON COMPUTED: TRIM1 XI1 XI

COMMON USED: ADEVXS EDEVXS MDIM1 TEVN W

XF XSL ZERO
QUASI Analysis

At a quasi-linear filtering event the most recent nominal trajectory is updated by using the most recent state deviation estimate. If \( \tilde{X}_j^- \) is the most recent nominal position/velocity state immediately preceding the event at time \( t_j \), and if \( \delta \tilde{X}_j^- \) is the position/velocity deviation estimate, then immediately following the quasi-linear filtering event, the most recent nominal position/velocity state is given by

\[
\tilde{X}_j^+ = \tilde{X}_j^- + \delta \tilde{X}_j^-
\]

The estimated and actual deviations from the most recent nominal trajectory must also be updated:

\[
\delta \tilde{X}_j^+ = 0
\]

\[
\delta \tilde{X}_j^+ = \delta \tilde{X}_j^- - \delta \tilde{X}_j^-
\]

A quasi-linear filtering event in no way alters the knowledge and control uncertainties at time \( t_j \). Thus knowledge covariance \( P_{k,j} \) and control covariance \( P_{c,j} \) remain constant across a quasi-linear filtering event.

Furthermore, since no velocity correction is performed, the (most recent) targeted nominal \( \tilde{X}_j^- \) is unchanged. Neither is the solve-for parameter state updated at a quasi-linear filtering event.
QUASI Flow Chart

ENTER

Increment quasi-linear filtering event counter.

Write out actual dynamic noise and estimated and actual deviations from most recent nominal at time $t_j$, immediately preceding the event.

Update and write out most recent nominal position/velocity state.

Update and write out estimated and actual state deviations at time $t_j^+$ immediately following the event.

Reset state vectors and trajectory time in preparation for next cycle. Set $N_{\text{MAX}} = 1$ since targeted nominal and most recent nominal no longer coincide.

RETURN
FUNCTION RNUN

PURPOSE: TO RETURN RANDOM NUMBERS ON A NORMAL DISTRIBUTION WITH MEAN ZERO AND STANDARD DEVIATION SIGMA.

CALLING SEQUENCE: Z=RNUM(SIGMA)

ARGUMENTS: SIGMA I STANDARD DEVIATION

SUBROUTINES SUPPORTED: SIMULL

LOCAL SYMBOLS: A SUM OF TWELVE RANDOM NUMBERS BETWEEN ZERO AND ONE
   NX CONTROLLING INTEGER
   N INTERMEDIATE INTEGER
   Q INTERMEDIATE VARIABLE
   RNUM RANDOM NUMBER FROM NORMAL DISTRIBUTION WITH MEAN ZERO AND STANDARD DEVIATION SIGMA
   RR INTERMEDIATE VARIABLE
   SS INTERMEDIATE VARIABLE
   WW INTERMEDIATE VARIABLE
   WI INTERMEDIATE VARIABLE
   YY INTERMEDIATE VARIABLE
   YI INTERMEDIATE VARIABLE
   ZZ INTERMEDIATE VARIABLE
   ZI INTERMEDIATE VARIABLE
Function subprogram RNUM supplies random numbers on a normal distribution with near zero and standard deviation $\sigma$.

Twelve random numbers $X_i$ between 0 and 1 are computed, which are then used to compute the returned random number RNUM using the following equation:

$$RNUM = \left[ \frac{\sum_{i=1}^{12} X_i - 6}{6} \right] \cdot \sigma$$
SUBROUTINE SCHED

PURPOSE: TO DETERMINE WHAT TYPE OF MEASUREMENT IS TO BE TAKEN NEXT AND AT WHAT TIME IT WILL OCCUR.

CALLING SEQUENCE: CALL SCHED(T1, T2, MMCODE)

ARGUMENTS: MMCODE 0 MEASUREMENT MODEL CODE

T1  I  PRESENT TRAJECTORY TIME

T2  O  TRAJECTORY TIME AT WHICH THE NEXT MEASUREMENT OCCURS

SUBROUTINES SUPPORTED: SIMULL  ERRANN

LOCAL SYMBOLS: M  INDEX

COMMON USED: MCTR  MCODE  NMN  TMN
SCHED Flow Chart

ENTER

MCNTR - NDN

=, -

LOOP ON M TO FIND TMN(M) GREATER OR EQUAL TO T1

T2 = TMN(M)
MPCODE = MCODE(M)

RETURN
SUBROUTINE SERIE

PURPOSE: TO COMPUTE THE TRANSCENDENTAL FUNCTIONS USED IN FLITE.

CALLING SEQUENCE: CALL SERIE(X,SX,CX)

ARGUMENTS: X I INDEPENDENT VARIABLE
            SX 0 BATTIN S-FUNCTION OF X
            CX 0 BATTIN C-FUNCTION OF X

SUBROUTINES SUPPORTED: FLITE

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS: GOSH STATEMENT FUNCTION FOR HYPERBOLIC COSINE
                 SIMH STATEMENT FUNCTION FOR HYPERBOLIC SINE
                 E    SQRT OF ABS VALUE OF X
SERIE Analysis

SERIE computes the transcendental functions $S(x)$ and $C(x)$ used in the FLIE program in the solution of Lambert's theorem.

The functions $S(x)$ and $C(x)$ are defined by

$$S(x) = \frac{\sqrt{x} - \sin \sqrt{x}}{3} \quad x > 0$$

$$= \sinh \sqrt{-x} - \frac{\sqrt{-x}}{3} \quad x < 0$$

$$= \frac{1}{6} \quad x = 0 \quad (1)$$

$$C(x) = \frac{1 - \cos \sqrt{x}}{x} \quad x > 0$$

$$= \cosh \sqrt{-x} - \frac{1}{-x} \quad x < 0$$

$$= \frac{1}{2} \quad x = 0 \quad (2)$$

For small values of $|x|$ the Taylor series expansions are used

$$S(x) = \frac{1}{3!} - \frac{x}{4!} + \frac{x^2}{5!} + ... \quad (3)$$

$$C(x) = \frac{1}{2!} - \frac{x}{3!} + \frac{x^2}{4!} + ...$$
SERIE Flow Chart

ENTER

Use Taylor expansion for S,C |x| < .01

x =? 

x > .01 Use std formula for S,C (x > 0)

x < -.01 Use std formula for S,C (x < 0)

RETURN
SUBROUTINE SETEVM

PURPOSE PERFORM ALL COMPUTATIONS COMMON TO MOST EVENTS IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE: CALL SETEVM(R1,TEVN,NCODE)

ARGUMENTS:
- NCODE I EVENT CODE
- RI I TARGETED NOMINAL SPACECRAFT STATE AT PREVIOUS MEASUREMENT OR EVENT TIME
- TEVN I EVENT TIME

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED:
- CORREL
- DYN0
- MYELS
- JACobi
- NAVM
- NTM
- PSIM
- STMPR
- TITLE
- TRAPAR

LOCAL SYMBOLS:
- BLANK DUMMY CALLING ARGUMENT
- EGVCT ARRAY OF EIGENVECTORS CORRESPONDING TO THE COLUMNS OF A GIVEN MATRIX
- EGVL ARRAY OF EIGENVALUES RELATED TO THE EIGENVECTORS CONTAINED IN EGVCT
- ICODE INTERNAL CONTROL FLAG
- OUT SQUARE ROOTS OF EIGENVALUES
- PEIG MATRIX FOR WHICH HYPERELLIPSOID IS TO BE COMPUTED
- RF NOMINAL SPACECRAFT STATE AT EVENT TIME
- VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED:
- TRTM1 XF

COMMON COMPUTED:
- DELTM XI

COMMON USED:
- CXXS CXXS CXXS
- CXV CXV CXV
- CXU CXU CXU
- FOV FOV FOV
- IEIG IEIG IEIG
- ISMC NTMC NTMC
- PS P Q
- UQ VO VO
- XLAB XLAB
SETENV Analysis

Prior to executing any event in the error analysis mode, subroutine SETENV is called to perform a series of computations which are common to all events. Subroutine SETENV computes the targeted nominal trajectory at \( t_j \), and propagates the knowledge covariance partitions at \( t_{k-1} \), the time of the previous event or measurement, forward to time \( t_j \) using the prediction equations found in the NAVM Analysis section.

For any event other than a prediction event, subroutine SETENV also computes eigenvalues, eigenvectors, and hyperellipsoids of the position and velocity partitions of the knowledge covariance at \( t_j \).
LETKVN Flow Chart

ENTER

Set $\Delta t = t_j - t_{k-1}$, Call NTM to compute the targeted nominal at $t_j$. Write out.

IF $\text{IPRT}(4) \neq 0$?

YES

Call TRAPAR to compute and write out set of navigation parameters for the targeted nominal.

Call FSIM and DYNO to compute state transition matrix partitions and dynamic noise covariance matrix over $[t_{k-1}, t_j]$. Write out.

Call NAVM to compute knowledge covariance partitions at $t_j$. Write out.

Is event a prediction event?

YES

RETURN

NO

Compute and write out eigenvalues, eigenvectors, and hyperellipsoids of position and velocity partitions of the knowledge covariance at $t_j$.

Is event a guidance event?

YES

RETURN

NO

Reset time and state vector in preparation for next cycle.

RETURN
SUBROUTINE SETEVS

PURPOSE PERFORM ALL COMPUTATIONS COMMON TO MOST EVENTS IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL SETEVS(R,TEVN,RI1,NCODE)

ARGUMENTS

NCODE I EVENT CODE

RI I TARGETED NOMINAL SPACECRAFT STATE AT PREVIOUS MEASUREMENT OR EVENT TIME

RII I MOST RECENT NOMINAL SPACECRAFT STATE AT PREVIOUS MEASUREMENT OR EVENT TIME

TEVN EVENT TIME

SUBROUTINES SUPPORTED: SIMILL

SUBROUTINES REQUIRED: CORREL DYNOS HYELS JACOBI NAVM NTMS PSIM SMMPL TITLES TRAPAR

LOCAL SYMBOLS

DUM INTERMEDIATE VECTOR

EGVT ARRAY OF EIGENVECTORS

EGVL ARRAY OF EIGENVALUES

ICODE INTERNAL CONTROL FLAG

OUT SQUARE ROOTS OF EIGENVALUES

PEIG MATRIX WHOSE HYPERELLIPSOID IS TO BE COMPUTED

RF1 MOST RECENT NOMINAL SPACECRAFT STATE AT EVENT TIME

RF TARGETED NOMINAL SPACECRAFT STATE AT EVENT TIME

VEIG MATRIX TO BE DIAGONALIZED

COMMON COMPUTED/USED: ADEVX EDEVX TRTM1 XF1 XF XI1 ZF ZI

COMMON COMPUTED: DELTM XI

COMMON USED: ADEVXS CXSU CXSV CXU CXV CXXS EDEVXS FOP FOV IEIG IHYP1 IPRT ISTMC NDIM1 NGE
SETEVXS Analysis

Prior to executing any event in the simulation mode, subroutine SETEVXS is called to perform a series of computations which are common to all events. After computing the targeted nominal and most recent nominal states at the time of the event $t_j$, knowledge covariance partitions are propagated forward to time $t_j$ from time $t_{k-1}$ of the previous event or measurement using the prediction equations found in the NAVN Analysis section. The actual trajectory state at $t_j$ is computed using

$$X_j = Z_j + \omega_j$$

where $Z_j$ is the actual trajectory state assuming no unmodeled acceleration has been acting on the spacecraft, and $\omega_j$ is the contribution of the actual unmodeled acceleration to the actual trajectory state at $t_j$. The actual and predicted position/velocity deviations from the most recent nominal at $t_j$ are given by

$$\delta \tilde{X}_j = X_j - \tilde{X}_j$$

and

$$\delta \tilde{X}_j = \Phi(t_j, t_{k-1}) \delta \tilde{X}_{k-1} + \Theta_{xx} (t_j, t_{k-1}) \delta X_{s_j},$$

respectively, where $\Phi$ and $\Theta_{xx}$ are the state transition matrix partitions over $[t_{k-1}, t_j]$.

For any event other than prediction and quasi-linear filtering events, subroutines SETEVXS also computes eigenvalues, eigenvectors, and hyper-ellipsoids of the position and velocity partitions of the knowledge covariance at $t_j$. 
SETEV5 Flow Chart

 ENTER

 Set $\Delta t = t_j - t_{k-1}$.
 Call NTMS to compute the targeted nominal at $t_j$.

 Is targeted nominal identical to most recent nominal?

 NO

 Set most recent nominal equal to targeted nominal at $t_j$.

 YES

 Call PSM and DYNMS to compute state transition matrix partitions and dynamic noise covariance matrix over the time interval $[t_{k-1}, t_j]$.

 Call NAVM to compute knowledge covariance partitions at $t_j$.

 Compute actual position/velocity state at $t_{k-1}$. Compute actual position/velocity state at $t_j$ before effect of unmodeled acceleration has been added.

 A
Call DYNOS to compute effect of unmodeled acceleration. Use to update actual position/velocity state at $t_j$.

Write out targeted nominal, most recent nominal, and actual trajectories at $t_j$.

IF IFRT(4) ≠ 0?

YES

Call TRAPAR to compute and write out set of navigation parameters for the actual trajectory.

Write out state transition matrix partitions and the diagonal of the dynamic noise covariance matrix over $[t_{k-1}, t_k]$. Write out knowledge correlation partitions and standard deviations at $t_j$.

Compute actual and predicted position/velocity deviations from the most recent nominal at $t_j$.

Is event a prediction or quasi-linear filtering event?

YES

RETURN

NO
B

Compute and write out eigenvalues, eigenvectors, and hyperellipsoids of position and velocity partitions of the knowledge covariance at $t_j$.

Is event a guidance or C$\Delta$A$\cap$ event?

- YES → RETURN
- NO →

Write out actual dynamic noise vector. Write out estimated and actual position/velocity and solve-for parameter deviations from the most recent nominal at $t_j$.

Reset time and state vectors in preparation for next cycle.

RETURN
PROGRAM SIMUL

PURPOSE: TO CONTROL THE COMPUTATIONAL FLOW THROUGH THE BASIC CYCLE (MEASUREMENT PROCESSING) AND ALL EVENTS IN THE SIMULATION PROGRAM

SUBROUTINES SUPPORTED: MAIN

SUBROUTINES REQUIRED: BIAS DYNOS MENOS NAVM NTMS TRAKS PRINT4 PSIM SCHED SETEVS

LOCAL SYMBOLS

BVAL ACTUAL MEASUREMENT BIAS VECTOR
DUMM INTERMEDIATE VARIABLE
DUM INTERMEDIATE VECTOR
IPRN MEASUREMENT PRINT TIME COUNTER
MMCODE MEASUREMENT CODE
NEVENT EVENT COUNTER
RNUGM RANDOM MEASUREMENT NOISE
TRTM2 TIME OF THE MEASUREMENT

COMMON COMPUTED/USED:

ADEVX ANGIS AY EDEVXS EDEVX
Ey ICODE MCNTR NAFG RES
RF1 RI1 RI TEVN TRTM1
XF1 XF XI1 XI ZF
ZI

COMMON COMPUTED:

AYMEY DELTM EDEVSM EDEVXM

COMMON USED:

AK AM AR FNTM H
IEVNT IPRINT ISTMC NAE NAF6
NDIM1 NEV NGE NMM NQE
NR NTMC PHI RF S
TEV TXXS W ZERO
SIMUL Analysis

The primary function of subroutine SIMUL is to control the computational flow through the basic cycle (measurement processing) and all events in the simulation mode. Subroutine SIMUL also performs some computations in the basic cycle. All event-related analysis is presented in the event subroutines themselves and will not be treated below.

In the basic cycle the first task of SIMUL is to control the generation of targeted nominal and most recent nominal spacecraft states, $\tilde{x}_{k+1}$ and $\tilde{x}_{k+1}$, respectively, at time $t_k$, given states $\tilde{x}_k$ and $\tilde{x}_k$ at time $t_k$. Then, calling PSIM, DYNNS, TRAKS, and MENG88, successively, SIMUL controls the computations of all matrix information required by subroutine NAVM in order to compute the covariance matrix partitions at time $t_{k+1}^+$ immediately following the measurement.

After computing the actual state $x_k$ at time $t_k$ from

$$x_k = \tilde{x}_k + \delta \tilde{x}_k$$

where $\delta \tilde{x}_k$ is the actual spacecraft state deviation from the most recent nominal, SIMUL controls the generation of the actual state $z_{k+1}$ at time $t_k$ before the effect of unmodeled acceleration has been added. Then, having called DYNNS to compute the effect of unmodeled acceleration $\omega_{k+1}$, SIMUL computes the actual state and actual state deviation at time $t_{k+1}$:

$$x_{k+1} = z_{k+1} + \omega_{k+1}$$

$$\delta \tilde{z}_{k+1} = x_{k+1} - \tilde{x}_{k+1}$$

With both the most recent nominal and actual spacecraft states available at $t_{k+1}$, SIMUL calls TRAKS twice in succession to compute the ideal measurements $\tilde{y}_{k+1}$ and $\tilde{y}_{k+1}$, respectively, which would be made at each of these trajectory states. Calling MENG88, RNUM, and BIAS to compute the actual measurement noise and bias corrupting the ideal measurement associated with the actual state, SIMUL computes the actual measurement at time $t_{k+1}$ using

$$y_{k+1} = \tilde{y}_{k+1} + b_{k+1} + \nu_{k+1}$$
where \( b_{k+1} \) and \( \nu_{k+1} \) represent the actual measurement bias and noise, respectively.

All information required for computing both predicted and filtered state deviations from the most recent nominal at \( t_{k+1} \) is now available. With \( \bar{\Phi} \) and \( \Theta_{xxs} \) denoting state transition matrix partitions over the time interval \([t_k, t_{k+1}]\), SIMUL computes the predicted spacecraft state deviations and solve-for parameter deviations at \( t_{k+1} \) using

\[
\delta\hat{x}_{k+1}^\pm = \bar{\Phi}\delta\hat{x}_k^\pm + \Theta_{xxs}\delta\hat{x}_{s_k}^\pm
\]

Prior to computing filtered deviations, SIMUL computes the measurement residual from

\[
\epsilon_{k+1} = (Y_{a_{k+1}} - \hat{Y}_{k+1}) - H_{k+1}\delta\hat{x}_k^\pm - M_{k+1}\delta\hat{x}_{s_{k+1}}^\pm
\]

where \( H_{k+1} \) and \( M_{k+1} \) are observation matrix partitions. Filtered spacecraft state deviations and solve-for parameter deviations are then computed from

\[
\delta\check{x}_{k+1}^\pm = \delta\hat{x}_{k+1}^\pm + K_{k+1}\epsilon_{k+1}
\]

\[
\delta\check{x}_{s_{k+1}}^\pm = \delta\hat{x}_{s_{k+1}}^\pm + S_{k+1}\epsilon_{k+1}
\]

where \( K_{k+1} \) and \( S_{k+1} \) are the filter gain constants.
Simul Flow Chart

1. ENTER

   Initialize event counter NEVENT and print counter IPRN.

2. Define states \( \tilde{x}_k \) and \( \tilde{x}_k \) at time \( t_k \).

3. Call SCHED to obtain the time \( t_{k+1} \) of the measurement and the measurement code.

4. Define time interval \( \Delta t = t_{k+1} - t_k \).

5. Does an event occur before \( t_{k+1} \)?
   - YES: Goto 360
   - NO: Goto 330

6. Call NTMS to compute state \( \tilde{x}_{k+1} \).

7. Does \( N_{MTRJ} = 1 \)?
   - YES: Goto 350
   - NO: Set \( \tilde{x}_{k+1} = \tilde{x}_{k+1} \).

330

350
Increment measurement counter MCNTR.

Call PSDM to compute state transition matrix partitions over \([t_k, t_{k+1}]\) for the most recent nominal trajectory.

Call DYN5S to compute \(Q_{k+1,k}\).

Call TRAKS to compute the observation matrix partitions at \(t_{k+1}\).

Call MEMS to compute \(P_{k+1}\).

Call NAVM to compute covariance matrix partitions at \(t^+_k\).

Compute actual state \(X_k\) at \(t_k\).

Call NTMS to compute actual state \(X_{k+1}\) before effect of unmodeled acceleration has been added.

Call DYN5S to compute effect of unmodeled acceleration \(\omega_{k+1}\).
Compute actual state $x_{k+1}$ after effect of unmodeled acceleration has been added.

Compute actual state deviation $\delta x_{k+1}$.

