#  

PROGRAMMERS MANUAL.
for the

Mission Analysis Evaluation and Space

Trajectory Operations Program

MAESTRO
(NASA-CR-132910) PROGEAMMER'S MANUAL FOR N74-16538 THE VISION ANALYSIS EVALUATION AND SPACE TRAJECTORY OPERATIONS PROGRAM (MAESTRO) (Analytical Mechanics Associates. Inc.) Uncles 485. PHC \$26.25 . CSCI 22C G3/30 29268

486 Contract No. NAS 5-11900

March 1973
Report No. 73-10

Prepared by

Analytical Mechanics Associates
11691 National Boulevard
Los Angeles, California 90034
for

Goddard Space Flight Center
Greenbelt, Maryland


# PROGRAMMER'S MANUAL 

for the

# Mission Analysis Evaluation and Space 

# Trajectory Operations Program <br> MAESTRO 

## Contract No. NAS 5-11900

March 1973

Goddard Spaceflight Center Contracting Officer
Mr. Robert Flick

Technical Officer
Mr. Charles Newman

## Prepared by

David Lutzky

William Bjorkman

James Schafer
Section Title ..... Page
I
Introduction ..... 1
II MAESTRO Structure ..... 2
III Cross Reference Map ..... 9
IV Common Blocks ..... 13
ANKOR ..... 14
AVG ..... 15
CETBL2 ..... 16
CETBL3 ..... 17
CETBL9 ..... 18
CNTRL ..... 19
CONST ..... 21
DUM ..... 22
ELMNT ..... 23
FIELDM ..... 24
GRAVTY ..... 25
INPUT ..... 26
INPUTS ..... 43
INTER ..... 44
INTVAR ..... 45
INTVRX ..... 46
MCCOM ..... 47
MOON ..... 52
OBSIT ..... 53
PERT ..... 54
PIT ..... 55
PLNET ..... 56
SAVE ..... 57
SHAD ..... 58
STATE ..... 59
V Input/Output Units ..... 61
VISubroutine Description64
MAIN ..... 65
ACCEL ..... 66
APROCH ..... 68
ARMPIT ..... 79
ATMO ..... 83
AVEQNS ..... 84
AVERGE ..... 89
AVSTRT ..... 93
BCONIC ..... 97
BELL ..... 107
BIGMAT ..... 114
BLOCK DATA ..... 116
BURND ..... 122
BVE ..... 125
CALEND ..... 132
CLOSE ..... 133
CONTRL ..... 136
COVERT ..... 138
CRASH ..... 140
CROSS ..... 144
DATE ..... 145
DOPLER ..... 146
DOT ..... 152
DRAG ..... 153
DVMAG ..... 155
EQNS ..... 156
FIELD2 ..... 166
FIND ..... 171
FIXATG ..... 173
FOWARD ..... 177
GETTAP ..... 179
GRAV ..... 182
HSDTHR ..... 185
INPUTF ..... 186
INTEG ..... 190
INTERP ..... 195
JET ..... 201
LUNA ..... 212
MATMPY ..... 219
MCBURN ..... 221
MCSET ..... 229
MCVERF ..... 233
MDCORS ..... 237
MINTF ..... 248
MINV ..... 251
MONTE ..... 255
MOTORS ..... 264
MULCON ..... 267
MVTRN ..... 277
M50EPM ..... 278
M50JPM ..... 281
M50LEQ ..... 284
M50MDT ..... 288
NUTAIT ..... 290
NUTATE ..... 293
OBLATE ..... 305
OBLE ..... 313
OBLTY ..... 315
OBSET ..... 317
ORBIT ..... 322
ORIENT ..... 327
OUTPUT ..... 330
OUT1 ..... 333
PLANET ..... 335
POST ..... 337
PRINT ..... 340
PROTO ..... 342
PUTELS ..... 347
QUARTC ..... 349
QUIKIE ..... 352
RANDM1 ..... 356
READE ..... 357
RETDV ..... 359
RETRO ..... 361
RKSEVN ..... 365
ROTAIT ..... 370
ROTATE ..... 371
SADOUT ..... 372
SENSO ..... 373
SETUP2 ..... 376
SHADOW ..... 382
SHORB2 ..... 391
SOL ..... 397
SOLP ..... 399
SPER ..... 402
SPNM ..... 403
SUNMIN ..... 406
TABINT ..... 409
TARGET ..... 411
TIMEC ..... 422
TOBODY ..... 430
TRIM ..... 432
TRIM2 ..... 437
TRMN ..... 447
TUBE1 ..... 452
TWELVE ..... 453
TWOPIT ..... 460
UPDATE ..... 465
VIEW ..... 466
VISIB ..... 471
VNORM ..... 473