Call TRAKS to compute ideal measurement $\hat{y}_{k+1}$ from most recent nominal.

Call TRAKS to compute ideal measurement from actual trajectory.

Call MENVS to compute $R_{k+1}$.

Compute actual measurement noise $\nu_{k+1}$.

Call BIAS to compute $b_{k+1}$.

Compute actual measurement $y_{k+1}^a$.

Compute quantities required for adaptive filtering.
Complete predicted state deviations $\delta \tilde{x}_{k+1}^-$ and $\delta \tilde{y}_{k+1}^-$ at $t_{k+1}^-$.  

Compute measurement residual $\epsilon_{k+1}$.  

Compute filtered state deviations $\delta \tilde{x}_{k+1}^+$ and $\delta \tilde{y}_{k+1}^+$ at $t_{k+1}^+$.  

Increment print counter IPRN.  

Is it time to print?  

NO  

YES  

Call PRINT4 to write out all basic cycle data.  

Reset time and targeted and most recent nominal states in preparation for next cycle.  

352  

400
Define event code ICØDE and event time \( t_1 \).

ICØDE \( \leq 5 \) ?

NO

YES

Call SETEV$S$ to compute information common to most types of events.

ICØDE = ?

1 2 3 4 5 6

Call prediction event overlay.
Call CØNGØM event overlay.
Call adaptive filtering overlay.

Call GUIŠIM.
Call quasi-linear filtering event overlay.

NAFC \( \neq \text{NAF}_6(\text{NAE}+1) \) ?

YES

NO

NAFC = 0

Increment even. counter.

400
Have all measurements been processed?

Have all events been performed?

Define states $\overline{x}_{k+1}$ and $\tilde{x}_{k+1}$ at time $t_{k+1}$.

Define time interval $\Delta t = t_f - t_{k+1}$ and states $\overline{x}_{k+1}$ and $\tilde{x}_{k+1}$.

Call NTMS to compute state $\overline{x}_f$.

Call NTMS to compute state $\overline{x}_f$.

Set $\tilde{x}_f = \overline{x}_f$.

NO

YES

NO

YES

RETURN

RETURN

320

390

420

400
Call PSM to compute state transition matrix partitions over \([t_{k+1}, t_f]\) for the most recent nominal trajectory.

Call DYN0S to compute \(Q_{f,k+1}\).

Call NAVM to compute covariance matrix partitions at \(t_f\).

Compute actual state \(X_{k+1} \) at \(t_{k+1}\).

Call NIMS to compute actual state \(X_t \) at \(t_f \) before effect of unmodeled acceleration has been added.

Call DYN0S to compute effect of unmodeled acceleration \(\omega_f\).

Compute actual state \(X_f \) after effect of unmodeled acceleration has been added.

Compute actual state deviations \(\delta^X_f\).

Compute predicted state deviations \(\delta^X_f \) and \(\bar{\delta}^X_f \) at \(t_f\).

RETURN
SUBROUTINE SPACE

PURPOSE: COUNTS THE NUMBER OF LINES BEING PRINTED TO DETERMINE WHEN TO SKIP TO THE NEXT PAGE WITH A NEW HEADING

CALLING SEQUENCE CALL SPACE(LINES)

ARGUMENT LINES I NUMBER OF LINES THAT WILL BE WRITTEN IN THE NEXT OUTPUT STATEMENT

SUBROUTINES SUPPORTED: INPUTZ PRINT VECTOR VMP

SUBROUTINES REQUIRED: NEWPGE

COMMON COMPUTED/USED: LINCT

COMMON USED: LINPGE
SUBROUTINE STAPRL

PURPOSE: TO COMPUTE THE PARTIAL DERIVATIVES OF STATION LOCATION ERRORS.

CALLING SEQUENCE: CALL STAPRL(AL,ALON,ALAT,PAT2,VEC,PA)

ARGUMENTS:
AL I ALTITUDE OF THE STATION
ALAT I LATITUDE OF THE STATION
ALON I LONGITUDE OF THE STATION
PA 0 PARTIAL OF STATION POSITION AND VELOCITY WITH RESPECT TO ALTITUDE, LATITUDE AND LONGITUDE
PAT2 I LONGITUDE + OMEGA*(CURRENT TIME-LAUNCH TIME)
VEC UNUSED

SUBROUTINES SUPPORTED: TRAKS  TRAKM

LOCAL SYMBOLS:
G1 SINE OF LATITUDE
G2 COSINE OF LATITUDE
G3 SINE(Phi + Omega(T-Univt))
G4 COSINE(Phi + Omega(T-Univt))
WHERE Phi =LONGITUDE
      Omega=EARTH ROTATION RATE
      T =TIME
      Univt=UNIVERSAL TIME
G5 SINE OF OBLIQUITY OF EARTH
G6 COSINE OF OBLIQUITY OF EARTH
OMEG OMEGA IN PROPER UNITS

COMMON USED: S OMEGA TM
STAPRL Analysis

The ecliptic components of the position and velocity of a tracking station relative to the Earth are related to station location parameters \( R, \theta, \) and \( \phi \) through the following set of equations:

\[
\begin{align*}
X_s &= R \cos \theta \cos G \\
Y_s &= -R \cos \theta \cos \phi \sin G + R \sin \theta \sin \epsilon \\
Z_s &= -R \cos \theta \sin \epsilon \sin G - R \sin \theta \cos \epsilon \\
\dot{X}_s &= -\omega R \cos \theta \sin G \\
\dot{Y}_s &= \omega R \cos \theta \cos \epsilon \cos G \\
\dot{Z}_s &= -\omega R \cos \theta \sin \epsilon \cos G \\
\end{align*}
\]

where \( G = \phi + \omega (t - T) \), and \( T \) is the universal time at some epoch (usually launch time).

Subroutine STAPRL computes the negative of the partials of the previous quantities with respect to the station location parameters \( R, \theta, \) and \( \phi \). These partials are summarized below:

\[
\begin{align*}
- \frac{\partial X_s}{\partial R} &= - \cos \theta \cos G \\
- \frac{\partial X_s}{\partial \theta} &= - R \sin \theta \cos G \\
- \frac{\partial X_s}{\partial \phi} &= - R \cos \theta \sin G \\
- \frac{\partial Y_s}{\partial R} &= - \left[ \sin \epsilon \sin \theta + \cos \epsilon \cos \theta \sin G \right] \\
- \frac{\partial Y_s}{\partial \theta} &= - R \cos \epsilon \sin \theta \sin G - R \sin \epsilon \cos \theta \\
- \frac{\partial Y_s}{\partial \phi} &= - R \cos \epsilon \cos \theta \cos G \\
- \frac{\partial Z_s}{\partial R} &= - \sin \epsilon \cos \theta \sin G - \cos \epsilon \sin \theta \\
\end{align*}
\]
\[-\frac{\partial z}{\partial \theta} = - [\sin \epsilon \sin \theta \sin G + \cos \epsilon \cos \theta]\]
\[-\frac{\partial z}{\partial \phi} = \sin \epsilon \cos \theta \cos G\]
\[-\frac{\partial z}{\partial \varphi} = \omega \cos \theta \sin G\]
\[-\frac{\partial z}{\partial \varphi} = -\omega R \sin \theta \sin G\]
\[-\frac{\partial z}{\partial \theta} = \omega R \cos \theta \cos G\]
\[-\frac{\partial z}{\partial \phi} = \omega R \cos \epsilon \sin \theta \cos G\]
\[-\frac{\partial z}{\partial \varphi} = \omega R \cos \epsilon \cos \theta \sin G\]
\[-\frac{\partial z}{\partial \varphi} = \omega \sin \epsilon \cos \theta \cos \varphi\]
\[-\frac{\partial z}{\partial \theta} = -\omega R \sin \theta \cos G\]
\[-\frac{\partial z}{\partial \phi} = -\omega \sin \epsilon \cos \theta \sin G\]
SUBROUTINE STMPC

PURPOSE: TO PRINT OUT THE TRANSPOSES OF THE STATE TRANSITION
MATRIX PARTITIONS PHI, TXS, AND TXU OVER AN ARBITRARY
INTERVAL OF TIME.

CALLING SEQUENCE: CALL STMPC(TRM1,TRM2)

ARGUMENTS:

TRM1 I TIME AT BEGINNING OF INTERVAL OVER WHICH
STATE TRANSITION MATRIX PARTITIONS HAVE
BEEN COMPUTED

TRM2 I TIME AT END OF INTERVAL OVER WHICH STATE
TRANSITION MATRIX PARTITIONS HAVE BEEN
COMPUTED

SUBROUTINES SUPPORTED: PRINT4 SETEVS GUSSIM GUISS PRESIM
PRINT3 SETEVM GUIDM GUID PRED

COMMON USED:

NDIM1 NDIM2 PHI TXU TXS
XLPB XSL XU
SUBROUTINE SUB1

PURPOSE: TO COMPUTE POSITION AND VELOCITY MAGNITUDES.

CALLING SEQUENCE: CALL SUB1(X,XE,XP)

ARGUMENTS:
X  I INERTIAL POSITION/VELOCITY OF THE VEHICLE
XE I EARTH-S POSITION/VELOCITY
XP I POSITION/VELOCITY OF THE TARGET PLANET

SUBROUTINES SUPPORTED: PRINT4

LOCAL SYMBOLS:
RX MAGNITUDE OF INERTIAL POSITION VECTOR
RY MAGNITUDE OF GEOCENTRIC POSITION VECTOR
RZ MAGNITUDE OF PLANETOCENTRIC POSITION VECTOR
VX MAGNITUDE OF INERTIAL VELOCITY VECTOR
VY MAGNITUDE OF GEOCENTRIC VELOCITY VECTOR
VZ MAGNITUDE OF PLANETOCENTRIC VELOCITY VECTOR
Y GEOCENTRIC POSITION/VELOCITY OF THE VEHICLE
Z PLANETOCENTRIC POSITION/VELOCITY OF THE VEHICLE
SUBROUTINE TARGET

PURPOSE: TO PERFORM EXECUTIVE FUNCTIONS OF THE TARGETING MODE AS CALLING REQUIRED SUBROUTINES TO READ THE INPUT DATA, COMPUTING THE ZERO ITERATE IF NECESSARY AND PERFORMING THE ACTUAL TARGETING THROUGH THE PROGRESSIVE STAGES USED BY STEAP.

CALLING SEQUENCE: CALL TARGET

SUBROUTINES SUPPORTED: GIODANS

SUBROUTINES REQUIRED: TAROPT TAMAX DESENT PECEQ VNP

LOCAL SYMBOLS: ABV INTERMEDIATE VARIABLE USED TO LIMIT EACH DELTAV COMPONENT CHANGE

ACC VECTOR OF ACCURACY LEVELS FOR THE CURRENT TARGETING EVENT

ACK ACTUAL ACCURACY USED BY SUBROUTINE VNP

AER ABSOLUTE VALUES OF DIFFERENCES BETWEEN DESIRED AND NOMINAL END CONDITIONS

CERROR CURRENT SUM OF WEIGHTED DIFFERENCES OF DESIRED AUXILIARY AND NOMINAL AUXILIARY END CONDITIONS

DEV DIFFERENCES (ERRORS) BETWEEN AUXILIARY END CONDITIONS (DESIRER AND NOMINAL)

ISP2 INDICATOR USED BY SUBROUTINE VNP

=1 STOP AT SPHERE-OF-INFLUENCE

=0 DO NOT STOP AT SPHERE-OF-INFLUENCE

ITBAD BAD STEP COUNTER

ITER ITERATION COUNTER

ITOL CONVERGENCE INDICATOR

=1 CASE CONVERGED

=0 CASE DID NOT CONVERGE

IT INDICATOR USED TO LOCATE DESIRED TIME VALUE FOR OUTER TARGETING

I INDEX

J INDEX

LOWMI INDICATOR USED TO CALCULATE THE PHASE 2
TARGETING MATRIX

NOMORE  INDICATOR USED TO LIMIT OUTER TARGETING
=0 OUTER TARGETING HAS NOT BEEN PERFORMED
=1 OUTER TARGETING HAS ALREADY BEEN PERFORMED

OSPH  ORIGINAL SPHERE OF INFLUENCE OF THE TARGET PLANET

PERROR  PREVIOUS VALUE OF CERROR

REDUC  INTERMEDIATE VARIABLE USED IN BAD STEP REDUCTION

RIS  LOCAL VECTOR USED TO SAVE AND RESTORE THE RIN VECTOR

RR  INTERMEDIATE VARIABLE FOR OUTER TARGETING

RSF  FINAL SPACECRAFT STATE RETURNED BY VMP

SSOI  INTERMEDIATE VARIABLE FOR OUTER TARGETING

STOL  INTERMEDIATE VARIABLE FOR OUTER TARGETING

TMDF  INTERMEDIATE VARIABLE FOR OUTER TARGETING

TVH  PHASE 1 TARGETED VELOCITY AT HIGHEST ACCURACY

TVL  PHASE 1 TARGETED VELOCITY AT LOWEST ACCURACY

VV  INTERMEDIATE VARIABLE FOR OUTER TARGETING

XTIME  CURRENT DT TIME USED TO CALCULATE EQECP FOR TARGET PLANET

COMMON COMPUTED/USED:
CTOL  OAUX  DELTAV  OTAR  IBAD
IBAST  IPHASE  ISPH  ISTART  ISTOP
ITAR  KEYSAR  LEVELS  LEV  MEAX
MAXBAD  NITS  NOPAR  NOPHAS  MOSO1
PHI  RIN  SPHERE

COMMON COMPUTED:
DELTP  DELV  ICL2  ICL  INCMT
INPR  IPRINT  KXAR  KMT  RRF
COMMON USED

<table>
<thead>
<tr>
<th>AAUX</th>
<th>AC</th>
<th>ALNGTH</th>
<th>ATAR</th>
<th>OC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTAT</td>
<td>DT</td>
<td>DVMAX</td>
<td>D1</td>
<td>EQCP</td>
</tr>
<tr>
<td>FAC</td>
<td>IBAOS</td>
<td>KTAR</td>
<td>KUR</td>
<td>LVLS</td>
</tr>
<tr>
<td>MAT</td>
<td>MAXB</td>
<td>NOIT</td>
<td>NPAR</td>
<td>NTP</td>
</tr>
<tr>
<td>ONE</td>
<td>RC</td>
<td>SPHFAC</td>
<td>TAR</td>
<td>TM</td>
</tr>
<tr>
<td>TOL</td>
<td>TRTM</td>
<td>TWO</td>
<td>ZERO</td>
<td>TM</td>
</tr>
</tbody>
</table>
TARGET Analysis

TARGET is responsible for the control of any targeting (nonlinear guidance) event. The targeting is done either by the Newton-Raphson technique or by a steepest descent-conjugate gradient algorithm, the method being specified by the user. In either case numerical differencing is used to compute the required sensitivities.

I. Preliminaries

The current inertial state of the spacecraft upon entering TARGET is first saved (RIS-RIN) along with the original SOI radius (OSPH-Sphere) since both variables may be changed during the course of the targeting. Before exiting from TARGET these values are restored.

The index of the current event KUR has been computed by TRJTRY. This enables the specific targeting parameters for the current event to be set:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHOD</td>
<td>Triggers Newton-Raphson (=0) or Steepest Descent (=1) technique</td>
</tr>
<tr>
<td>MATX</td>
<td>Determines whether Newton-Raphson matrix is computed</td>
</tr>
<tr>
<td></td>
<td>always (=2) or only at low level (=1)</td>
</tr>
<tr>
<td>IBAST</td>
<td>Determines whether bad step checks are made never (=1), high level only (=2) or always (=3)</td>
</tr>
<tr>
<td>LEVELS</td>
<td>Number of integration accuracy levels to be used</td>
</tr>
<tr>
<td>NOPAR</td>
<td>Number of target parameters to be used</td>
</tr>
<tr>
<td>ACC</td>
<td>Actual accuracy levels used in targeting</td>
</tr>
</tbody>
</table>

The following flags are then initialized to zero:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITDS</td>
<td>Counter for steepest descent iterations</td>
</tr>
<tr>
<td>LOWHI</td>
<td>Flag indicating whether first phase complete (=1) or not (=0)</td>
</tr>
<tr>
<td>NUMORE</td>
<td>Flag indicating whether outer targeting has been done (=1) or not (=0)</td>
</tr>
</tbody>
</table>

The target time is computed and using that time the transformation matrix \( \Phi_{EC EQ} \) from ecliptic to target planet equatorial coordinates is calculated (PECER).

II. Phase Preparations

TARGET performs the targeting in one phase unless targeting to TCA (time of closest approach). In that case the trajectory is targeted in two phases: the first phase targets to the target planet SOI (sphere of influence), the second phase to the closest approach conditions. IPHASE is the phase counter, NOPHAS is the number of phases needed.
If all the phases have been completed, the program prepares to exit. If the last iterate satisfied the target tolerances ITOL will have been set to 1. If it did not, ITOL will be zero and this requires that IKIT be set to 1 to terminate the program upon return to the basic cycle.

If the last phase has not yet been completed TAROPT is now called with an argument 1 to compute the following phase parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYTAR(3)</td>
<td>Vector of codes of target parameters</td>
</tr>
<tr>
<td>KAXTAR(3)</td>
<td>Vector of codes of auxiliary parameters</td>
</tr>
<tr>
<td>DTAR(3)</td>
<td>Vector of desired values of target parameters</td>
</tr>
<tr>
<td>DAUX(3)</td>
<td>Vector of desired values of auxiliary parameters</td>
</tr>
<tr>
<td>FAC(3)</td>
<td>Weighting factors for loss function for auxiliary parameters</td>
</tr>
<tr>
<td>ISTOP</td>
<td>Flag indicating integration stopping conditions with ISTOP = 1,2,3 indicating fixed final time, SOI, or CA encounter</td>
</tr>
</tbody>
</table>

The target parameters are the parameters actually desired; the auxiliary parameters are the parameters used to do the targeting. The target and auxiliary parameters are identical except when i and r are targets. In that case the corresponding auxiliary parameters are B·T and B·R which are much more linear variables. The codes of the target and auxiliary parameters are as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRP</td>
</tr>
<tr>
<td>2</td>
<td>TSI</td>
</tr>
<tr>
<td>3</td>
<td>TCS</td>
</tr>
<tr>
<td>4</td>
<td>TCA</td>
</tr>
<tr>
<td>5</td>
<td>BDT</td>
</tr>
<tr>
<td>6</td>
<td>BDR</td>
</tr>
<tr>
<td>7</td>
<td>RCA</td>
</tr>
<tr>
<td>8</td>
<td>INC</td>
</tr>
<tr>
<td>9</td>
<td>SMA</td>
</tr>
<tr>
<td>10</td>
<td>XRF</td>
</tr>
<tr>
<td>11</td>
<td>YRF</td>
</tr>
<tr>
<td>12</td>
<td>ZRF</td>
</tr>
</tbody>
</table>

* not currently available

III. Level Preparations

Within any phase TARGET operates through a series of integration accuracy levels prescribed by the user. After completing each level TARGET checks to see if the maximum number of levels LEVELS has been exceeded. If it has the program cycles to the beginning of the "phase loop" to go to the next phase. If the current level LEV is less than LEVELS the following computations are made.

The flag ITARM controls whether the previous targeting matrix is to be used (=1) or whether the matrix is to be recomputed (=2) during the current level. ITARM is set according to the current values of MATH, ISTART, and LEV.

The flag IBAD controls the bad step logic. If IBAD=1 no bad step check will be made during the current level; if IBAD=2 the bad step check will be in effect. TARGET sets IBAD according to the values of IBAST and LEV.

The flags ITOL, ITER, ITBAD are set to 0 to begin the iterations. The allowable iterations NITS and bad iterations MAXBAD are also set at this time.
IV. Iterate Calculations

Within each level the program makes one or more iterations. After each iteration the program updates the iteration counter ITER. If the maximum number of iterations for this level NITS has been exceeded, the program sets NIT to 1 and prepares for the return from TARGET. Otherwise TARGET computes the target and auxiliary values corresponding to the current iterate values of state (position and velocity) RIN.

The integration parameters are first set. VMP is then called to propagate the initial state to the final stopping conditions. Checks are made to insure that the target planet SO1 was intersected if the stopping conditions were SO1 or CA. If it was not intersected and this is the first iteration, the "outer targeting" phase is entered (see below). If "outer targeting" has already been performed, the bad-step check is entered to reduce the previous correction by REDUC.

Otherwise TAROPT is called with the argument 2 to compute the desired and actual target (UTAR, ATAR) and auxiliary (DAUX, AAUX) parameter values. The absolute error in target values AER and the error in auxiliary values DEV are then computed.

If the current iterate is the first integration at the low level during the second phase of targeting (LOWH=1) TARMAX is now called to compute the phase 2 targeting matrix. Then the state RIN is reset to the targeted velocity at the high level TVH to prepare for the second phase targeting. The program then returns to the level loop.

Otherwise the program now checks the actual target variables to determine whether they satisfy the input tolerances or not.

V. Tolerances Satisfied

If the tolerances are satisfied, the program first checks to see if the current targeting phase is outer targeting. If it is TARGET restores the original target parameters and initiates the normal targeting (see Outer Targeting below).

If the current targeting is already normal targeting, TARGET sets ITOL=1 to indicate the satisfaction of the tolerances. If the problem is a 2-phase and the current level is the highest level in phase 1 targeting, the targeted high level velocity TVH=RIN is saved, LOWH is set to 1 and the targeted low level velocity is recalled RIN=TVL for the construct of the phase 2 targeting matrix. Then the level loop is reentered.

VI. Bad Step Reduction

If the target parameter values of any iterate are not within the acceptable tolerances TARGET now assigns a scalar error $\epsilon$ to the iterate using the weighting factors $\bar{w}$

$$\epsilon = \bar{w} \cdot \Delta \bar{F}$$
If the bad-step check is to be made on this iterate the current error $\epsilon$ is compared to the previous error $\epsilon_p$. If $\epsilon > \epsilon_p$ and the maximum number of bad steps has not been exceeded, the previous correction $\Delta \mathbf{v}$ is reduced by REDUC (usually 1/4). The initial state BIN is adjusted by this and the iterate loop is reentered. If the maximum number of bad steps has been made, KMIT is set to 1 and the preparations for return are made.

VII. Generation of Next Iterate

The correction $\Delta \mathbf{v}$ to any iterate may be computed from either of two techniques selected by the flag METHOD. If METHOD $\neq 0$, subroutine DESCENT is called for the computation of the $\Delta \mathbf{v}$ by a steepest descent algorithm. The numerical value of METHOD determines the number $n$ of conjugate gradient steps between each straight gradient step where $n = $ METHOD-1. Thus if METHOD=1, every step is in the direction of the gradient. But if METHOD=5, four steps are taken following the conjugate gradient direction before rectification by the gradient direction.

If METHOD=0, the Newton-Raphson correction is used. If ITARM=0, TARMAX is called for the computation of the targeting matrix $\phi$ by numerical differencing. If any of the integrations made in constructing that matrix satisfy the tolerances in $\tau$, the flag IEND is set to 1 before returning to TARGET. Thus a check must be made on IEND. If ITARM=1 the previous targeting matrix is used. The correction is then given by

$$\Delta \mathbf{v} = \phi \cdot \Delta \mathbf{a}$$

where $\Delta \mathbf{a}$ are the deviations in the iterate auxiliary values. The $\Delta \mathbf{v}$ is checked to ensure that the maximum step size DVMAX is not violated; if it is, the $\Delta \mathbf{v}$ is reduced proportionately to satisfy it. The next iterate is then set to

$$(\mathbf{r}, \mathbf{v}) = (\mathbf{r}, \mathbf{v} + \Delta \mathbf{v})$$

and the return is made to the iterate loop.

VIII. Outer Targeting

Occasionally the zero iterate initial state leads to a trajectory missing the target body SOI. Since all target options except one (targeting to a specified position, i.e., KFAIL = 10, 11, 12) require the trajectory to intersect the target body SOI steps must be taken to correct this.

Let the initial state propagated forward lead to a trajectory with a closest approach to the target body of $r_{CA}$ with $r_{CA} > r_{SI}$ where $r_{SI}$ is the radius of the SOI.
Until the initial trajectory intersects the SOI the usual targeting cannot be done. Therefore an "artificial" SOI is introduced having a radius of

\[ r_{ASI} = 1.2 \times r_{CA} \]

The initial trajectory obviously intersects the artificial SOI and hence may be targeted to conditions on the ASO1. If the target conditions are established as \( B \cdot T_A = B \cdot R_A = 0 \), when this artificial targeting is completed, the refined trajectory will be headed straight for the target body when it hits the ASOI. Thus the refined trajectory should automatically hit the normal SOI when propagated past the ASOI. To ensure that the time of intersection with the normal SOI is consistent with the target time, an artificial target time is also used. Let the speed of the spacecraft with respect to the target body at \( r_{CA} \) be \( v_{CA} \). Make the approximation that this speed will be roughly the same for the refined trajectory. Then the time that the spacecraft should intersect the ASOI is

\[ t_{ASI} = t_{CA} - \frac{r_{ASI}}{v_{CA}} \]

or

\[ t_{ASI} = t_{SI} - \frac{r_{ASI} - r_{SI}}{v_{CA}} \]

where the first formula should be used if the target time is \( t_{CA} \) or \( t_{CS} \) and the second formula is used for \( t_{SI} \).

Thus when a trajectory is found which misses the normal SOI, the closest approach state \( r_{CA}, v_{CA} \) is recorded. The normal SOI radius is stored and the artificial SOI radius given above is used in its place. Target parameters of \( B \cdot T_A, B \cdot R_A \), and \( t_{ASI} \) are then set up as the targets. When targeting of this artificial problem is complete, the resulting trajectory will intersect the normal SOI and the original problem may be solved.
TARGET Flow Chart

PRELIMINARIES

Save original SPHERE, state RIN
Set parameters for current event:
    METHOD, MATX, IBAST, LEVELS, ACC, NOPAR
Initialize flags: ITDS, LOWHI, NOMORE, PHASE, NOPHAS
Compute $\phi_{ECEQ}$ for target time

PHASE PREPARATIONS

A

$\text{IPHASE} = \text{IPHASE} + 1$
$\text{LEV} = 0$

$\text{IPHASE} : \text{NOPHASE}$

B

$\text{LEV} = \text{LEV} + 1$

LEVEL PREPARATIONS

Set ITARM flag

C

Call TAROPT(1) to compute KEYTAR
    KAXTAR, DTAZ, DAUX, FAC, ISTOP, NOPHAS

TARGET-6

ISTART = 0
Set ZBAD flag.

Set other constants for current level: ITUL = ITER = ITBAD = 0, PERROR, NITS, MAXBAD

ITERATE CALCULATIONS

\( I_T E_R - I_T E_R + 1 \)

\( I_T E_R : N I_T S \)

Set integration parameters: INTR, DELTP, IPRINT, ISPs1, ICL1, ISPs2, ICL2, ACK, INCMT

Call VMP to propagate current iterate BS

\( I_S_T_O_P = ? \)

\( I_S_P = ? \)

\( S_M_O_R_E = ? \)

\( I_T_E_R = 1 \) or IBAD = 1

Call TAROPT(2) to compute desired and actual target (DTAR,ATAR) and auxiliary (DAUX,AAUX) parameters and differences

\( A_E_R(1) = |D_T_A_R(1) - A_T_A_R(1)| \)

\( D_E_V(1) = D_A_U_X(1) - A_A_U_X(1) \)

Enter "Outer Targeting"

Set NOS01 = 1

Store original parameters SOL, TOL

Set up artificial parameters KEYTAK, KAXTAR, DTAR, DAUX, CTOL, ISTOP, ITARN, IBAD, NITS, DVMAX, FERV

TARGET-7
TARGET-8

LOWHI=?

=1

Call TARMAX for computation of low level phase 2 targeting matrix

=0

Are all errors tolerable? AER(i) < CTOL(i) NO

YES

F

Set parameters for "Phase 2 Targeting"
RIN-TVH, ITARM=1, ISTART=2,
LEV=LEVELS=1, LOWHI=0

NOSOI=?

=1

Exit from "Outer Targeting"
Set NOSOI = 0
Restore CTOL, SPHERE
Set parameters for return to inner targeting: LEV, NOMORE, PERV, DMVMAX

ITOL=1

NOPHAS=?

=1

B

ISTART=?

=2

B

LEV=?

=1

TVL = RIN

#1

LEV:LEVELS

TVH=RIN, RIN-TVH
LOWHI=1, ISTART=0

B
BAD-STEP REDUCTION

Compute current scalar measure of error = CERROR using factors FAC

IBAD = ?

CERROR > PERROR

IBAD = IBAD + 1

FACBS = FACBS * REDUC

NEXT ITERATE

METHOD = ?

ITMAX = ?

Call TAREMA for computation of targeting matrix PHI for Newton-Raphson procedure.

IEND = ?

DELTAV = PHI * DEV

Ensure that each component of DELTAV & IMAX

RIN = RIN + DELTAV

DELTAV = DELTAV / REDUC

RIN = RIN + DELTAV

Call DESERT for computation of DELTAV by steepest descent or conjugate gradient technique.

D

NITS = ?

TARGET-9
PREPARATIONS FOR RETURN

H

Compute total \( \Delta v \) refinement
\[ \text{DELTAV} = \text{RIN} - \text{RIS} \]
and restore variables
\[ \text{RIN} = \text{RIS} \]
\[ \text{SPHERE} \]

RETURN
SUBROUTINE TARMAX

PURPOSE: TO CALCULATE A TARGET MATRIX FROM NOMINAL INJECTION CONDITIONS, AND A PERTURBATION FACTOR DELV FOR A GIVEN ACCURACY LEVEL.

CALLING SEQUENCE: CALL TARMAX

SUBROUTINES SUPPORTED: TARGET

SUBROUTINES REQUIRED: MATIN TAROPT VMP

LOCAL SYMBOLS: ACK ACCURACY USED TO GENERATE THE TARGET MATRIX
               AER DIFFERENCES BETWEEN DESIRED AND ACTUAL END CONDITIONS
               AUXN NOMINAL AUXILIARY END CONDITIONS
               CHI STATE TRANSITION MATRIX RELATING PERTURBATIONS IN THE RIN VECTOR TO CHANGES IN AUXN
               DVEE VECTOR OF VELOCITY COMPONENT PERTURBATIONS
               ISP2 INDICATOR USED BY SUBROUTINE VMP
                       =0 DO NOT STOP AT SPHERE OF INFLUENCE
                       =1 STOP AT SPHERE OF INFLUENCE
               I INDEX
               J INDEX
               KOMP INDEX
               PSI TARGET MATRIX FOR 2 X 2 CASE, STORED INTO PHI
               RSF FINAL SPACECRAFT STATE RETURNED BY VMP

COMMON COMPUTED/USED: ISPH PHI RIN TRTM

COMMON COMPUTED: JCL2 ICL INCMT

COMMON USED: AAUX AC ATAR CTOL DAUX
              DELTAT DELTAV DTAR D1 ISTOP
              KUR LEV LVLS NOPAR PERV

ZERO
TARMAX Analysis

TARMAX computes the targeting matrix used by TARGET for Newton-Raphson refinements. The targeting matrix is computed by numerical differencing.

Let the current iterate initial state be denoted $\mathbf{r}$, $\mathbf{v}$. Let the auxiliary parameters corresponding to this state be $\mathbf{a}$. Let the perturbation size for the sensitivities be $\Delta \mathbf{v}$.

The $k$-th column of the sensitivity matrix is computed as follows. Perturb the $k$-th component of velocity by $\Delta v$:

$$
\frac{\mathbf{v}}{p} = \frac{\mathbf{v}}{p} + \Delta v \left[ \delta_{1k}, \delta_{2k}, \delta_{3k} \right]^T
$$

(1)

Propagate the perturbed initial state $(\mathbf{r}_p, \frac{\mathbf{v}}{p})$ to the final stopping conditions. Let the auxiliary parameters of that trajectory be denoted $\mathbf{a}_p$.

The $k$-th column of the sensitivity matrix $\mathbf{x}$ is then given by

$$
\mathbf{x}_k = \frac{\mathbf{a}_p - \mathbf{a}}{\Delta v}
$$

(2)

Having computed all the columns of $\mathbf{x}$, the targeting matrix is then given by the inverse of $\mathbf{x}$:

$$
\mathbf{\phi} = \mathbf{x}^{-1}
$$

(3)

The targeting matrix then has the property that to obtain a change $\Delta \mathbf{a}$ in the terminal auxiliary parameters, the velocity should be changed by the amount

$$
\Delta \mathbf{v} = \mathbf{\phi} \cdot \Delta \mathbf{a}
$$

(4)
Set KOMP = 0, set up accuracy level, perturbation \( \Delta v \), and save nominal auxiliary values \( \bar{\alpha} \).

Set up ISP2, ICL2 flags based on ISTOP flag.

A

KOMP = KOMP + 1

B

\[ \bar{v} = \bar{v} + \Delta v \text{ (KOMP)} \]

Call VMP to integrate trajectory to stopping conditions.

Did trajectory miss SOI and ISTOP \( \neq 1 \)

\[ \bar{v} = \bar{v} - \Delta v \text{ (KOMP)} \]
\[ \Delta v \text{ (KOMP)} = \Delta v \text{ (KOMP)}/a \]

NO

Call TAROPT(3) to compute and store trajectory target parameters \( \bar{\alpha}_p \) and auxiliary parameters \( \bar{\alpha}_p ' \).

Do target parameters \( v_p \) satisfy tolerances?

YES

IEND = 1

RETURN

NO

Compute KOMP column of sensitivity matrix

\[ x_k = \frac{\bar{\alpha}_p - \bar{\alpha}}{\Delta v \text{ (KOMP)}} \]

KOMP: NOPAR

Compute targeting matrix \( \phi = x^{-1} \)

RETURN
SUBROUTINE TAROPT

PURPOSE: TO COMPUTE THE DESIRED AND ACHIEVED TARGET PARAMETER VALUES FOR ALL THE TARGETING SUBROUTINES.

CALLING SEQUENCE: CALL TAROPT(ITARO)

ARGUMENTS:

ITARO I

OPTION FLAG
=1 SET UP TARGETING PARAMETERS FOR TARGET KEYS
=2 COMPUTE ACTUAL VALUES OF PARAMETERS
=3 COMPUTE ACTUAL AND DESIRED VALUES OF PARAMETERS

SUBROUTINES SUPPORTED: TARGET TARMAX DESENT

SUBROUTINES REQUIRED: CAREL CPWMS IMPACT

LOCAL SYMBOLS:

ACK CURRENT ACCURACY BEING USED
A SEMI-MAJOR AXIS OF THE TARGET PLANETOCENTRIC CONIC
CPT TOTAL COMPUTER TIME USED (SECS)
DBOR DESIRED VALUE OF B DOT R
DBDT DESIRED VALUE OF B DOT T
DINC DESIRED VALUE OF INCLINATION
DRCA DESIRED VALUE OF RCA
E ECCENTRICITY OF THE TARGET PLANETOCENTRIC CONIC
IAUX INDICATOR FOR AUXILIARY END CONDITIONS
=0 TARGET TO ACTUAL END CONDITIONS
=1 TARGET TO AUXILIARY END CONDITIONS
IING LOCATES DESIRED INCLINATION IN THE DTAR ARRAY
IRCA LOCATES DESIRED RCA IN THE DTAR ARRAY
I INDEX
KEY LOCAL VARIABLE USED TO COMPLETE INFORMATION IN THE KAXTAR AND KEYTA2 ARRAY
PP DUMMY VARIABLE FOR CALL TO CAREL
QQ DUMMY VARIABLE FOR CALL TO CAREL
RM  MAGNITUDE OF SPACECRAFT USED TO COMPUTE SEMI-MAJOR AXIS
TA  DUMMY ARGUMENT FOR CALL TO CAREL
TDBR DUMMY ARGUMENT FOR CALL TO IMPACT
TDBT DUMMY ARGUMENT FOR CALL TO IMPACT
TFP  TIME OF FLIGHT FROM PERIAPSIS ON THE TARGET PLANETOCENTRIC CONIC
TINC  INTERMEDIATE VARIABLE TO COMPUTE CPT
TSICA DUMMY VARIABLE FOR CALL TO IMPACT
VX  INTERMEDIATE VARIABLE USED TO CALCULATE SEMI-MAJOR AXIS FOR OPTION 9
WM DUMMY VARIABLE FOR CALL TO CAREL
W  DUMMY VARIABLE FOR CALL TO CAREL
XI DUMMY VARIABLE FOR CALL TO CAREL
XN  DUMMY VARIABLE FOR CALL TO CAREL

COMMON COMPUTED/USED:
AAUX  ATAR  DAUX  DELTAT  DSTAR
ISTOP  KAXTAR  KEYTAR  NOPAR  NOPHAS
RCA

COMMON COMPUTED:
CTOL  FAC

COMMON USED:
AC  BDR  BDT  CAINC  OC
DG  DSI  DT  EQECP  IBAD
ICL2 INGMT  IPHASE  KTAR  KUR
LEV  NOSO1  NPAR  ONE  RC
RIN  RRF  RSI  TAR  TIMS
TMU  TM  TOL  TWO  VSI
TAROPT Analysis

TAROPT is responsible for computing the desired and achieved target parameter values for all the targeting subroutines. Thus to add any new target parameters TAROPT is the only subroutine that must be modified.

The key variables used by TAROPT and their definitions are

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTAR(6,10)</td>
<td>Codes of target parameters of all targeting events</td>
</tr>
<tr>
<td>TAR(6,10)</td>
<td>Desired values of target parameters of all targeting events</td>
</tr>
<tr>
<td>KEYTAR(3)</td>
<td>Codes of target parameters of current event</td>
</tr>
<tr>
<td>DTAR(3)</td>
<td>Desired values of target parameters of current event</td>
</tr>
<tr>
<td>ATAR(3)</td>
<td>Actual values of target parameters of current iterate</td>
</tr>
<tr>
<td>KAXTAR(3)</td>
<td>Codes of auxiliary parameters of current iterate</td>
</tr>
<tr>
<td>DAUX(3)</td>
<td>Desired values of auxiliary parameters on current iterate</td>
</tr>
<tr>
<td>AAUX(3)</td>
<td>Actual values of auxiliary parameters of current iterate</td>
</tr>
</tbody>
</table>

The available target parameters and their codes and definitions are

<table>
<thead>
<tr>
<th>Code</th>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>t_SI</td>
<td>Available for use</td>
</tr>
<tr>
<td>2</td>
<td>t_CS</td>
<td>Time at SOI of target body (n-body integration to SOI)</td>
</tr>
<tr>
<td>3</td>
<td>t_CA</td>
<td>Time at CA (n-body integration to SOI, conic propagation to CA)</td>
</tr>
<tr>
<td>4</td>
<td>t_CA</td>
<td>Time at CA (n-body integration to CA)</td>
</tr>
<tr>
<td>5</td>
<td>B · T</td>
<td>Impact parameter B · T</td>
</tr>
<tr>
<td>6</td>
<td>B · R</td>
<td>Impact parameter B · R</td>
</tr>
<tr>
<td>7</td>
<td>i</td>
<td>Inclination to target planet equator</td>
</tr>
<tr>
<td>8</td>
<td>r_CA</td>
<td>Radius of closest approach to target body</td>
</tr>
<tr>
<td>9</td>
<td>a_SI</td>
<td>Semi-major axis of conic w.r.t target body</td>
</tr>
<tr>
<td>10</td>
<td>x_f</td>
<td>X-component of final state (inertial ecliptic system)</td>
</tr>
<tr>
<td>11</td>
<td>y_f</td>
<td>Y-component of final state</td>
</tr>
<tr>
<td>12</td>
<td>z_f</td>
<td>Z-component of final state</td>
</tr>
</tbody>
</table>

The term target parameter refers to a variable whose final value is to conform to a desired value. The term auxiliary parameter refers to a variable which is used to compute the progressive corrections. The target parameters and auxiliary parameters are identical unless the target parameters i and r_CA are used. In this case the more linear variables B · T and B · R are used in their place as auxiliary parameters. The desired values of B · T and B · R are then computed (by IMPACT) based on the desired values of i and r_CA and the approach asymptote.
TAROPT is called under three different options distinguished by an argument ITARO. The three different options will be discussed in order.