## LIST OF FIGURES

Page
2.1.1 Main Control Flow Chart ..... 6
2.2.1 Subroutine Foward Structure ..... 7
2.3.1 Midcourse Guidance Structure ..... 8

## LIST OF TABLES

Page
3. 1 MAIN Control Logic Cross Reference Map ..... 10
3.2 Midcourse Guidance Cross Reference Map ..... 11
3.3 Subroutine FOWARD Cross Reference Map ..... 12
5.1 Input/Output Units ..... 62

| MAIN | Initiates MAESTRO. |
| :---: | :---: |
| ACCEL | Determines spacecraft acceleration. |
| APROCH | Control subroutine for an approach analysis. |
| ARMPIT | Determines post injection trim requirements. |
| ATMO | Determines atmospheric density. |
| AVEQNS | Determines state derivatives when averaging. |
| AVERGE | Determines mean solar and lunar accelerations for use in multi-conic. |
| AVSTRT | Determines initial averaged elements from osculating elements. |
| BCONIC | Solves Lambert's problem. |
| BELL | Propagates midcourse execution errors. |
| BIGMAT | Determines eigenvalues and eigenvectors of a matrix. |
| BLOCKDATA | Initializes common blocks. |
| BURND | Computes burn time from impulsive velocity. |
| BVE | Calculates the impact parameter vector. |
| CALEND | Converts modified julian date to calendar date. |
| CLOSE | Determines central planet. |
| CONTRL | Initialization routine. |
| COVERT | Transforms a covariance matrix from local tangent plane to inertial. |
| CRASH | Determines time of closest approach. |
| CROSS | Vector cross product computation. |
| DATE | Converts calendar date to modified julian date. |
| DOPLER | Determines doppler effect. |
| DOT | Performs vector dot product computation. |


| DRAG | Calculates acceleration due to atmospheric drag. |
| :---: | :---: |
| DVMAG | Determines magnitude of a vector. |
| EQNS | Determines derivatives of the state. |
| FIELD2 | Evaluates the disturbing acceleration due to an oblate central planet. |
| FIND | Retrieves elements from the GTDS 24 -hour hold file. |
| FIXATG | Controls logic for fixed-attitude midcourse scan. |
| FOWARD | Propagates the state forward in time. |
| GETTAP | Reads ephemeris tape. |
| GRAV | Determines disturbing acceleration due to the planets. |
| HSDTHR | Determines thrust/weight characteristics of midcourse motor. |
| INPUTF | Data input and editing routine. |
| INTEG | Sets up arrays for numerical integration. |
| INTERP | Interpolates for the state as a function of time. |
| JET | Midcourse pre-targeting determination. |
| LUNA | Approximate Lunar ephemeris. |
| MATMPY | Generalized matrix multiplication routine, |
| MCBURN | Provides the post-midcourse state. |
| MCSE T | Midcourse initialization. |
| MCVERF | Midcourse verification analysis logic. |
| MDCORS | Midcourse guidance package. |
| MINTF | Controls minimum fuel guidance logic. |
| MINV | Determines minimum post-injection trim velocity for an approach hyperbola. |
| MONTE | Control subroutine for a Monte-Carlo analysis. |
| MOTORS | Determines engine thrust/weight characteristics. |
| MUICON | State propagator using Multi-conic. |