TAROPT(1) is called by TARGET at the beginning of each phase to set up the proper variables for the targeting. The arrays KEYTAR, KAXTAR, DTAR, and DAUX are set to the current event values of KTAR and TAR. If $t_{CA}$ is a target parameter, the number of phases NOPHAS is set to 2. If the current phase is the first phase of a two-phase problem, $t_{CA}$ is replaced by $t_{CS}$ in the KEYTAR and KAXTAR arrays. If $i$ and $t_{CA}$ are target parameters, the corresponding indices of the KAXTAR array are set up for $B\cdot T^*$ and $B\cdot R^*$. TAROPT then sets up the integration parameters. The integration time interval $\Delta t$ is set to the nominal difference of the current guidance event time and target time:

$$\Delta t = t_{G} - t_{T}$$

Then if none of the target times are triggered ISTOP is set to 1 so that the integration proceeds to the target time exactly. If the target time is $t_{SI}$ or $t_{CS}$, ISTOP is set to 2 and $\Delta t = 1.1 \Delta t$. Thus the integration will be stopped at the target body SOI. Finally if the target time is $t_{CA}$, ISTOP is set to 3 and $\Delta t = 1.1 \Delta t$. For this case the integration will be stopped at closest approach to the target body. Finally the weighting factors FAC(3) to be used in computing the scalar loss function are set.

Since all auxiliary parameters are units of length except for the time parameters only the relative weight of time to length need be input. Thus the length factors are set to unity, the time factor is set to the input parameter WHTM.

TAROPT(2) is called by TARGET after integrating each iterate to the final stopping conditions. Here TAROPT perform mainly a bookkeeping role. It must fill the proper cells of the ATAR, AAX, and DAUX arrays with values generally computed by the virtual mass routines. The desired values of $B\cdot T^*$ and $B\cdot R^*$ are computed by calling IMPACT if needed.

TAROPT(3) is called by TARMAX and DESENT after integrating each perturbed trajectory to compute the perturbed values of the auxiliary parameters. Thus the desired values of DAUX need not be computed at this time. Once again, this task is simply a bookkeeping job to store the trajectory data correctly in the ATAR and AAX cells. TARMAX and DESENT may then operate easily on these arrays to compute the targeting matrix or gradient directions.

In both calls TAROPT(2) and TAROPT(3) the trajectory data are printed out before exiting from TAROPT.
TABOFT Flow Chart

ENTER

\[ Q = I + 1 \]

Store parameters for current event
NOPAR, KEYTAR, KAXTAR, DVAR, DAUX, CTOL

Is \( t_{CA} \) target?

NO

\[ I = 0 \]
\[ I_{AUX} = 0 \]

A

\[ I = I + 1 \]
\[ KEY = KEYTAR(I) \]

YES

\[ KEY = 4 ? \]

NO

\[ KEY = ?? \]

NO

\[ KEY = 8 ? \]

NO

\[ I = ? \]

B

YES

\[ I_{PHASE} = ? \]

\[ KAXTAR(I) = 3 \]
\[ KEYTAR(I) = 3 \]

YES

\[ KAXTAR(I) = 5 \]
\[ I_{AUX} = 1 \]

YES

\[ KAXTAR(I) = 6 \]
\[ I_{AUX} = 1 \]

= 3
= 2
= 1
I = 0
ISTOP = 1
Δt = t - t_G

I = I + 1
KEY = KAXTAR(1)

KEY = 9?
YES → ISTOP = 2
NO

KEY = 2 or KEY = 3?
YES → ISTOP = 2
 t = 1.1 Δt
NO

KEY = 4?
YES → ISTOP = 3
 t = 1.1 Δt
NO

I = ?
I = 3 → C

FAC(1) = FAC(2) = FAC(3) = 1.

Is KAXTAR(1) ≤ 4?
YES → FAC(4) = WGHXM
NO

RETURN
Set ATAR = AUX

IAUX = ?

KEYTAR(1) = ?

ISTOP = ?

Call CAREL to compute a,e

ATAR(1) = rCA

r = a(1-e)

KEYTAR(1) = ?

ATAR(1) = INC

ITARO = ?

IAUX = ?

DAUX(IRCA) = B·T*
DAUX(INC) = B·R*

Record data

RETURN
SUBROUTINE TARPR

PURPOSE: TO COMPUTE THE PARTIAL DERIVATIVES OF THE POSITION COMPONENTS OF A PLANET WITH RESPECT TO EACH OF ITS ORBITAL ELEMENTS.

CALLING SEQUENCE: CALL TARPR.(ICODE,PAR)

ARGUMENTS:
ICODE  I  CODE DEFINING ORBITAL ELEMENT OF INTEREST
PAR     Q  VECTOR OF 3 POSITION PARTIALS WITH RESPECT TO THE ORBITAL ELEMENT OF INTEREST

SUBROUTINES SUPPORTED: TRAKS  TRAKH

LOCAL SYMBOLS:
CBO    COSINE OF LONGITUDE OF ASCENDING NODE
CI     COSINE OF ANGLE OF INCLINATION
CLO    COSINE OF ARGUMENT OF PERIAPSIS
COSNU  COSINE OF TRUE ANOMALY
COSONU COSINE OF THE SUM OF THE ANGLES OF TRUE ANOMALY PLUS THE ARGUMENT OF PERIAPSIS
DNUDE  PLANET DISTANCE TIMES THE PARTIAL OF TRUE ANOMALY WITH RESPECT TO ECCENTRICITY
DNUDEM PARTIAL OF TRUE ANOMALY WITH RESPECT TO MEAN ANOMALY
DOPAR  1./R*(R/E + (R*NU)*E)
WHERE R= PLANET DISTANCE
      NU=TRUE ANOMALY
      E=ECCENTRICITY
      = PARTIAL OF
DRNU   PARTIAL R WITH RESPECT TO NU
E2     SQUARE OF ECCENTRICITY
IND    INDEX USED IN ARRAY STORING ORBITAL ELEMENTS OF PLANETS
PCOMP  SEMI-MAJOR AXIS TIMES THE TERM (1-E=E)
WHERE E=ECCENTRICITY
R      PLANET DISTANCE
SBO    SINE OF LONGITUDE OF THE ASCENDING NODE
SI     SINE OF INCLINATION

535
SINNU  SINE OF TRUE ANOMALY
SINOMU SINE OF THE SUM OF THE ANGLES OF TRUE ANOMALY PLUS THE ARGUMENT OF PERIAPSIS
SLO   SINE OF THE ARGUMENT OF PERIAPSIS
XXX   SCRATCH CELL

COMMON USED:
ALNGTH ELMNT NTP ONE XP ZERO
The position components of a planet are related to its orbital elements $a$, $e$, $i$, $\Omega$, $\omega$, and $\mu$ through the following set of equations:

\[
x = r \left[ \cos \Omega \cos(\omega + \nu) - \sin \Omega \sin(\omega + \nu) \cos i \right]
\]
\[
y = r \left[ \sin \Omega \cos(\omega + \nu) + \cos \Omega \sin(\omega + \nu) \cos i \right]
\]
\[
z = r \sin(\omega + \nu) \sin i
\]
\[
r = a(1 - e^2)
\]
\[
1 + e \cos \nu
\]
\[
\tan \frac{\nu}{2} = \sqrt{\frac{1 + e}{1 - e}} \tan \frac{\mu}{2}
\]
\[
M = E - e \sin E
\]

We can write equations (1), (2), (3), and (4) symbolically as

\[x_{1} = f_{1}(a, e, i, \Omega, \omega, \nu)\]

and equations (5) and (6) as

\[\nu = \nu(e, M)\]

Then the partials of $x_{1}$ with respect to $a$, $e$, $i$, $\Omega$, $\omega$, and $M$ can be evaluated as follows:

\[
\frac{\partial x_{1}}{\partial a} = \frac{\partial f_{1}}{\partial a}
\]
\[
\frac{\partial x_{1}}{\partial e} = \left( \frac{\partial f_{1}}{\partial e} \right)_{\nu} + \frac{\partial f_{1}}{\partial \nu} \cdot \frac{\partial \nu}{\partial e}
\]
\[
\frac{\partial x_{1}}{\partial i} = \frac{\partial f_{1}}{\partial i}
\]
\[
\frac{\partial x_{1}}{\partial \Omega} = \frac{\partial f_{1}}{\partial \Omega}
\]
\[ \frac{\partial x_1}{\partial \omega} = \frac{\partial f_1}{\partial \omega} \]  
\[ \frac{\partial x_1}{\partial \nu} = \frac{\partial f_1}{\partial \nu} \cdot \frac{\partial \nu}{\partial \nu} \] (12)

Only \( \frac{\partial \nu}{\partial e} \) and \( \frac{\partial \nu}{\partial \nu} \) require further consideration before equations (7) through (11) can be used to obtain expressions for the 13 desired partial derivatives.

We obtain \( \frac{\partial \nu}{\partial \nu} \) by first differentiating equation (5) with respect to \( \frac{\partial \nu}{\partial \nu} \) and equation (6) with respect to \( \frac{\partial \nu}{\partial \nu} \) to obtain

\[ \frac{\partial \nu}{\partial \nu} = \frac{\partial \nu}{\partial \nu} \cdot \frac{\partial \nu}{\partial \nu} = \left( \frac{a}{r} \right)^2 \sqrt{1 - e^2} \] (13)

We obtain \( \frac{\partial \nu}{\partial e} \) by first differentiating equation (6) with respect to \( e \) to obtain

\[ \frac{\partial \nu}{\partial e} = \frac{\sqrt{1 - e^2} \sin \nu}{1 + e \cos \nu} \]

This result is then combined with equation (13) to yield

\[ \frac{\partial \nu}{\partial e} = \frac{\partial \nu}{\partial \nu} \cdot \frac{\partial \nu}{\partial \nu} = \left( \frac{a}{r} \right)^2 \frac{(1 - e^2) \sin \nu}{1 + e \cos \nu} \] (14).
The evaluation of the desired partials can now proceed. The results are summarized below.

a. Partials with respect to $a$.

$$\frac{\partial x}{\partial a} = \frac{x}{a}$$

$$\frac{\partial y}{\partial a} = \frac{y}{a}$$

$$\frac{\partial z}{\partial a} = \frac{z}{a}$$

b. Partials with respect to $e$.

$$\frac{\partial x}{\partial e} = \frac{xq + r \frac{\partial y}{\partial e}}{r} \left[ - \cos \Omega \sin(\omega + \nu) - \sin \Omega \cos(\omega + \nu) \cos \iota \right]$$

$$\frac{\partial y}{\partial e} = \frac{yq + r \frac{\partial y}{\partial e}}{r} \left[ - \sin \Omega \sin(\omega + \nu) + \cos \Omega \cos(\omega + \nu) \cos \iota \right]$$

$$\frac{\partial z}{\partial e} = \frac{zq + r \frac{\partial y}{\partial e}}{r} \cos(\omega + \nu) \sin \iota$$

where $q = \frac{r}{ae(l - e^2)} \left[ r - a - ae^2(l + \sin^2 \nu) \right]$.

c. Partials with respect to $i$.

$$\frac{\partial x}{\partial i} = r \sin \Omega \sin(\omega + \nu) \sin \iota$$

$$\frac{\partial y}{\partial i} = - r \cos \Omega \sin(\omega + \nu) \sin \iota$$

$$\frac{\partial z}{\partial i} = r \sin(\omega + \nu) \cos \iota$$

d. Partials with respect to $\Omega$.

$$\frac{\partial x}{\partial \Omega} = - y$$
\frac{\delta y}{\delta \Omega} = x \\
\frac{\delta z}{\delta \Omega} = 0 \\

a. Partial with respect to \omega.

\frac{\delta x}{\delta \omega} = r \left[ - \cos \Omega \sin (\omega + \nu) - \sin \Omega \cos (\omega + \nu) \cos i \right] \\
\frac{\delta y}{\delta \omega} = r \left[ - \sin \Omega \sin (\omega + \nu) + \cos \Omega \cos (\omega + \nu) \cos i \right] \\
\frac{\delta z}{\delta \omega} = r \cos (\omega + \nu) \sin i \\

f. Partial with respect to \mu.

\frac{\partial x}{\partial \mu} = \frac{x_s}{r} + r \frac{\delta y}{\partial \mu} \left[ - \cos \Omega \sin (\omega + \nu) - \sin \Omega \cos (\omega + \nu) \cos i \right] \\
\frac{\partial y}{\partial \mu} = \frac{y_s}{r} + r \frac{\delta y}{\partial \mu} \left[ - \sin \Omega \sin (\omega + \nu) + \cos \Omega \cos (\omega + \nu) \cos i \right] \\
\frac{\partial z}{\partial \mu} = \frac{z_s}{r} + r \frac{\delta y}{\partial \mu} \cos (\omega + \nu) \sin i \\

where \ s = \frac{a e \sin \nu}{\sqrt{1 - e^2}} \\

TARFRL Flow Chart

ENTER

Compute the position magnitude of the target planet relative to the Sun.

Compute sines and cosines of \( \nu, \Omega, \omega, i, \) and \( \omega + \varpi \) of the target planet.

Compute the partial derivatives of the position components of the target planet with respect to the orbital element indicated by IC\( \varnothing \)DE.

RETURN
SUBROUTINE TIME

PURPOSE: TO COMPUTE THE JULIAN DATE, EPOCH 1900, FROM THE CALENDAR DATE OR TO COMPUTE THE CALENDAR DATE FROM THE JULIAN DATE.

CALLING SEQUENCE: CALL TIME(DAY,IYR,MO,IDAY,IMR,MIN,SEC,ICODE)

ARGUMENTS:

DAY I/O JULIAN DATE, EPOCH 1900
IYR O/I CALENDAR YEAR
MO O/I CALENDAR MONTH
IDAY O/I CALENDAR DAY
IMR O/I HOUR OF THE DAY
MIN O/I MINUTE OF HOUR
SEC O/I FRACTIONAL SECONDS
ICODE I OPERATIONAL MODE
   = 1, INDICATES THE JULIAN DATE IS INPUT, CALENDAR DATE IS OUTPUT
   = 0, INDICATES THE CALENDAR DATE IS INPUT, JULIAN DATE IS OUTPUT

SUBROUTINES SUPPORTED: DATAS INPUTZ PRINT VMP GIDANS
PREPUL PRNTS4 DATA PRNTS3 PRELIM
GIDANS HELIO MULTAR

SUBROUTINES REQUIRED: NONE

LOCAL SYMBOLS:

IA NUMBER OF CENTURIES
IB YEARS IN PRESENT CENTURY
IP NUMBER OF MONTH (BASED ON MARCH AS NUMBER ZERO)
IQ NUMBER OF YEARS
IR NUMBER OF CENTURIES DIVIDED BY 4
IS NUMBER OF YEARS SINCE LAST 400 YEAR SECTION BEGAN
IT NUMBER OF LEAP YEARS IN PRESENT CENTURY
IU NUMBER OF YEARS SINCE LAST LEAP YEAR
IV NUMBER OF DAYS IN LAST YEAR
IX  INTERMEDIATE INTEGER
J  INTERMEDIATE INTEGER
JD  NUMBER OF DAYS IN JULIAN DATE
P  JULIAN DATE
R  FRACTIONAL PORTION OF DAY IN JULIAN DATE
SUBROUTINE TITLE

PURPOSE: TO PRINT TITLES FOR ERRAN.

CALLING SEQUENCE: CALL TITLE(LINES, TEVN, ICODE)

ARGUMENTS:
- LINES NOT USED
- TEVN I EVENT TIME
- ICODE I EVENT CODE

SUBROUTINES SUPPORTED: SETEVN

LOCAL SYMBOLS:
- TPT TIME PREDICTING TO

COMMON USED:
- IPROB
- NPE
- TPT2
SUBROUTINE TITLES

PURPOSE: TO PRINT TITLES FOR SIMUL.

CALLING SEQUENCE: CALL TITLES(TEVN,ICODE)

ARGUMENTS: TEVN I EVENT TIME
ICODE I EVENT CODE

SUBROUTINES SUPPORTED: SETEVS

LOCAL SYMBOLS: TPT TIME PREDICTING TO

COMMON USED: IPROB NPE TPT2
SUBROUTINE TRAKM

PURPOSE: THE OBSERVATIONS AND OBSERVATION MATRIX FOR A GIVEN TYPE OF MEASUREMENT IS COMPUTED BY THIS ROUTINE.

CALLING SEQUENCE: CALL TRAKM(HECV, ITRK, NR, IOBS, VECTOR)

ARGUMENTS:
- HECV: I POSITION AND VELOCITY OF SPACECRAFT AT TIME OF MEASUREMENT
- IOBS: I CODE WHICH SPECIFIES IF MEASUREMENT OR OBSERVATION MATRIX IS TO BE COMPUTED
- ITRK: I CODE WHICH SPECIFIES MEASUREMENT TYPE (CALLED MMCODE ELSEWHERE IN PROGRAM)
- NR: O NUMBER OF ROWS IN THE OBSERVATION MATRIX
- VECTOR: O ACTUAL MEASUREMENT

SUBROUTINES SUPPORTED: ERRANN

SUBROUTINES REQUIRED: EPHEM ORB STAPRL TARPR.

LOCAL SYMBOLS:
- AD1: INTERMEDIATE VARIABLE
- AD2: INTERMEDIATE VARIABLE
- AD3: INTERMEDIATE VARIABLE
- AL: ALTITUDE
- ALAT: LATITUDE
- ALOM: LONGITUDE
- A1: PARTIAL OF RANGE WITH RESPECT TO X
- A2: PARTIAL OF RANGE WITH RESPECT TO Y
- A3: PARTIAL OF RANGE WITH RESPECT TO Z
- A1: PARTIAL OF RANGE-RATE WITH RESPECT TO X
- A2: PARTIAL OF RANGE-RATE WITH RESPECT TO Y
- A3: PARTIAL OF RANGE-RATE WITH RESPECT TO Z
- CE: COSINE OF OBLIQUITY OF EARTH
- COAL: COSINE OF STAR-PLANET ANGLE
- CP: COSINE OF LONGITUDE + CONSTANT
DADP PARTIALS OF STAR-PLANET ANGLE INDEX INCREMENT VALUE
TO VEHICLE POSITION AND VELOCITY

DBDP PARTIALS OF APPARENT PLANET DIAMETER WITH RESPECT TO VEHICLE POSITION AND VELOCITY

DENOM INTERMEDIATE VARIABLE

D INTERMEDIATE TIME

EK RANGE-RATE PARTIAL WITH RESPECT TO STATION LOCATION ERRORS

GECS GECCENTRIC EQUATORIAL COORDINATES OF STATION

GELS GECCENTRIC ECLIPTIC COORDINATES OF STATION

HECE COORDINATES OF EARTH

HECP COORDINATES OF TARGET PLANET

IA TRACKING STATION LOCATION SELECTION CODE

IC CODE CORRESPONDING TO TRACKING STATION LOCATION ERRORS

IC COLUMN NUMBER IN OBSERVATION MATRIX
PARTITION WHERE EK IS TO BE STORED

IEND VARIABLE INDEX VALUE

IR STAR-PLANET ANGLE INDEX INCREMENT VALUE

NA STAR-PLANET ANGLE INDEX LOWER LIMIT
=1 FOR 3 STAR-PLANET ANGLES
=ITRK-10 FOR SINGLE STAR-PLANET ANGLE

NC STAR-PLANET ANGLE INDEX UPPER LIMIT
=3 FOR 3 STAR-PLANET ANGLES
=ITRK-10 FOR SINGLE STAR-PLANET ANGLE

PAR PARTIALS RETURNED FROM SUBROUTINE IADP

PAT1 INTERMEDIATE VARIABLE

PAT2 INTERMEDIATE VARIABLE

PA PARTIALS

RADNTP RADIUS OF TARGET PLANET
RRATE: RANGE-RATE
R1: RANGE
R2: SQUARE OF RANGE
SA: PARTIALS OF STAR-PLANET ANGLES WITH RESPECT TO VEHICLE POSITION
SE: SINE OF OBLIQUITY OF EARTH
SIAL: SINE OF STAR-PLANET ANGLE
SP: SINE OF LONGITUDE + CONSTANT
VEC: INTERMEDIATE VECTOR

COMMON COMPUTED/USED:
AAL AM H NO T
XP

COMMON COMPUTED:
G

COMMON USED:
ALNGTH DATEJ DELTM EMJ EPS
F IAUGDC IAUGIN IAUGMC IAUG
IBARY NBOD N8 NTP OMEGA
ONE RADIUS SAL SLAT SLON
TH TRTM1 TWO UNIVT UST
VST WST ZERO
TRAKM Analysis

Subroutine TRAKM computes observation matrix partitions in the error analysis mode. It is completely equivalent to the simulation mode subroutine TRAXS with IOBS always set to zero. See subroutine TRAXS for further analytical details. A flow chart is not presented for TRAKM since it is but a subset of the TRAXS flow chart.
SUBROUTINE TRAKS

PURPOSE: TO COMPUTE ALL OBSERVATION MATRIX PARTITIONS FOR THE MEASUREMENT TYPE AND TO COMPUTE THE MEASUREMENT ITSELF.

CALLING SEQUENCE: CALL TRAKS(HECV,ITRK,NR,IOBS,VECTOR)

ARGUMENTS:
- HECV: I POSITION AND VELOCITY OF SPACECRAFT AT TIME OF MEASUREMENT
- IOBS: I CODE WHICH SPECIFIES IF MEASUREMENT OR OBSERVATION MATRIX IS TO BE COMPUTED
- ITRK: I CODE WHICH SPECIFIES MEASUREMENT TYPE (CALLED MMCODE ELSEWHERE IN PROGRAM)
- NR: O NUMBER OF ROWS IN THE OBSERVATION MATRIX
- VECTOR: O ACTUAL MEASUREMENT

SUBROUTINES SUPPORTED: SIMULL

SUBROUTINES REQUIRED: EPHEM ORB STAPRL TARPL

LOCAL SYMBOLS:
- AD1: INTERMEDIATE VARIABLE
- AD2: INTERMEDIATE VARIABLE
- AD3: INTERMEDIATE VARIABLE
- ALAT: LATITUDE
- ALOM: LONGITUDE
- AL: ALTITUDE
- A1: PARTIAL OF RANGE WITH RESPECT TO X
- A2: PARTIAL OF RANGE WITH RESPECT TO Y
- A3: PARTIAL OF RANGE WITH RESPECT TO Z
- B1: PARTIAL OF RANGE-RATE WITH RESPECT TO X
- B2: PARTIAL OF RANGE-RATE WITH RESPECT TO Y
- B3: PARTIAL OF RANGE-RATE WITH RESPECT TO Z
- CE: COSINE OF OBLIQUITY OF EARTH
- COAL: COSINE OF STAR-PLANET ANGLE
- CP: COSINE OF LONGITUDE + CONSTANT
DAOP  PARTIALS OF STAR-PLANET ANGLE WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DBDP  PARTIALS OF APPARENT PLANET DIAMETER WITH RESPECT TO VEHICLE POSITION AND VELOCITY
DENOM  INTERMEDIATE VARIABLE
D  INTERMEDIATE TIME
EX  RANGE-RATE PARTIAL WITH RESPECT TO STATION LOCATION ERRORS
GECG  GEOCENTRIC EQUATORIAL COORDINATES OF STATION
GELS  GEOCENTRIC ECLIPTIC COORDINATES OF STATION
HECG  COORDINATES OF EARTH
HECG  COORDINATES OF TARGET PLANET
IA  TRACKING STATION LOCATION SELECTION CODE
ICO  CODE CORRESPONDING TO TRACKING STATION LOCATION ERRORS
IC  COLUMN NUMBER IN OBSERVATION MATRIX PARTITION WHERE IS TO BE STORED
IEND  VARIABLE INDEX VALUE
IR  STAR-PLANET ANGLE INDEX INCREMENT VALUE
WA  STAR-PLANET ANGLE INDEX LOWER LIMIT
NC  STAR-PLANET ANGLE INDEX UPPER LIMIT
PAR  PARTIALS RETURNED FROM SUBROUTINE TARPRL
PAT1  INTERMEDIATE VARIABLE
PAT2  INTERMEDIATE VARIABLE
PA  PARTIALS
RADNTP  RADIUS OF TARGET PLANET
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRATE</td>
<td>RANGE-RATE</td>
</tr>
<tr>
<td>R1</td>
<td>RANGE</td>
</tr>
<tr>
<td>R2</td>
<td>SQUARE OF RANGE</td>
</tr>
<tr>
<td>SA</td>
<td>PARTIALS OF STAR-PLANET ANGLES WITH RESPECT TO VEHICLE POSITION</td>
</tr>
<tr>
<td>SE</td>
<td>SINE OF OBLIQUITY OF EARTH</td>
</tr>
<tr>
<td>SIAL</td>
<td>SINE OF STAR-PLANET ANGLE</td>
</tr>
<tr>
<td>SP</td>
<td>SINE OF LONGITUDE + CONSTANT</td>
</tr>
<tr>
<td>VEC</td>
<td>INTERMEDIATE VECTOR</td>
</tr>
</tbody>
</table>

**COMMON COMPUTED/USED:**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAL</td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>XP</td>
<td></td>
</tr>
</tbody>
</table>

**COMMON USED:**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALNGTH</td>
<td></td>
</tr>
<tr>
<td>DATEJ</td>
<td></td>
</tr>
<tr>
<td>DELTM</td>
<td></td>
</tr>
<tr>
<td>EM3</td>
<td></td>
</tr>
<tr>
<td>EPS</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
<tr>
<td>IAUGDC</td>
<td></td>
</tr>
<tr>
<td>IAUGIN</td>
<td></td>
</tr>
<tr>
<td>IAUGMC</td>
<td></td>
</tr>
<tr>
<td>IAUG</td>
<td></td>
</tr>
<tr>
<td>IBARY</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>N8</td>
<td></td>
</tr>
<tr>
<td>NTP</td>
<td></td>
</tr>
<tr>
<td>OMEGA</td>
<td></td>
</tr>
<tr>
<td>ONE</td>
<td></td>
</tr>
<tr>
<td>RADIUS</td>
<td></td>
</tr>
<tr>
<td>SAL</td>
<td></td>
</tr>
<tr>
<td>SLAT</td>
<td></td>
</tr>
<tr>
<td>SLB</td>
<td></td>
</tr>
<tr>
<td>SLOM</td>
<td></td>
</tr>
<tr>
<td>TH</td>
<td></td>
</tr>
<tr>
<td>TRMB</td>
<td></td>
</tr>
<tr>
<td>TRMI</td>
<td></td>
</tr>
<tr>
<td>TWO</td>
<td></td>
</tr>
<tr>
<td>UNIUT</td>
<td></td>
</tr>
<tr>
<td>UST</td>
<td></td>
</tr>
<tr>
<td>VST</td>
<td></td>
</tr>
<tr>
<td>WST</td>
<td></td>
</tr>
<tr>
<td>ZERO</td>
<td></td>
</tr>
</tbody>
</table>

---

TRANS-C
Subroutine TRAKS performs two functions in the simulation mode. The first function, which corresponds to I0BS = 0, is to compute all observation matrix partitions for the measurement type indicated by ITRK. The second function, which corresponds to I0BS ≠ 0, is to compute the measurement itself. If I0BS = 1, TRAKS computes the measurement corresponding to the most recent nominal spacecraft state. If I0BS = 2, TRAKS computes the measurement corresponding to the actual spacecraft state, and, if the measurement is a range or range-rate measurement, to the actual tracking station locations. The number of rows, NR, in the measurement and the observation matrix partitions is also computed.

A general measurement has form

\[ \mathbf{Y} = \mathbf{Y}(\mathbf{x}, \mathbf{p}, t) \]

where \( \mathbf{x} \) is the spacecraft position/velocity state at time \( t \) and \( \mathbf{p} \) is a vector of parameters. This equation can be linearized about nominal \( \overline{\mathbf{x}} \) and \( \overline{\mathbf{p}} \) to obtain

\[ \delta \mathbf{Y} = \left( \frac{\partial \mathbf{Y}}{\partial \mathbf{x}} \right)^* \delta \mathbf{x} + \left( \frac{\partial \mathbf{Y}}{\partial \mathbf{p}} \right)^* \delta \mathbf{p} \]

where \( (\quad)^* \) indicates matrices are evaluated at the nominal condition. This perturbation equation can be rewritten as

\[ \delta \mathbf{Y} = \mathbf{H} \delta \mathbf{x} + \mathbf{M} \delta \mathbf{x}_d + \mathbf{G} \delta \mathbf{u} + \mathbf{L} \delta \mathbf{v} \]

where \( \mathbf{H} = \left( \frac{\partial \mathbf{Y}}{\partial \mathbf{x}} \right)^* \), and \( \left( \frac{\partial \mathbf{Y}}{\partial \mathbf{p}} \right)^* \) is distributed among the \( \mathbf{M}, \mathbf{G}, \) and \( \mathbf{L} \) partitions to correspond to the partition of the parameter vector \( \delta \mathbf{p} \) into solve-for parameters \( \delta \mathbf{x}_d \), dynamic consider parameters \( \delta \mathbf{u} \), and measurement consider parameters \( \delta \mathbf{v} \).

In the remainder of this section the measurement equation and all partial derivatives required to construct the \( \mathbf{H}, \mathbf{M}, \mathbf{G}, \) and \( \mathbf{L} \) observation matrix partitions will be summarized for each measurement type.

A. Range measurement \( \rho \).

A range measurement has form

\[ \rho = \rho(\mathbf{x}, R, \theta, \phi, t) \]

where \( R, \theta, \) and \( \phi \) are the radius, latitude, and longitude of the relevant tracking station.
More explicitly,
\[ \rho = \left( (X - X_E - X_S)^2 + (Y - Y_E - Y_S)^2 + (Z - Z_E - Z_S)^2 \right)^{\frac{1}{2}} \]

where \( X, Y, Z \) = inertial position components of spacecraft
\( X_E, Y_E, Z_E \) = inertial position components of Earth
\( X_S, Y_S, Z_S \) = station position components relative to Earth.

\( X_S, Y_S, \) and \( Z_S \) are related to \( R, \theta, \) and \( \phi \) as follows:
\[ X_S = R \cos \theta \cos \phi \]
\[ Y_S = R \cos \theta \cos \epsilon \sin \phi + R \sin \theta \sin \epsilon \]
\[ Z_S = -R \cos \theta \sin \epsilon \sin \phi + R \sin \theta \cos \epsilon \]

where \( \epsilon \) is the obliquity of the Earth, and
\[ G = \phi + \text{GHA} \]
where GHA is the Greenwich hour angle at time \( t \).

Partials of \( \rho \) with respect to spacecraft state are given by
\[ \frac{\partial \rho}{\partial x} = \frac{1}{\rho} (X - X_E - X_S) \quad \frac{\partial \rho}{\partial x} = 0 \]
\[ \frac{\partial \rho}{\partial y} = \frac{1}{\rho} (Y - Y_E - Y_S) \quad \frac{\partial \rho}{\partial y} = 0 \]
\[ \frac{\partial \rho}{\partial z} = \frac{1}{\rho} (Z - Z_E - Z_S) \quad \frac{\partial \rho}{\partial z} = 0 \]

Partials of \( \rho \) with respect to \( R, \theta, \) and \( \phi \) are given by
\[ \frac{\partial \rho}{\partial R} = \frac{\partial \rho}{\partial x} \frac{\partial x_S}{\partial R} + \frac{\partial \rho}{\partial y} \frac{\partial y_S}{\partial R} + \frac{\partial \rho}{\partial z} \frac{\partial z_S}{\partial R} \]
\[ \frac{\partial \rho}{\partial \theta} = \frac{\partial \rho}{\partial x} \frac{\partial x_S}{\partial \theta} + \frac{\partial \rho}{\partial y} \frac{\partial y_S}{\partial \theta} + \frac{\partial \rho}{\partial z} \frac{\partial z_S}{\partial \theta} \]
\[ \frac{\partial \rho}{\partial \phi} = \frac{\partial \rho}{\partial x} \frac{\partial x_S}{\partial \phi} + \frac{\partial \rho}{\partial y} \frac{\partial y_S}{\partial \phi} + \frac{\partial \rho}{\partial z} \frac{\partial z_S}{\partial \phi} \]
and the negatives of the partials of $X_S, Y_S$, and $Z_S$ with respect to $R, \theta,$ and $\phi$ are summarized in the subroutine STAPRL analysis.

B. Range-rate measurement $\dot{\rho}$.

A range-rate measurement has form

$$\dot{\rho} = \dot{\rho} (\bar{x}, R, \theta, \phi, t)$$

where all arguments have been defined previously. More explicitly,

$$\dot{\rho} = \frac{\rho_x \dot{\rho}_x + \rho_y \dot{\rho}_y + \rho_z \dot{\rho}_z}{\rho}$$

where

$$\rho_x = X - X_E - X_S \quad \rho_x = \dot{x} - \dot{x}_E - \dot{x}_S$$

$$\rho_y = Y - Y_E - Y_S \quad \rho_y = \dot{y} - \dot{y}_E - \dot{y}_S$$

$$\rho_z = Z - Z_E - Z_S \quad \rho_z = \dot{z} - \dot{z}_E - \dot{z}_S$$

$\dot{x}_S, \dot{y}_S,$ and $\dot{z}_S$ are related to $R, \theta,$ and $\phi$ as follows:

$$\dot{x}_S = -\omega R \cos \theta \sin \phi$$

$$\dot{y}_S = \omega R \cos \theta \cos \phi \cos \phi$$

$$\dot{z}_S = -\omega R \cos \theta \sin \phi \cos \phi$$

where $\omega$ is the rotational rate of the Earth.

Partials of $\dot{\rho}$ with respect to spacecraft state are given by

$$\frac{\partial \dot{\rho}}{\partial x} = \frac{\rho_x - \rho_x \dot{\rho}}{\rho}$$

$$\frac{\partial \dot{\rho}}{\partial y} = \frac{\rho_y - \rho_y \dot{\rho}}{\rho}$$

$$\frac{\partial \dot{\rho}}{\partial z} = \frac{\rho_z - \rho_z \dot{\rho}}{\rho}$$
The partial of \( \dot{\rho} \) with respect to \( R \) is given by

\[
\frac{\partial \dot{\rho}}{\partial R} = \frac{\partial \dot{\rho}}{\partial x_S} \cdot \frac{\partial x_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial y_S} \cdot \frac{\partial y_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial z_S} \cdot \frac{\partial z_S}{\partial R} + \\
\frac{\partial \dot{\rho}}{\partial \dot{x}_S} \cdot \frac{\partial \dot{x}_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial \dot{y}_S} \cdot \frac{\partial \dot{y}_S}{\partial R} + \frac{\partial \dot{\rho}}{\partial \dot{z}_S} \cdot \frac{\partial \dot{z}_S}{\partial R}
\]

where

\[
\frac{\partial \dot{\rho}}{\partial x_S} = -\frac{\partial \dot{\rho}}{\partial x}, \text{ etc.}
\]

and

\[
\frac{\partial \dot{\rho}}{\partial \dot{x}_S} = -\frac{\partial \dot{\rho}}{\partial \dot{x}}, \text{ etc.}
\]

The negatives of the partials of \( x_S, y_S, z_S, \dot{x}_S, \dot{y}_S, \) and \( \dot{z}_S \) with respect to \( R, \Theta, \) and \( \Phi \) are summarized in the subroutine STAPRL analysis. Partials of \( \dot{\rho} \) with respect to \( \Theta \) and \( \Phi \) are treated similarly.

C. Star-planet angle measurement \( \alpha \).

A star-planet angle measurement has form \( \alpha \)

\[
\alpha = \alpha (\dot{x}, a, e, i, \Omega, \omega, M)
\]

where \( a, e, i, \Omega, \omega, \) and \( M \) are the standard set of target planet orbital elements.

If we define \( \rho = (\rho_x, \rho_y, \rho_z) \) to be the position of the target planet relative to the spacecraft and \((u, v, w)\) to be the direction cosines of the relevant star, then

\[
\alpha = \cos^{-1} \left[ \frac{1}{\rho} (u\rho_x + v\rho_y + w\rho_z) \right]
\]

where

\[
\rho_x = \rho_x - X_p, \rho_y = \rho_y - Y_p, \rho_z = \rho_z - Z_p,
\]

and \((X_p, Y_p, Z_p)\) represent the position coordinates of the target planet.
Partial derivatives of $\alpha$ with respect to spacecraft state are given by

$$\frac{\partial \alpha}{\partial x} = \frac{1}{\sin \alpha} \left( \frac{x}{\rho} - \frac{p_x \cos \alpha}{\rho^2} \right) \quad \frac{\partial \alpha}{\partial y} = 0$$

$$\frac{\partial \alpha}{\partial y} = \frac{1}{\sin \alpha} \left( \frac{y}{\rho} - \frac{p_y \cos \alpha}{\rho^2} \right) \quad \frac{\partial \alpha}{\partial z} = 0$$

$$\frac{\partial \alpha}{\partial z} = \frac{1}{\sin \alpha} \left( \frac{z}{\rho} - \frac{p_z \cos \alpha}{\rho^2} \right)$$

where

$$\sin \alpha = 1 - \cos^2 \alpha$$

The partial of $\alpha$ with respect to target planet semi-major axis is given by

$$\frac{\partial \alpha}{\partial a} = \frac{\partial \alpha}{\partial x_p} \cdot \frac{\partial x_p}{\partial a} + \frac{\partial \alpha}{\partial y_p} \cdot \frac{\partial y_p}{\partial a} + \frac{\partial \alpha}{\partial z_p} \cdot \frac{\partial z_p}{\partial a}$$

where

$$\frac{\partial \alpha}{\partial x_p} = -\frac{\partial \alpha}{\partial x}, \quad \frac{\partial \alpha}{\partial y_p} = -\frac{\partial \alpha}{\partial y}, \quad \frac{\partial \alpha}{\partial z_p} = -\frac{\partial \alpha}{\partial z}$$

and partials of $x_p$, $y_p$, and $z_p$ with respect to semi-major axis are summarized in the subroutine TARPRPL analysis. Partial derivatives of $\alpha$ with respect to $x_p$, $y_p$, and $z_p$ do not appear in the above expression since they are all zero. Partial derivatives of $\alpha$ with respect to the remaining target planet orbital elements are treated similarly.

D. Apparent planet diameter measurement $\beta$

An apparent planet diameter measurement has form

$$\beta = \beta(\bar{x}, a, e, i, \Omega, \omega, M)$$

where all arguments have been defined previously.

Defining $\bar{p} = (p_x, p_y, p_z)$ to be the position of the target planet relative to the spacecraft and $R_p$ to be the radius of the target planet, the apparent planet diameter can then be written as

$$\beta = 2 \sin^{-1} \left( \frac{R_p}{\rho} \right)$$
Partials of $\beta$ with respect to spacecraft state are given by

$$\frac{\partial \beta}{\partial x} = \frac{2 R_p \rho_x}{\rho^2 \left[ \rho^2 - R_p^2 \right]^{\frac{1}{2}}}$$

$$\frac{\partial \beta}{\partial y} = \frac{2 R_p \rho_y}{\rho^2 \left[ \rho^2 - R_p^2 \right]^{\frac{1}{2}}}$$

$$\frac{\partial \beta}{\partial z} = \frac{2 R_p \rho_z}{\rho^2 \left[ \rho^2 - R_p^2 \right]^{\frac{1}{2}}}$$

$$\frac{\partial \beta}{\partial x} = 0$$

$$\frac{\partial \beta}{\partial y} = 0$$

$$\frac{\partial \beta}{\partial z} = 0$$

The partial of $\beta$ with respect to target planet semi-major axis is given by

$$\frac{\partial \beta}{\partial a} = \frac{\partial \beta}{\partial x_P} \cdot \frac{\partial x_P}{\partial a} + \frac{\partial \beta}{\partial y_P} \cdot \frac{\partial y_P}{\partial a} + \frac{\partial \beta}{\partial z_P} \cdot \frac{\partial z_P}{\partial a}$$

where

$$\frac{\partial \beta}{\partial x_P} = -\frac{\partial \beta}{\partial x}, \quad \frac{\partial \beta}{\partial y_P} = -\frac{\partial \beta}{\partial y}, \quad \frac{\partial \beta}{\partial z_P} = -\frac{\partial \beta}{\partial z}$$

and partials of $x_P$, $y_P$, and $z_P$ with respect to semi-major axis are summarized in the subroutine TARPRU analysis.

Partials of $\beta$ with respect to $\dot{x}_P$, $\dot{y}_P$, and $\dot{z}_P$ do not appear in the above expression since they are all zero. Partial of $\beta$ with respect to the remaining target planet orbital elements are treated similarly.
TRAX Flow Chart

ENTER

Compute Julian date at measurement time $t_k$.

Call $\Omega$B and EPHEM to compute the inertial ecliptic position and velocity coordinates of the earth and the target planet.

$\Omega$BS ≠ 0?

YES

NO


1, 2

I$\Omega$EK = ?

110
50

Compute spacecraft position and velocity coordinates relative to the Earth.