| MVTRN | Matrix multiplication. |
| :---: | :---: |
| MEOEPM | Determines Earth mean of 1950 to Earth prime meridian rotation matrix. |
| M50JPM | Determines Earth mean of 1950 to prime meridian of planet $J$ rotation matrix. |
| M50LEQ | Determines Earth mean of 1950 to selenographic rotation matrix. |
| M50MDT | Determines Earth mean of 1950 to Earth mean of date rotation matrix. |
| NUTAIT | Determines rotation matrix from Earth mean of date to prime meridian. |
| NUTATE | Determines rotation matrices from Earth mean of date to Earth prime meridian and selenographic. |
| OBLATE. | Determines approximate effect due to $J_{2}$ when using multiconic. |
| OBLE | Determines disturbing acceleration due to an oblate Earth. |
| OBLTY | Determines matrix that rotates through the mean obliquity of the ecliptic. |
| OBSET | Gravitational field initiation. |
| ORBIT | Determines cartesian state from orbital elements and vice versa. |
| ORIENT | Determines rotation so that two vectors have desired dot product. |
| OUTPUT | Outputs trajectory time history. |
| OUT1 | Determines time to output trajectory. |
| PLANET | Determines the positions of the planets. |
| POST | Computes post=targeting information for output. |
| PRINT | Outputs common blocks |
| PROTO | Controls midcourse analysis. |
| PUTELS | Outputs elements into a hold file for later use by the GTDS program. |

QUARTC
QUIKIE

RANDM1

READE

RETDV

RETRO

RKSEVN

ROTAIT

ROTATE

SADOUT Outputs shadow times.
SENSO

SETUP2

SHADOW

SHORB2

SOL
SOLP

SPER
SPNM

SUNMIN

TABINT

TARGET

TIMEC

TOBODY

TRIM
Solves a quartic. between 0 and 1 . ephemeris.

Rotates two vectors in a plane. velocity.

Initialization of constants and flags.

Approximate Solar ephemeris. Calculates Legendre polynomials spacecraft is reorientated. Numerical integration control.

Determines the number of days an orbit has shadows.
Determines a random number uniformly distributed

Determines the planet's ephemerides from tape or disk

Impulsive velocity calculation for retro motor.
Determines the retro firing conditions.
Seventh-order Runge-Kutta numerical integration.

Matrix vector multiplier for $3 \times 3$ matrix (entry name for MVTRN)

Computes gradient of end constraints w.r.t. midcourse

Determines times of shadow crossing.
Determines the intersections of an orbit with the shadow cone.

Determines the acceleration due to solar pressure.

Determines spherical coordinates from cartesian coordinates.

Determines minimum angle between spin axis and Sun as

Determines thrust and weight from tabular input . Determines midcourse constraint vector at target planet. Flies along a Keplerian conic for a time interval. Determines post-injection trim velocity.

| TRIM2 | Determines two-imnulse nlane change maneuver. |
| :--- | :--- |
| TRMN | Determines mean from true anomaly and vice versa. |
| TUBE1 | Generates graphics data base for use on IBM 2250. |
| TWELVE | Twelfth-order multi-step numerical integrator. |
| TWOPIT | Determines optimum position on orbit for 2 -impulse plane <br> change maneuver. <br> UPDATE |
| Establishes array of back values for interpolation. |  |
| VISIB | Determines lighting characteristics of planets. |
| VNORM | Computes tracking station visibility. |
| Determines unit vector. |  |

## Section 1

## INTRODUCTION

This manual contains a description of the Mission Analysis Evaluation and Space Trajectory Operations program known as MAESTRO. MAESTRO is an all FORTRAN, block style, computer program designed to perform various mission control tasks. It is intended that this manual serve as a guide to MAESTRO, thereby providing individuals the capability of modifying the program to suit their needs. Of most importance to this task, this manual contains descriptions of each of the subroutines which comprise MAESTRO. These subroutine descriptions consist of input/output description, theory, subroutine description and a flow chart where applicable. The programmer's manual also contains a detailed description of the common blocks, a subroutine cross reference map and a general description of the program structure.

## Section 2

## MAESTRO STRUCTURE

MAESTRO is an all FORTRAN, block style, computer program designed to perform various mission control tasks. These tasks include

1. Retro motor firing time and attitude determination, APROCH.
2. Midcourse correction determination, PROTO.
3. Verification of midcourse correction, MCVERF.
4. Lunar lifetime prediction, FOWARD.
5. Post-injection trim determination, ARMPIT.

The subroutines responsible for performing the mission control tasks are also shown above. All of these subroutines are essentially self-contained except for PROTO which controls the midcourse determination. The logic involved in this task is more complex than the others, therefore, it will be discussed in detail later in this section.