Compute range \( \rho \) and range-rate \( \dot{\rho} \) of spacecraft relative to the Earth.

\[ \text{YES} \]

\[ \text{NO} \]

\[ I_B S = 0 \] ?

\[ I_T R K = 1 \] ?

\[ \text{YES} \]

\[ \text{NO} \]

Set \( Y = \dot{\rho} \) and \( \text{NR} = 1 \).

Set \( Y = [\rho, \dot{\rho}]^T \) and \( \text{NR} = 2 \).

RETURN

Compute partials of \( \dot{\rho} \) with respect to \( \vec{x} \) for the idealized station and insert in \( H \) matrix. Set \( \text{NR} = I_T R K \).

\[ I_T R K = 1 \] ?

\[ \text{YES} \]

RETURN

\[ \text{NO} \]

Compute partials of \( \rho \) with respect to \( \vec{x} \) for the idealized station and insert in \( H \) matrix.

RETURN
Compute nominal altitude, latitude, and longitude of the IA-th tracking station.

YES

Compute actual altitude, latitude, and longitude of the IA-th tracking station.

Compute geocentric equatorial components of tracking station position and velocity relative to the center of the Earth.

Compute geocentric ecliptic components of tracking station position and velocity relative to the Earth.

Compute spacecraft position, velocity, range \( \rho \), and range-rate \( \dot{\rho} \) relative to IA-th tracking station.

YES

150

NO

Compute geocentric equatorial components of tracking station position and velocity relative to the center of the Earth.

Compute geocentric ecliptic components of tracking station position and velocity relative to the Earth.

Compute spacecraft position, velocity, range \( \rho \), and range-rate \( \dot{\rho} \) relative to IA-th tracking station.

YES

NO

ITEK odd?

Set \( Y = \rho \) and \( NR = 1 \).

Set \( Y = [\rho, \dot{\rho}]^T \) and \( NR = 2 \).

RETURN
Compute partials of $\dot{\rho}$ with respect to $\dot{X}$ for the IA-th station and insert in $H$ matrix.

Are any station location errors in $\delta x_s$ or $\delta v$ ?

YES

Call STAPRL to compute partials of station position and velocity components with respect to station radius $R$, latitude $\theta$, and longitude $\phi$.

$K = 1$

$\delta p_k = \begin{cases} \delta R, & k=1 \\ \delta \theta, & k=2 \\ \delta \phi, & k=3 \end{cases}$

A

Is $\delta p_k$ in $\delta x_s$ or $\delta v$ ?

YES

Compute partial of $\dot{\rho}$ with respect to $\delta p_k$ and insert in the appropriate column of the $M$ or $L$ matrix.

YES $K > 3$ ?

NO $K = K + 1$

B

YES $\text{ITEX} > 4$ ?

NO

A

220

150
Is range-rate bias \( \delta \beta \) in \( \delta \hat{x}_b \) or \( \delta \hat{v}_b \)?

YES

Compute partial of \( \rho \) with respect to \( \delta \beta \) and insert in the appropriate column of the \( M \) or \( L \) matrix.

NO

Set \( NR = 1 \)

ITK odd?

YES

RETURN

NO

Set \( NR = 2 \) and shift all previously computed \( \rho \) partials into the 2nd row of the \( H, M, \) and \( L \) matrices.

Compute partials of \( \rho \) with respect to \( \hat{x}_b \) for the \( IA \)-th station and insert in \( H \) matrix.

\[ \delta p_k = \begin{cases} \delta R, & k=1 \\ \delta \theta, & k=2 \\ \delta \phi, & k=3 \end{cases} \]

\( K = 1 \)

Is \( \delta p_k \) in \( \delta \hat{x}_b \) or \( \delta \hat{v}_b \)?

NO

YES

Compute partial of \( \rho \) with respect to \( \delta p_k \) and insert in the appropriate column of the \( M \) or \( L \) matrix.

\( K > 3 \)?

NO

\( K = K + 1 \)

YES

300

C

C
Is range bias $\delta \rho$ in $\delta \xi$ or $\delta \nu$?

---

Define do loop parameters for three simultaneous star-planet angle measurements:
$NA=1$, $NC=3$, $NR=3$}

Define do loop parameters for the appropriate single star-planet angle measurement:
$NA=ITK-10$, $NC=NA$, $NR=1$}

Compute target planet position and velocity coordinates and range relative to the spacecraft.
Compute cosine and sine of I-th star-planet angle $\alpha_i$.

- **YES** $I_{\text{POS}} = 0$?
  - Compute partials of $\alpha_i$ with respect to $x$ and insert in $Y$ matrix.
  - Is bias $\delta \alpha$ in $\delta X$ or $\delta \nu$?
    - **YES** Compute partial of $\alpha_i$ with respect to $\delta \alpha$ and insert in the appropriate column of the $M$ or $L$ matrix.
    - **NO** $I = NC$?
      - **YES** RETURN
      - **NO** $I = I + 1$
      - $I > NC$?
        - **YES** $K = L$
        - $F$
      - **NO** $E$
Call TARFRL to compute target planet position partials with respect to element $\delta p_k$.

Compute partial of $\alpha_i$ with respect to $\delta p_k$ for all pertinent star-planet angles, and insert in the appropriate column of the $M$ or $G$ matrix.

Is $\delta p_k$ in $\delta \vec{x}$ or $\delta \vec{u}$? YES

$\delta p_k = \begin{cases} \delta a, & K=1 \\ \delta e, & K=2 \\ \delta i, & K=3 \\ \delta \Omega, & K=4 \\ \delta \omega, & K=5 \\ \delta M, & K=6 \end{cases}$

$K > \beta \? NO \ P \ K = K + 1 \ YES \ RETURN$

520 Compute target planet position and velocity coordinates and range relative to spacecraft. Compute target planet radius.

540 $\text{IFR} E = 0 \? NO \ P \ YES$

Set $Y = \beta$, the apparent planet diameter. Set $\beta = 1$.

RETURN
Compute partials of $\beta$ with respect to $X$ and insert in $H$ matrix. Set $NR = 1$.

Is bias $\delta \beta$ in $\delta X$ or $\delta \nu$?

YES

Compute partial of $\beta$ with respect to $\delta \beta$ and insert in the appropriate column of the $M$ or $L$ matrix.

$K = 1$

$\delta p_k = \{ \delta a, K=1 \}
\delta e, K=2
\delta i, K=3
\delta \Omega, K=4
\delta \omega, K=5
\delta \kappa, K=6$

NO

Is $\delta p_k$ in $\delta x$ or $\delta u$?

YES

Call TARFRL to compute target planet position partials with respect to element $\delta p_k$.

Compute partial of $\beta$ with respect to $\delta p_k$ and insert in the appropriate column of the $M$ or $G$ matrix.

NO

$K > 6$?

YES

$K = K + 1$

RETURN

G
SUBROUTINE TRANS

PURPOSE: TO PERFORM ONE OF THE FOLLOWING THREE OPTIONS.
1. CONVERT FROM GEOCENTRIC EQUATORIAL RECTANGULAR COORDINATES TO GEOCENTRIC ECLIPTIC COORDINATES
2. CONVERT FROM GEOCENTRIC EQUATORIAL COORDINATES TO HELIOCENTRIC ECLIPTIC COORDINATES
3. CONVERT FROM GEOCENTRIC ECLIPTIC COORDINATES TO HELIOCENTRIC ECLIPTIC COORDINATES

CALLING SEQUENCE: CALL TRANS(ICODE, X,Y,Z,VX,VY,VZ,XE,YE,ZE,VXE, VYE,VZE,EPS,ICODE2)

ARGUMENTS:
EPS I OBliquity of Earth
ICODE I AN INTERNAL CODE THAT DETERMINES IF OPTION 1 OR 2 ABOVE WILL BE EXERCISED
ICODE2 I AN INTERNAL CODE THAT DETERMINES IF OPTION 3 ABOVE IS TO BE EXERCISED
VX I/O X-VELOCITY COMPONENT OF THE VEHICLE
VXE I X-VELOCITY COMPONENT OF EARTH IN HELIOCENTRIC ECLIPTIC COORDINATES
VY I/O Y-VELOCITY COMPONENT OF THE VEHICLE
VYE I Y-VELOCITY COMPONENT OF EARTH
VZ I/O Z-VELOCITY COMPONENT OF THE VEHICLE
VZE I Z-VELOCITY COMPONENT OF EARTH
X I/O X-POSITION COMPONENT OF THE VEHICLE
XE I X-POSITION COMPONENT OF THE EARTH IN HELIOCENTRIC ECLIPTIC COORDINATES
Y I/O Y-POSITION COMPONENT OF THE VEHICLE
YE I Y-POSITION COMPONENT OF THE EARTH
Z I/O Z-POSITION COMPONENT OF THE VEHICLE
ZE I Z-POSITION COMPONENT OF THE EARTH

SUBROUTINES SUPPORTED: DATA DATAS

LOCAL SYMBOLS: GE COSine of Obliquity of Earth
DUM INTERMEDIATE VARIABLE
SE SINE OF OBLIQUITY OF EARTH
Subroutine TRANS transforms the position and velocity components of the spacecraft from one coordinate system to another. The three options available with this subroutine are summarized below.

1) Convert from geocentric equatorial coordinates to geocentric ecliptic coordinates using the following equations:

\[
\begin{align*}
X &= X \\
Y &= Y \cos \epsilon + Z \sin \epsilon \\
Z &= -Y \sin \epsilon + Z \cos \epsilon \\
\end{align*}
\]

\[
\begin{align*}
\dot{X} &= \dot{X} \\
\dot{Y} &= Y \cos \epsilon + Z \sin \epsilon \\
\dot{Z} &= -Y \sin \epsilon + Z \cos \epsilon \\
\end{align*}
\]

2) Convert from geocentric equatorial coordinates to heliocentric ecliptic coordinates. The same procedure as above is used to convert from geocentric equatorial to geocentric ecliptic. Then translate according to the following equations:

\[
\begin{align*}
X &= X + X_E \\
Y &= Y + Y_E \\
Z &= Z + Z_E \\
\end{align*}
\]

3) Convert from geocentric ecliptic coordinates to heliocentric ecliptic coordinates using the following equations:

\[
\begin{align*}
X &= X + X_E \\
Y &= Y + Y_E \\
Z &= Z + Z_E \\
\end{align*}
\]
SUBROUTINE TRAPAR

PURPOSE: TO COMPUTE THE FOLLOWING SET OF NAVIGATION PARAMETERS -- FLIGHT PATH ANGLE, ANGLE BETWEEN RELATIVE VELOCITY AND PLANE OF THE SKY, GEOCENTRIC DECLINATION, EARTH/SPACECRAFT/TARGET PLANET ANGLE, ANTENNA AXIS/LIMB OF SUN ANGLE, AND SPACECRAFT OCCULTATION RATIOS FOR SUN, MOON, AND PLANETS.

CALLING SEQUENCE: CALL TRAPAR

SUBROUTINES SUPPORTED: PRINT PRINT4 SETVEVS PRINT3 SETEVN

SUBROUTINES REQUIRED: EPHEM ORB PECEQ

LOCAL SYMBOLS: ALFA VECTOR FORMING RIGHT-HANDED ORTHOGONAL TRIAD WITH XN AND SSS VECTORS FOR CALCULATION OF ANTENNA AXIS/LIMB OF ANGLE OF SUN

AMAG MAGNITUDE OF THE ALFA VECTOR

BETA ANTENNA AXIS/EARTH ANGLE

CD COSINE OF GEOCENTRIC DECLINATION

CT INTERMEDIATE VARIABLE FOR ALL CALCULATIONS

CZAE COSINE OF EARTH/SPACECRAFT/TARGET PLANE ANGLE

DELTA GEOCENTRIC DECLINATION

D0 INTERMEDIATE VARIABLE FOR CALCULATION OF OCCULTATION RATIOS

ECEQP TRANSFORMATION FROM EARTH ECLIPTIC TO EQUATORIAL FRAME FOR CALCULATION OF GEOCENTRIC DECLINATION

GAMMA INERTIAL FLIGHT PATH ANGLE

IND LOCATION IN THE F ARRAY OF THE EARTH POSITION AND VELOCITY IN THE INERTIAL FRAME

ISAVE SAVES AND RESTORES FIRST ELEMENT OF THE NO-ARRAY FOR BARYCENTRIC NAVIGATION

JND LOCATION IN THE F ARRAY OF THE TARGET PLANET POSITION AND VELOCITY IN THE INERTIAL FRAME
NINETY CONSTANT VALUE, EQUAL TO 90.000
OCCULT OCCULTATION RATIO OF THE I-TH PLANET
PHI ANTENNA AXIS/LIMB OF SUN ANGLE
RDV INTERMEDIATE VARIABLE, DOT PRODUCT OF TWO VECTORS
REMAG MAGNITUDE OF THE EARTH HELIOCENTRIC POSITION
RIMAG MAGNITUDE OF THE POSITION OF THE I-TH PLANET IN THE GEOCENTRIC ECLIPTIC FRAME
RMAG MAGNITUDE OF THE SPACECRAFT HELIOCENTRIC POSITION
RSS SPACECRAFT HELIOCENTRIC POSITION
SD SINE OF GEOCENTRIC DECLINATION
SKYI ANGLE BETWEEN SPACECRAFT VELOCITY RELATIVE TO EARTH AND PLANE OF THE SKY
SRDV DOT PRODUCT OF SPACECRAFT GEOCENTRIC POSITION AND VELOCITY VECTORS
SRE SPACECRAFT GEOCENTRIC ECLIPTIC POSITION AND VELOCITY
SRMAG MAGNITUDE OF SPACECRAFT GEOCENTRIC POSITION
SRQ SPACECRAFT GEOCENTRIC EQUATORIAL POSITION
SRTMAG MAGNITUDE OF SRTP VECTOR
SRTP SPACECRAFT ECLIPTIC POSITION RELATIVE TO TARGET PLANET
SVMAG MAGNITUDE OF SPACECRAFT GEOCENTRIC VELOCITY
SX INTERMEDIATE VARIABLE FOR ALL CALCULATIONS
SZAE SINE OF EARTH/SPACECRAFT/TARGET PLANET ANGLE
THETA INTERMEDIATE ANGLE USED TO CALCULATE NAVIGATION PARAMETERS
V MAG MAGNITUDE OF SPACECRAFT VELOCITY RELATIVE
TO INERTIAL FRAME

Xmag  MAGNITUDE OF THE XN VECTOR BEFORE
     UNITIZING

Xn    CROSS PRODUCT OF SPACECRAFT GEOCENTRIC
     POSITION AND SPACECRAFT SPIN AXIS

Zae   EARTH/SPACECRAFT/TARGET PLANET ANGLE

COMMON COMPUTED/USED:
B     NO     RE

COMMON USED?:
F     IBARY   NBOD   NB   NTP
ONE    PLANET  RADIUS  RAD   SSS
TWO     V   XP     ZERO
TRAPAR Analysis

The coordinate systems and variables required for the derivation of the first four navigation parameters are shown in Figure 1. The inertial coordinate system XYZ may be heliocentric or barycentric ecliptic. The position and velocity of the earth in inertial space is given by \( \mathbf{r}_E \) and \( \mathbf{v}_E \); that of the spacecraft, by \( \mathbf{r} \) and \( \mathbf{v} \); and that of the target planet (or moon), by \( \mathbf{r}_{TP} \) and \( \mathbf{v}_{TP} \). The xyz coordinate system is geocentric equatorial.

1. Flight path angle, \( \theta \).

Let \( \theta \) denote the angle between \( \mathbf{r} \) and \( \mathbf{v} \), so that

\[
\cos \theta = \frac{\mathbf{r} \cdot \mathbf{v}}{r \cdot v} \quad \text{and} \quad \sin \theta = \sqrt{1 - \cos^2 \theta}.
\]

Then

\[
\gamma = \frac{\pi}{2} - \theta.
\]

2. Angle between relative velocity and plane of the sky, \( \iota' \).

The plane of the sky is defined as the plane perpendicular to the vector \( \mathbf{r} - \mathbf{r}_E \). Let \( \theta' \) denote the angle between \( \mathbf{r} - \mathbf{r}_E \) and \( \mathbf{v} - \mathbf{v}_E \), so that

\[
\cos \theta' = \frac{(\mathbf{r} - \mathbf{r}_E) \cdot (\mathbf{v} - \mathbf{v}_E)}{|\mathbf{r} - \mathbf{r}_E| \cdot |\mathbf{v} - \mathbf{v}_E|} \quad \text{and} \quad \sin \theta' = \sqrt{1 - \cos^2 \theta'}.
\]

Then

\[
\iota' = \frac{\pi}{2} - \theta'.
\]

Note that \( \iota' \) is not defined if the relative velocity \( \mathbf{v} - \mathbf{v}_E \) is zero.

3. Geocentric declination, \( \delta \).

Let \( (x, y, z) \) denote the geocentric equatorial components of \( \mathbf{r} - \mathbf{r}_E \).

Then

\[
\delta = \tan^{-1} \left( \frac{z}{\sqrt{x^2 + y^2}} \right).
\]
4. Earth/spacecraft/target planet angle, $\xi$.

The angle $\xi$ is the angle between the vectors $\vec{r} - \vec{r}_E$ and $\vec{r} - \vec{r}_{TP}$, so that

$$\cos \xi = \frac{(\vec{r} - \vec{r}_E) \cdot (\vec{r} - \vec{r}_{TP})}{|\vec{r} - \vec{r}_E| |\vec{r} - \vec{r}_{TP}|}$$

and

$$\sin \xi = \pm \left[ 1 - \cos^2 \xi \right]^{\frac{1}{2}}.$$

The next two navigation parameters relate to the spacecraft antenna axis. The pertinent geometry is shown in Figure 2. The antenna axis $\vec{a}$ is defined as the intersection between the antenna plane (the plane perpendicular to the spacecraft spin axis $\vec{\omega}$) and the plane formed by the $\vec{r} - \vec{r}_E$ and $\vec{a}$ vectors. The vector $\vec{\rho}$ originates from the limb of the sun and lies in the $\vec{r}, \vec{a}$ plane.

5. Antenna axis/Earth angle, $\phi$.

Let $\psi$ denote the angle between the unit spin axis vector $\vec{\omega}$ and $\vec{r} - \vec{r}_E$, so that

$$\cos \psi = \frac{\vec{\omega} \cdot (\vec{r} - \vec{r}_E)}{|\vec{r} - \vec{r}_E|} \quad \text{and} \quad \sin \psi = \pm \left[ 1 - \cos^2 \psi \right]^{\frac{1}{2}}.$$

Then

$$\beta = \frac{\pi}{2} - \psi.$$

Note that the antenna axis is not uniquely defined when the angle $\psi = 0$. 

Figure 2. Antenna Axis Geometry

The axis/limb of Sun angle, $\vartheta$. The unit vector $\vec{n}$ normal to the $\vec{s}, \vec{r} - \vec{r}_E$ plane is given by

$$\vec{n} = \frac{(\vec{r} - \vec{r}_E) \times \vec{s}}{|(\vec{r} - \vec{r}_E) \times \vec{s}|}.$$ 

The unit antenna axis vector $\vec{\alpha}$ is given by

$$\vec{\alpha} = \vec{n} \times \vec{s}.$$
The angle $\theta_2$ denotes the angle between the vectors $\vec{r}$ and $\vec{a}$, so that

$$\cos \theta_2 = -\frac{\vec{r} \cdot \vec{a}}{r} \quad \text{and} \quad \sin \theta_2 = \sqrt{1 - \cos^2 \theta_2}.$$ 

The angle $\theta_1$ denotes the angle between the vectors $\vec{p}$ and $\vec{r}$, so that

$$\theta_1 = \sin^{-1} \left( \frac{R_s}{r} \right), \quad 0 \leq \theta_1 \leq \frac{\pi}{2}$$

where $R_s$ is the radius of the Sun.

Then

$$\vartheta = \theta_2 - \theta_1.$$ 

The final set of navigation parameters relate to spacecraft occultation ratios for the Sun and all other celestial bodies assumed in the dynamic model. The pertinent geometry is shown in Figure 3. The position of the $i$-th celestial body relative to the Sun is denoted by $\vec{r}_i$. Occultation parameters $d_s$ and $d_i$ are defined as the minimal distances from the centers of the Sun and $i$-th body, respectively, to the Earth/spacecraft vector $\vec{r} - \vec{r}_E$.

7. **Spacecraft occultation ratio for the Sun.**

The occultation ratio for the Sun is defined as $d_s/R_s$, where $R_s$ is the Sun radius. As long as the occultation ratio is greater than one, the spacecraft is neither being occulted by the Sun nor passing in front of the Sun. The occultation ratio is computed only when the angle between the $\vec{r} - \vec{r}_E$ and $\vec{r}_E$ vectors is less than or equal to 90 degrees, or, equivalently, when

$$\vec{r}_E \cdot (\vec{r} - \vec{r}_E) \leq 0.$$ 

If this condition is satisfied, the occultation ratio is computed using the equations

$$d_s = \sqrt{\frac{r_E^2 - b^2}{r}}.$$
Occultation occurs if $\frac{d_s}{R_s} \leq 1$ and $|\vec{r} - \vec{r}_E| \geq r_E$; if $\frac{d_i}{R_i} \leq 1$ and $|\vec{r} - \vec{r}_E| < r_E$, then the spacecraft is passing in front of the Sun.

8. Spacecraft occultation ratios for other celestial bodies.

The occultation ratio for the $i$-th celestial body is defined as $\frac{d_i}{R_i}$, where $R_i$ is the radius of the $i$-th body. The occultation ratio is
computed only when
\[(\vec{r} - \vec{r}_E) \cdot (\vec{r}_i - \vec{r}_E) \geq 0\,.

If this condition is satisfied, the occultation ratio is computed using the equations
\[d_i = \left[ a_1^2 - b_1^2 \right]^\frac{1}{2}\]
\[a_1 = |\vec{r}_i - \vec{r}_E|\]
and
\[b_1 = \frac{|\vec{r} - \vec{r}_E| \cdot |\vec{r}_i - \vec{r}_E|}{|\vec{r} - \vec{r}_E|} \,.

Occultation occurs if \(\frac{d_i}{R_i} \leq 1\) and \(|\vec{r} - \vec{r}_E| \geq |\vec{r}_i - \vec{r}_E|\); if \(\frac{d_i}{R_i} \leq 1\) and \(|\vec{r} - \vec{r}_E| < |\vec{r}_i - \vec{r}_E|\), then the spacecraft is passing in front of the \(i\)-th celestial body.
Compute the following quantities:
1. Flight path angle, $\gamma$.
2. Angle between relative velocity and plane of the sky, $i$ (only if $|\vec{V} - \vec{V}_E| > 1 \times 10^{-7}$ km/sec)
3. Geocentric declination, $\delta$.
4. Earth/spacecraft/target planet angle, $\xi$.
5. Antenna axis/Earth angle, $\beta$ (only if $\beta > 0.1$ deg).

NO

IBARY = 1?

A

Compute the following quantities:
1. Antenna axis/limb of Sun angle, $\phi$ (only if antenna axis defined).
2. Occultation ratio.
3. Occultation ratios of celestial objects.

RETURN

YES

CALL $\phi$RB and EPHEM. Return position of Earth relative to Sun in XP array.

Compute position of spacecraft relative to Sun.

A
SUBROUTINE TRJTRY

PURPOSE: TO DETERMINE THE TIME OF THE NEXT GUIDANCE EVENT AND INTEGRATE THE NOMINAL TRAJECTORY FROM THE PREVIOUS EVENT TIME TO THE NEXT TIME.

CALLING SEQUENCE: CALL TRJTRY

SUBROUTINES SUPPORTED: NOMINAL

SUBROUTINES REQUIRED: VMP

LOCAL SYMBOLS:

ACK: ACCURACY USED TO INTEGRATE THE NOMINAL TRAJECTORY

DELMIN: TIME (DAYS) BETWEEN THE LAST EVENT AND THE NEXT EVENT

DELTMIN: SAME AS DELMIN - THE TIME VMP IS TO INTEGRATE THE TRAJECTORY UNLESS ANOTHER STOPPING CONDITION OCCURS

DELT: SAME AS DELMIN

ERROR: MINIMUM ALLOWABLE VALUE OF DELMIN

ISP2: FLAG TO CONTROL STOPPING CONDITION

=1 STOP AT SPHERE-OF-INFLUENCE

=0 DO NOT STOP AT SPHERE-OF-INFLUENCE

I: INDEX

J: INDEX

RSF: SPACECRAFT STATE AT FINAL TIME

COMMON COMPUTED/USED:

DO IC1 ICL ICL ISPH KSICA

KTIM RIN TIMG TRIM

COMMON COMPUTED:

DELPTE IEPHEM IMPR IPRINT KUR

COMMON USED:

ACKT KGVD NCPR NOGYD TMPY

V
TRJTRY Analysis

TRJTRY determines the time of the next guidance event and integrates the nominal trajectory from the previous event time to the next time.

Special provisions must be made in determining the next guidance event because of the flexibility permitted in specifying the time of those guidance events. For every guidance event $i$, parameters $KTM(i)$ and $TDM(i)$ will have been set before entering TRJTRY. $KTM(i)$ specifies the epoch to which the guidance event $i$ is referenced with $KTM(i) = 1, 2, 3$ corresponding to epochs of initial time, sphere of influence (SOI) intersection, and closest approach (CA) passage respectively. $TDM(i)$ then specifies the time interval (days) from the epoch to the guidance event. The guidance events do not need to be arranged chronologically. After execution of each guidance event, the flag $KTM(i)$ is set equal to 0.

The first computational procedure in TRJTRY is the sequencing loop. Here a search determines the minimum value of $TDM(i)$ over all values of $i$ such that $KTM(i) = 1$. The time interval $\Delta t$ between that time and the current time is then computed. If $\Delta t$ is less than an allowable tolerance ($=10^{-3}$ days) the program returns to NOMINAL for the processing of the current event.

If $\Delta t \geq \Delta t$ TRJTRY must perform an integration to the next guidance event. TRJTRY first sets up flags controlling integration stopping conditions depending upon the current value of KSICA. The flag KSICA determines the current phase of the trajectory. KSICA is initially set equal to 1 (PRELIM). When the target planet SOI is encountered KSICA is set to 2. Finally when CA to the target planet occurs it is set to 3.

The stopping condition flags are ISP2 and ICL2. The flag ISP2 determines whether the integration should be stopped at SOI if encountered (ISP2 = 1) or not (ISP2 = 0). The flag ICL2 determines whether the integration should be stopped at CA if encountered (ICL2 = 1) or not (ICL2 = 0).

Therefore if $KSICA = 1$, TRJTRY sets ISP2 = 1 so that the integration will stop at the guidance event time only if that time occurs before SOI. But if the SOI is encountered before the event time, all times referenced to the SOI must be updated before determining the next event. Similarly when $KSICA = 2$ TRJTRY sets ICL2 = 1 so that times referenced to CA may be updated when CA occurs. Of course when $KSICA = 3$, all times have been updated (referenced to initial time) and neither ISP2 nor ICL2 need be set to 1.

Having set the stopping condition flags, TRJTRY now calls KNP for the propagation of the trajectory to the required stopping condition. At the end of the integration it records the current trajectory time and state.

TRJTRY now sorts again on KSICA. If $KSICA = 3$, the trajectory has been integrated to the time of the current event and so control may be returned to NOMINAL.
If KSICA = 1 the SOI had not yet been reached at the previous event. TRJTRY then checks the flag ISPH. The flag ISPH reveals whether the current trajectory intersected the target planet SOI (ISPH = 1) or did not (ISPH = 0). Therefore if ISPH = 0, the current guidance event occurred before the trajectory intersected the SOI and thus the current state corresponds to the time of the guidance event. Therefore the return is made to NORMAL.

If however KSICA = 1 and ISPH = 1 the trajectory integration was stopped at the SOI. TRJTRY now sets KSICA = 2 and updates all times referenced to the SOI so that they are now referenced to initial time (KTIM(i) = 1).

It reenters the sequencing loop to determine the time of the next guidance event where the candidate events now include those originally referenced to SOI.

Similar steps are made when KSICA = 2. The flag ICL designates whether the current trajectory had a CA (ICL = 1) or not (ICL = 0). If KSICA = 2 and ICL = 0, the trajectory encountered the guidance event before reaching a CA so the return is made to NORMAL. If KSICA = 2 and ICL = 1, the final time and state of the trajectory refer to closest approach. In this case TRJTRY sets KSICA = 3 and updates to initial time all times originally referenced to CA. It then returns to the sequencing loop.
TRJTRY View Chart

ENTER

A

Have $t_k = \text{current traj. time}$

Choose KUR as index of min
value of all TIMG(i) such that
KTM(i) = 1. Set

$\Delta t = \text{TIMG(KUR)} - t_k$

$\Delta t \leq$

-1

KSICA = ?

-2

Set ICL2=0
LSP2=1

Set ICL2=1
LSP2=0

Set ICL2=0
LSP2=0

Integrate trajectory to stopping cond
($\Delta t, \text{SOL,CA}$) and set $t_k = t_k + t_a$

RETURN

RETURN

RETURN

RETURN

Set KSICA = 2
For KTM(i)=2 set
TIMG(j)=t_k+TIMG(j)
KTM(j) = 1

Set KSICA = 3
For KTM(j)=3 set
TIMG(j)=t_k+TIMG(j)
KTM(j) = 1

A
SUBROUTINE VARADA

PURPOSE COMPUTE VARIATION MATRIX FOR THREE-VARIABLE B-PLANE
GUIDANCE POLICY IN THE ERROR ANALYSIS PROGRAM

CALLING SEQUENCE CALL VARADA(RI,XSIP,XSIV,TEVN,TSI,ADA,BS,BOTS,BDTS)

ARGUMENTS

ADA O VARIATION MATRIX
BS I B OF THE NOMINAL TRAJECTORY
BDRS I B DOT R OF THE NOMINAL TRAJECTORY
BOTS I B DOT T OF THE NOMINAL TRAJECTORY
RI I POSITION AND VELOCITY OF THE VEHICLE AT THE
   TIME OF GUIDANCE EVENT
TEVN I TRAJECTORY TIME OF THE GUIDANCE EVENT
TSI I TRAJECTORY TIME AT WHICH THE VEHICLE
   REACHED THE SPHERE OF INFLUENCE ON THE
   NOMINAL TRAJECTORY
XSIP I POSITION OF THE VEHICLE AT THE SPHERE OF
   INFLUENCE ON THE NOMINAL TRAJECTORY
XSIV I VELOCITY OF THE VEHICLE AT THE SPHERE OF
   INFLUENCE ON THE NOMINAL TRAJECTORY

SUBROUTINES SUPPORTED: GUID

SUBROUTINES REQUIRED: NTH

LOCAL SYMBOLS:

BDR1 TEMPORARY STORAGE FOR BDR
BDT1 TEMPORARY STORAGE FOR BDT
81 TEMPORARY STORAGE FOR B
DSI1 TEMPORARY STORAGE FOR DSI
IPR TEMPORARY STORAGE FOR IPRINT
ISP TEMPORARY STORAGE FOR ISP2
IP0 TEMPORARY STORAGE FOR IPRINT
RF ALTERED FINAL STATE OF VEHICLE
TSI1 TEMPORARY STORAGE FOR TSI
### ALTERED INITIAL STATE OF VEHICLE

<table>
<thead>
<tr>
<th>COMMON COMPUTED/USED</th>
<th>BDR</th>
<th>BOT</th>
<th>DSI</th>
<th>IPRINT</th>
<th>ISPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON COMPUTED</td>
<td>B</td>
<td>DELTH</td>
<td>RSI</td>
<td>ITM1</td>
<td>VSI</td>
</tr>
<tr>
<td>COMMON USED</td>
<td>DATEJ</td>
<td>FACP</td>
<td>FACV</td>
<td>FTOL</td>
<td>NTMC</td>
</tr>
</tbody>
</table>
VARADA Analysis

Subroutine VARADA employs numerical differencing to compute the variation matrix $\eta$ for the three-variable $B$-plane guidance policy in the guidance event of the error analysis mode. See subroutine VARSIM Analysis for further analytical details, since the only difference between VARADA and VARSIM is that VARADA computations are based on the most recent targeted nominal, while VARSIM computations are based on the most recent nominal. The VARADA flow chart is identical to that of VARSIM except for the fact that in VARADA the nominal position/velocity state at $t_{SI}$ is saved prior to calling VARADA, while in VARSIM it is saved locally.
SUBROUTINE VARSIM

PURPOSE: COMPUTE VARIATION MATRIX FOR THREE-VARIABLE B-PLANE GUIDANCE POLICY IN THE SIMULATION PROGRAM

CALLING SEQUENCE: CALL VARSIM(RII, TEVN, TSI, ADA)

ARGUMENTS:

RAI I VEHICLE POSITION/VELOCITY ON MOST RECENT NOMINAL TRAJECTORY AT TIME OF THE GUIDANCE EVENT

TEVN I TRAJECTORY TIME OF GUIDANCE EVENT

TSI I TRAJECTORY TIME AT SPHERE OF INFLUENCE

SUBROUTINES SUPPORTED: GLISS

SUBROUTINES REQUIRED: NTMS

LOCAL SYMBOLS:

BDRS TEMPORARY STORAGE FOR BDR
BDTS TEMPORARY STORAGE FOR BDT
BS TEMPORARY STORAGE FOR B
IPR TEMPORARY STORAGE FOR IPRINT
ISP2 TEMPORARY STORAGE FOR ISP2
RF1 ALTERED FINAL STATE OF VEHICLE ON MOST RECENT NOMINAL
RSIS TEMPORARY STORAGE FOR RSI
TSI1 TEMPORARY STORAGE FOR TSI
VSIS TEMPORARY STORAGE FOR VSI
XC ALTERED INITIAL STATE OF VEHICLE ON MOST RECENT NOMINAL

COMMON PREPARED/USED:

BDR BDT B DSI IPRINT
ISP1 ISP2 RSI VS1

COMMON CREATED:

TRTM1

COMMON DELETED:

DATEJ FACP FACV NTMC NTP
VARSIM-1

VARSIM Analysis

Subroutine VARSIM employs numerical differencing to compute the variation matrix $\eta$ for the three-variable B-plane guidance policy in the guidance event of the simulation mode. This variation matrix relates deviations in the position/velocity state at $t_k$ to deviations in $B$-$T$, $B$-$R$, and $t_{SI}$:

$$
\begin{bmatrix}
\delta B\cdot T \\
\delta B\cdot R \\
\delta t_{SI}
\end{bmatrix} = \eta
\begin{bmatrix}
\delta \vec{R}_k \\
\delta \vec{V}_k
\end{bmatrix} = \eta \vec{\delta x}_k
$$

Since no good analytical formulas which relate $\delta t_{SI}$ to $\delta \vec{R}_k$ and $\delta \vec{V}_k$ exist, numerical differencing must be employed to compute $\eta$.

Let $\tilde{\eta}_j$ be the j-th column of the matrix $\eta$, and assume (most recent) nominal $B$-$T^*$, $B$-$R^*$, $t_{SI}^*$, and $\vec{x}_k^*$ are available. To obtain $\tilde{\eta}_j$ we increment the j-th element of $\vec{x}_k^*$ by the numerical differencing factor $\Delta x_j$ and numerically integrate the spacecraft equations of motion from $t_k$ to the sphere of influence of the target planet to obtain the new values of $B$-$T$, $B$-$R$, and $t_{SI}$. Then

$$
\tilde{\eta}_j = \begin{bmatrix}
\frac{B\cdot T - B\cdot T^*}{\Delta x_j} \\
\frac{B\cdot R - B\cdot R^*}{\Delta x_j} \\
\frac{t_{SI} - t_{SI}^*}{\Delta x_j}
\end{bmatrix}
$$

\[ j = 1, 2, \ldots, 6 \]
Compute the $N$-th column of the variation matrix $\eta$.

$N = N + 1$

$N \leq 6$?

YES

A

NO

C

Restore $B \cdot T$, $3 \cdot R$, $B$, $\vec{E}_{S1}$, and $\vec{V}_{S1}$ for the most recent nominal trajectory.

Restore IPRINT and ISPZ flags.

RETURN
SUBROUTINE VECTOR

PURPOSE: TO COMPUTE THE VECTOR ORBITAL ELEMENTS K (ANGULAR MOMENTUM VECTOR), E (ECCENTRICITY VECTOR TOWARD PERIHELION), TO COMPUTE THE SPACECRAFT FINAL POSITION ON THE ORBIT TO ACCURATELY APPROXIMATE THE DESIRED TIME INTERVAL, AND TO COMPUTE THE CONIC SECTION TIME OF FLIGHT.

CALLING SEQUENCE: CALL VECTOR

SUBROUTINES SUPPORTED: VMP

SUBROUTINES REQUIRED: SPACE

COMMON COMPUTED/USED: V

COMMON COMPUTED: KOUNT

COMMON USED: HALF ITRAT ONE PI THREE TWOPI TWO

VECTOR-A
VECTOR Analysis

The Kepler vector $\vec{k}$ representing twice the areal rate of the spacecraft with respect to the virtual mass to be used during the current interval is computed from

$$\vec{k} = \vec{r}_{VS_B} \times \vec{v}_{VS_B}$$

(1)

where the position and velocity vectors are referenced to the virtual mass at the beginning of the interval. The eccentricity vector for the interval is given by

$$\vec{e} = -\frac{\vec{r}_{VS_B}}{r_{VS_B}} - \frac{\vec{k} \times \vec{r}_{VS_B}}{\mu_V}$$

(2)

where $\mu_V$ is the average value of the virtual mass during the interval.

The current time interval is computed from

$$\Delta T = \Delta t + k \Delta t^2$$

(3)

where the factor $k$ was precomputed during the previous iterations. The direction of the final position $\vec{\sigma}$ is determined from

$$\vec{\sigma} = \vec{r}_{VS_B} + \Delta T \vec{v}_{VS_B}$$

(4)

The magnitude factor $B$ is chosen to force the final position to satisfy the orbit equation ($\vec{e} \cdot \vec{r} = -r + \frac{k^2}{\mu}$)

$$B = \frac{k^2/\mu_V}{\vec{e} \cdot \vec{\sigma} + |\vec{\sigma}|}$$

(5)

The position and velocity vectors of the spacecraft relative to the virtual mass at the end of the interval are then

$$\vec{r}_{VS_E} = B \vec{\sigma}$$

$$\vec{v}_{VS_E} = \frac{\mu_V}{k^2} \left[ \vec{k} \times \left( \vec{e} + \frac{\vec{r}_{VS_E}}{\vec{r}_{VS_E}} \right) \right]$$

(6)
The final position and velocity of the spacecraft in the reference initial coordinates are computed from

$$\mathbf{r}_S = \mathbf{r}_{VS} + \mathbf{v}_E$$
$$\mathbf{v}_S = \mathbf{v}_{VS} + \mathbf{a}_E$$  \hspace{1cm} (7)

The exact conic section time of flight is now computed. The projection of the major axis is

$$\mathbf{n} = \frac{\mathbf{k} \times \mathbf{e}}{\mu_v} \quad e \neq 0$$
$$\mathbf{k} \times \mathbf{r}_{VS} \quad \frac{\mathbf{k} \times \mathbf{r}_{VS}}{\mu_v} \quad e = 0$$  \hspace{1cm} (8)

The length of the semi-major axis is given by

$$a = \frac{k^2}{\mu_v |1-e^2|^\frac{1}{2}} \quad e \neq 1$$
$$a_1 = \frac{2}{r_{VS_1} - \frac{k^2}{\mu_v}} \quad e = 1, i = B, E$$  \hspace{1cm} (9)

The projection of the radius vector orthogonal to the major axis is given by

$$\mathbf{x}_i = \frac{\mathbf{n} \cdot \mathbf{r}_{VS}}{a_i} \quad i = B, E$$  \hspace{1cm} (10)

The mean angular rate is

$$\omega = \frac{\mu_v (1-e^2)}{ka} \quad e \neq 1$$
$$= \frac{k}{2} \quad e = 1$$
where \( \omega < 0 \) for hyperbolic orbits. The eccentric anomaly is given by

\[
E_i = \sin^{-1} x_i \quad e < 1 \\
i = B, E
\]

\[
E_i = \frac{k^2/\mu}{V} x_i \quad e = 1
\]

\[
E_i = \sinh^{-1} x_i \quad e > 1
\]

Then

\[
M_i = E_i - e x_i \quad i = B, E
\]

and the actual conic time of flight is

\[
\Delta t = t_E - t_B = \frac{M_E - M_B}{\omega}
\]

The value of the time factor \( K \) to be used in the following interval is then computed

\[
K = \frac{\Delta t - \Delta t}{(\Delta t)^2}
\]
VECTOR Flow Chart

ENTER

Calculate vector orbital elements $k, e$.

ITRAT = ?

$=3$ 

RETURN

Compute spacecraft final position and velocity.

Compute Keplerian time of flight $\Delta t_k$.

Compute time factor $K$ to be used in next iteration.

RETURN
SUBROUTINE VMASS

PURPOSE: TO COMPUTE THE POSITION, VELOCITY, MAGNITUDE, AND MAGNITUDE RATE OF THE VIRTUAL MASS.

CALLING SEQUENCE: CALL VMASS

SUBROUTINES SUPPORTED: VHP

SUBROUTINES REQUIRED: NONE

COMMON COMPUTED/USED: F, V

COMMON USED: NBODY THREE ZERO
VMASS Analysis

The current virtual mass data is computed by VMASS. The magnitude and position of the virtual mass is given by

\[ \mu_V = r_{VS}^3 M_S \]
\[ \vec{r}_V = \frac{\vec{M}}{M_S} \]

where the intermediate variables are given by

\[ \vec{M} = \sum_{i=1}^{n} \frac{\mu_i \vec{r}_i}{r_{iS}^3} \]
\[ M_S = \sum_{i=1}^{n} \frac{\mu_i}{r_{iS}^3} \]

and of course \( r_{iS} = |\vec{r}_i - \vec{r}_S| \) and \( r_{VS} = |\vec{r}_V - \vec{r}_S| \) where \( \vec{r}_i \) represents the inertial position vector of the i-th body.

The time derivatives of these variables are given by

\[ \dot{\mu}_V = \mu_V \left( \alpha + \frac{\dot{M}_S}{M} \right) \]
\[ \frac{\dot{\vec{r}}}{r_V} = -\frac{\vec{M} - r_V \dot{M}_S}{M_S} \]

\[ \dot{\vec{M}} = \sum_{i=1}^{n} \frac{\mu_i}{r_{iS}^3} \left[ \frac{\dot{\vec{r}}}{r_i} - \alpha \frac{\dot{\vec{r}}}{r_i} \right] \]

\[ \dot{\dot{M}} = -\sum_{i=1}^{n} \frac{\mu_i}{r_{iS}^3} \alpha_{iS} \]
where
\[ \alpha_{IS} = \frac{3 \bar{r}_{IS} \cdot \bar{r}_{IS}}{r_{IS}^2} \quad (9) \]

Finally, the velocity of the spacecraft with respect to the virtual mass is
\[ \frac{\dot{r}^*}{\nu} = \frac{\dot{r}}{\nu} - \frac{\dot{r}^*}{\nu} \quad (10) \]
VMASS Flow Chart

ENTER

Compute relative position vectors of S/C $\mathbf{r}_{IS}$ and virtual mass magnitude $\gamma$

Compute vector $\mathbf{H}$, virtual mass position $\mathbf{r}_V$, spacecraft position relative to virtual mass $\mathbf{r}_{VS}$, and virtual mass magnitude $\gamma$.

Compute velocity of S/C with respect to planets $\mathbf{r}_{IS}$, scalar parameter $\alpha_{IS}$ and auxiliary parameter $\dot{\alpha}_S$.

Compute $\mathbf{H}, \dot{\mathbf{r}}_V, \mathbf{r}_{VS}$ and $\dot{\mu}_V$

RETURN
SUBROUTINE VMP

PURPOSE PROVIDE LOGIC TO GENERATE VIRTUAL MASS TRAJECTORY

CALLING SEQUENCE CALL VMP(RS,ACC,D1,TRTM,DELTM,RSF,ISP2)

ARGUMENTS

RS(6) I INERTIAL POSITION AND VELOCITY OF S/C AT INITIAL TIME

ACC I ACCURACY USED IN INTEGRATION

D1 I JULIAN DATE, EPOCH 1900, OF INITIAL TIME

TRTM I TRAJECTORY TIME (DAYS) AT INITIAL TIME

DELTM I TIME INTERVAL IN DAYS OVER WHICH THE TRAJECTORY IS TO BE PROPAGATED UNLESS A STOPPING CONDITION IS REACHED

RSF(6) O INERTIAL POSITION AND VELOCITY OF S/C AT FINAL TIME

ISP2 I SPHERE OF INFLUENCE STOPPING FLAG

=0 DO NOT STOP AT SOI

=1 STOP AT SOI IF INTERSECTED BEFORE FINAL TIME

SUBROUTINES SUPPORTED: CASCAD NTMS GIDANS TARGET TARMAX

SUBROUTINES REQUIRED: CAREL ELCAR EPHEM IMPACT ORB

PECEQ TIME ESTMT INPUTZ PRINT

LOCAL SYMBOLS

AU NOT USED

CXI COSINE OF THE TRAJECTORY INCLINATION AT CLOSEST APPROACH

D INTERMEDIATE DATE FOR PRINTOUT PURPOSES

DELR INTERMEDIATE VARIABLE FOR INTERSECTION OF SPHERE-OF-INFLUENCE

DELT INTERMEDIATE TIME INCREMENT FOR INTERPOLATED SPHERE-OF-INFLUENCE POSITION

ECEQP TRANSFORMATION FROM ECLIPTIC TO EQUATORIAL SYSTEM FOR TARGET PLANET

ICUT CUTOFF FLAG USED WHEN CLOSEST APPROACH CUTOFF WAS DESIRED BUT NO VALID CLOSEST
APPROACH FOUND

IODAY PRINTOUT CALENDAR DAY

IMH PRINTOUT CALENDAR HOUR

IMO PRINTOUT CALENDAR MONTH

IP NUMBER OF PLANET, USED IN PRINTOUT

ISPMH INDICATOR FOR CALCULATION OF SPECIAL COMPUTING INTERVAL NEAR TARGET PLANET SPHERE-OF-INFLUENCE

IYR PRINTOUT CALENDAR YEAR

JJ COUNTER FOR NUMBER OF ITERATIONS FOR INTERPOLATED SPHERE OF INFLUENCE

LARCA INDICATOR FOR CALCULATION OF PSUEDO CLOSEST APPROACH

MIN PRINTOUT CALENDAR MINUTES

NSPI INDEX OF THE SPACECRAFT VECTORS IN THE F-ARRAY WITH RESPECT TO THE TARGET PLANET

RCM MAGNITUDE OF POSITION OF VEHICLE RELATIVE TARGET PLANET AT CLOSEST APPROACH

RCM1 PREVIOUS RADIUS OF VEHICLE RELATIVE TO TARGET PLANET

RCM2 PRESENT RADIUS OF VEHICLE RELATIVE TO TARGET PLANET

RDT NOT USED

RTEMP SPACECRAFT POSITION AT INTERPOLATED CLOSEST APPROACH IN THE TARGET PLANET EQUATORIAL SYSTEM

SEC PRINTOUT CALENDAR SECONDS

TIMCR TIME INCREMENT USED FOR INTERPOLATED CLOSEST APPROACH

TMIN TOTAL TIME USED IN ONE INTEGRATED TRAJECTORY

TIM1 CLOCK TIME AT BEGINNING OF TRAJECTORY

TIM2 CLOCK TIME AT END OF TRAJECTORY
CRLVI
TATIONAL
CONSTANT
OF
TARGET
PLANET
(KM**3/SEC**2)

TP
INTERMEDIATE
VARIABLE
FOR
CALCULATION
OF
SPECIAL
COMPUTING
INTERVAL
NEAR
SPHERE-OF-INFLUENCE
OF
TARGET
PLANET

ITG
GRAVITATIONAL
CONSTANT
OF
TARGET
PLANET
(KM**3/SEC**2)

VCA
VELOCITY
MAGNITUDE
WITH
RESPECT
to
TARGET
PLANET
AT
INTERPOLATED
CLOSEST
APPROACH

VCH
MAGNITUDE
OF
VELOCITY
OF
VEHICLE
RELATIVE
TO
TARGET
PLANET
AT
CLOSEST
APPROACH
BEFORE
INTERPOLATION

VQ
EQUATORIAL
SPACECRAFT
VELOCITY
RELATIVE
to
TARGET
PLANET
AT
CLOSEST
APPROACH
BEFORE
INTERPOLATION

VTEMP
SPACECRAFT
VELOCITY
AT
INTERPOLATED
CLOSEST
APPROACH
IN
THE
TARGET
PLANET
EQUATORIAL
SYSTEM

XI
UNINTERPOLATED
EQUATORIAL
INCLINATION
FOR
PRINTOUT
PURPOSES

XMAG
INTERMEDIATE
VARIABLE
FOR
CALCULATION
OF
XI

XN
VECTOR
NORMAL
to
TRAJECTORY
PLANE
IN
TARGET
PLANET
EQUATORIAL
SYSTEM
FOR
CALCULATION
OF
XI

XQ
EQUATORIAL
SPACECRAFT
POSITION
RELATIVE
TO
TARGET
PLANET
AT
UNINTERPOLATED
CLOSEST
APPROACH

ZM
INTERMEDIATE
VARIABLE
IN
CALCULATION
OF
INTERPOLATED
CLOSEST
APPROACH
INCLINATION

ZTEMP
VECTOR
NORMAL
to
TRAJECTORY
PLANE
FOR
CALCULATION
OF
INTERPOLATED
CLOSEST
APPROACH
INCLINATION

COMMON
COMPUTED/USED:
CAINC  DC  DSI  ICL  INCMNT
INCMT  ISPM  ITRAT  KOUNT  NBODY1
NO  RCA  RC  RSI  TIMINT
VSI  V

COMMON
COMPUTED:
DELTIM  INCMPR  RE  RTP  RVS

604
<table>
<thead>
<tr>
<th>COMMON USED</th>
<th>ALNCH</th>
<th>BDR</th>
<th>BDT</th>
<th>B</th>
<th>DELTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM7</td>
<td>EMB</td>
<td>F</td>
<td>HALF</td>
<td>ICL2</td>
<td></td>
</tr>
<tr>
<td>IEPHNS</td>
<td>INPR</td>
<td>IPRINT</td>
<td>NBOD</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>NTP</td>
<td>ONE</td>
<td>PLANET</td>
<td>PMASS</td>
<td>RADIUS</td>
<td></td>
</tr>
<tr>
<td>RAD</td>
<td>SPHERE</td>
<td>TM</td>
<td>TWO</td>
<td>ZERO</td>
<td></td>
</tr>
</tbody>
</table>
VMF Analysis

VMF provides the logic to integrate an N-body trajectory from an initial spacecraft state \( \mathbf{r}_0, \mathbf{v}_0 \) at time \( t_0 \) to one of the following stopping conditions.

1. Target planet sphere of influence (SOI) is reached (ISP2 \( \neq 0 \)).
2. The closest approach to the target planet has been reached (ICL2 = 1).
3. The preset final trajectory time \( t_f \) has been exceeded.

The integration logic is controlled by ITRAT

ITRAT = 1 First pass through computation cycle (including ephemeris computation).

2 Second pass through computation cycle (excluding ephemeris).

3 Initialization flag.

To start the integration, appropriate variables are initialized (PRINTZ) and ITRAT is set equal to 3. The state of all gravitational bodies at \( t_B \) are found (ORB, EPHEM). The initial virtual mass position \( \mathbf{r}_B, \) velocity \( \mathbf{v}_B, \) magnitude \( \mu_B \) and magnitude rate \( \dot{\mu}_B \) are found by

\[ \text{VMASS. Virtual mass dependent values are then initialized} \]

\[ \mu_{AVE} = \mu_{VE} = \mu_{VB} \]  
\[ \dot{\mathbf{r}}_{AVE} = \dot{\mathbf{r}}_{VB} \]  
\[ \dot{\mathbf{r}}_{VS_E} = \dot{\mathbf{r}}_{VS_B} \]  
\[ (Ar)^2 = 1 \]  
\[ ISPH1 = 0 \]
At this point the standard integration routine is entered by calling \textsc{VECTOR}.

In the standard integration routine, a new increment is initiated by calling \textsc{ESTMT} which:

1. Initializes all appropriate variables at the beginning of the increment (subscript \( B \)) to equal their values at the end of the previous increment.
2. Computes a \( \Delta t \) for the increment based on a modified true anomaly passage.
3. Computes the time at the end of the increment \( t_E \).
4. Estimates the final (subscript \( E \)) position \( \mathbf{r}_V \) and magnitude \( \mu_V^E \) of the virtual mass.

Based on these estimates, the average magnitude and velocity of the virtual mass is computed

\[
\mu_{V_{\text{AVE}}} = \frac{1}{2} \left( \mu_{V_B} + \mu_{V_E} \right) \tag{7}
\]

\[
\mathbf{v}_{V_{\text{AVE}}} = \frac{(\mathbf{r}_{V_E} - \mathbf{r}_{V_B})}{\Delta t} \tag{8}
\]

Subroutine \textsc{VECTOR} then computes the orbit relative to the virtual mass based on these estimates. It also refines the estimate of the spacecraft final state \( (\mathbf{r}_E, \mathbf{v}_E) \). \textsc{ORB} and \textsc{EPHEM} are called to determine the state at \( t_E \) of all gravitational bodies being considered. The virtual mass position \( \mathbf{r}_{V_E} \), velocity \( \mathbf{v}_{V_E} \), magnitude \( \mu_{V_E} \) and magnitude rate \( \mathbf{\dot{\mu}}_{V_E} \) are determined by \textsc{VMASS}.

Using these refined values, the virtual mass average magnitude \( \mu_{V_{\text{AVE}}} \) and velocity \( \mathbf{v}_{V_{\text{AVE}}} \) are recomputed using equations (7) and (8). At this point a second pass is made through \textsc{VECTOR} to compute the spacecraft final state \( (\mathbf{r}_S, \mathbf{v}_S) \) which will be used in all subsequent calculations. \textsc{VMASS} is again called to make a final determination of the virtual mass.

\[\text{VMR-2}\]
position, velocity, magnitude and magnitude rate at the end of the increment.

The virtual mass average accelerations are then computed

\[
\ddot{\mu}_v = \left[ \mu_v - \dot{\mu}_v - \dot{\mu}_v (\Delta t) \right] / (\Delta t)^2
\]

\[
\ddot{\nu}_v = \left[ \nu_v - \dot{\nu}_v - \dot{\nu}_v (\Delta t) \right] / (\Delta t)^2
\]

These values are subsequently used by ESTMT to estimate the final position \( r_v \) and magnitude \( \nu_v \) of the virtual mass for the next increment, \( E \).

Tests are now made to determine whether the target planet SOI has been pierced. If it has the interpolated state at the SOI is found using the following iterative routine

\[
\delta r = r_{SOI} - |\dot{\bar{r}}_{ST}^{(n)}|
\]

(11)

where \( r_{SOI} \) = radius of the target planet SOI

\[
\dot{\bar{r}}_{ST}^{(n)} = \text{position of spacecraft WRT target planet at } n\text{-th iteration}
\]

\[
\delta t^{(n)} = \frac{\dot{\bar{r}}_{ST}^{(n)}}{\bar{r}_{ST}^{(n)} - \bar{v}_{ST}^{(0)}}
\]

(12)

\[
\bar{r}_{ST}^{(n+1)} = \bar{r}_{ST}^{(n)} + \bar{v}_{ST}^{(0)} \delta t^{(n)}
\]

(13)

\[
\bar{t}_{SI} = \bar{t}_{E}^{(0)}
\]

(14)

The interpolated state at the SOI is then used by IMPACT to compute B-T and B-R.

If trajectory data is to be printed at this point, the orbit inclination (assuming a hyperbolic orbit about the planet) is computed by first determining the "Kepler vector"

\[
\mathbf{k} = \bar{r}_{ST} \times \bar{v}_{ST}
\]

(15)
in planetocentric equatorial coordinates. Then

$$\cos i = \frac{k_z}{|\hat{r}|}$$  \hspace{1cm} (16)

where $i$ = orbit inclination

$k_z$ = component of $\hat{r}$ normal to planet equatorial plane

Tests are now made to determine if the spacecraft has reached a closest approach to the target planet. If it has, the interpolated state at closest approach ($r_{CA}, v_{CA}$) is computed by calling CAREL with the spacecraft state just following closest approach. CAREL returns the element of the near planet conic. ELCAR is then called with these conic element and returns the interpolated state at closest approach.

If the spacecraft is not within 10 SOI of the target planet, print out of closest approach data may occur; however, integration continues.

The final tests before starting a new integration increment determines if the maximum trajectory time $t_p$ has been exceeded or a planet has been impacted. If these tests are passed, a new integration cycle is initiated by calling ESTMT.
Compute the state of the gravitational bodies at $t_E$, ($\hat{r}_1, \hat{v}_1$).

Compute virtual mass data

Compute virtual mass average accelerations

Initialization of virtual mass dependent values

Initialization

$\tau_f = \tau_b + \Delta t_M$

$\tau_S = \tau_B$

$\hat{r}_E = \hat{r}_B$

(INPUTZ)

($\text{ORB}, \text{EPHEM}$)

Computes the virtual mass data

Virtual mass average magnitude

Virtual mass average velocity

Virtual mass average acceleration

Virtual mass average mass

$\mu_V = \mu_{V_AVE}$

$\hat{r}_V = \hat{r}_{V_AVE}$

$\hat{v}_V = \hat{v}_{V_AVE}$

$\hat{a}_V = \hat{a}_{V_AVE}$

$\hat{m} = \hat{m}_{V_AVE}$

Initialization of virtual mass dependent values

$\hat{\mu} = \mu_{V_AVE}$

$\hat{r}_V = \hat{r}_{V_AVE}$

$\hat{v}_V = \hat{v}_{V_AVE}$

$\hat{a}_V = \hat{a}_{V_AVE}$

$\hat{m} = \hat{m}_{V_AVE}$

$\mu_V = \mu_{V_AVE}$

$\hat{r}_V = \hat{r}_{V_AVE}$

$\hat{v}_V = \hat{v}_{V_AVE}$

$\hat{a}_V = \hat{a}_{V_AVE}$

$\hat{m} = \hat{m}_{V_AVE}$

ISPHI = 0
Compute vector orbital elements $(\ell, e)_A$ spacecraft final state $(\tilde{r}_v, \tilde{v}_v)$ and conic section time of flight $(\Delta t)$ (VECTOR)

- 1

ITERAT = ?

- 2

B

- 3

D

A

E

H

No

Yes

Inside SOI ?

ISPH1 = ?

= 0

= 1

Entered 1.025 SOI ?

Yes

Reduce C2

ISPH1 = 1

Entered SOI ?

No

Yes

Restore C2 to previous value
Refine position at SOI
Compute B·T, B·R (IMPACT)
Compute orbit inclination

= 0

= 1

IPRINT = ?

Write SOI data

Is integration to stop at SOI

No

Yes

F

I

VMP-6
Passed Closest Approach?

- No
  - Is distance to target planet increasing?
    - Yes
      - Refine position at closest approach (CAVEL, ELCAR)
      - Compute orbit inclination
      - IPRINT = 0
      - IPRINT = 1
    - Write closest approach data
  - No

- Yes
  - Is integration to stop at closest approach?
    - Yes
      - F
    - No
      - Has maximum trajectory time been exceeded?
        - Yes
          - IPRINT = 1
          - Write exceeded maximum time
        - No
          - Has planet been impacted?
            - Yes
              - IPRINT = 1
              - Write planet impacted
            - No
              - Is trajectory data to be printed?
                - Yes
                  - Write trajectory data (PRINT)
                - No
                  - Initialize for next increment
                      - Compute t
                      - Estimate final virtual mass position (ESTMT)

- RETURN
SUBROUTINE ZERIT

PURPOSE: TO COMPUTE THE COMPUTATION OF THE ZERO ITERATE "VALUES OF TIME, POSITION VECTOR, AND VELOCITY VECTOR."

CALLING SEQUENCE: CALL ZERIT

SUBROUTINES SUPPORTED: PRELIM, GIDANS

SUBROUTINES REQUIRED: HELIO, LUNA

COMMON USED: IZERO, LTARG
ZERIT Analysis

ZERIT is the executive subroutine handling the computation of the zero iterate values of time, position vector, and velocity vector.

The flag IZERO controls the operation of ZERIT. If IZERO = 0, no zero iterate computation is needed and so ZERIT is exited. If IZERO < 10, the zero iterate is to be computed for a interplanetary trajectory so HELIO is called before returning. If IZERO ≥ 10, the zero iterate is to be computed for a lunar trajectory so LUNA is called for that computation.

ZERIT Flow Chart
BIBLIOGRAPHY