All of the mission analysis tasks require a means of propagating the state forward in time. Subroutine FOWARD and its related subroutines accomplish this task. The subroutines that make up subroutine FOWARD comprise the major portion of MAESTRO. The second subsection describes the structure of FOWARD.

### 2.1 Main Control Logic

MAESTRO computation is initialized in the MAIN program. Figure 2.1.1 presents a flow chart of this routine.

The first task of the MAIN program is to clear some common blocks and call subroutine INPUTF. Subroutine INPUTF reads the input data cards and establishes INPUT and FIELDM common blocks. MAIN calls subroutine CONTRL to initialize common blocks. Most of the initialization is actually accomplished in subroutine SETUP2. After the initialization process is complete, MAIN calls one of the mission analysis subroutines to perform the desired analysis, The MODE flag of input common is used to key
the proper subroutine. The logic flow is returned to the input of data to initiate a second case after the analysis is complete.

### 2.2 Trajectory Propagation, FOWARD

A call to subroutine FOWARD will propagate the state forward in time. The state will be propagated by numerical integration or by the multiconic algorithm. Subroutine MULCON is used when the multiconic algorithm is employed to propagate the state. Subroutines TIMEC, INTEG, EQNS, and ACCEL are employed for numerical integration.

Figure 2.1.1 presents a flow chart of the basic control logic involved in trajectory propagation. The initial state is brought into the subroutines via STATE common. The METH flag in INPUT common is used to determine the trajectory propagation scheme. If the multi-conic technique is desired, subroutine MULCON is called and the state propagated forward in that routine. If one of the numerical intregration schemes is requested, subroutine FOWARD calls subroutine TIMEC to initiate the numerical integration. As seen from Figure 2.1.1, the numerical integration logic consists of looping through subroutines TIMEC, INTEG and RKSEVN or TWELVE. Each loop through the subroutines numerically integrates the state one compute interval. The size of the compute interval is determined in TIMEC. Thus, when flow returns to TIMEC at the end of a loop, the state in STATE common corresponds to the spacecraft's state at the end of the time step.

Subroutine INTEG is used to establish the integration array before integration and determine the position and velocity vectors after the step is complete. An intermediate array is used in the numerical integration subroutines RKSEVN and TWELVE. This array is necessary because a variety of variables can be integrated. The trajectory propagation flag, METH, determines which set of variables are transferred from STATE common to the integration array in INTVAR common. After integration is complete, the variables at the end of the state are transferred back into STATE common. The position and velocity vectors are also determined because they are needed for use in other auxiliary calculations after integration. The integration array separates the numerical integration logic from the trajectory propagation logic. Thus, new numerical integration or trajectory propagation techniques can be easily added
without affecting logic in the other segment. Integration of other quantities can also easily be included. For example, the equations which describe the state transition matrix can be included in the numerical integration without any changes to the numerical integration subroutines.

The actual integration that is carried out is a loop comprised of subroutines. EQNS, ACCEL and RKSEVN or TWELVE. Subroutines RKSEVN and TWELVE are single and multi-step numerical integration schemes. The derivatives of the quantities to be integrated are determined in subroutines ACCEL and EQNS. EQNS contains logic to determine the derivatives according to the trajectory propagation scheme employed. The derivatives are transferred to the numerical integration routines via RATES of INTVAR common.

There are many other subroutines besides the ones mentioned which are involved in propagating the state forward in time. These routines are used in the determination of shadow times, closest approach times, trajectory output and interpolation. Most of the calls to perform these auxiliary calculations are made from subroutine TIMEC.

### 2.3 Midcourse Guidance Structure

The objective of the midcourse guidance package is to determine the velocity correction which nulls a constraint vector at the target planet. The velocity correction is determined through a generalized Newton-Raphson technique. The partial derivatives of the constraints with respect to the velocity correction are determined by finite differences.

The midcourse guidance calculations are initiated in subroutine PROTO, (see Figure 2.3.1). PROTO's functions are to initialize constants, increment execution times, write the midcourse tape and output displays.

The Newton-Raphson logic to determine the correction is contained in a loop inside subroutine MDCORS. This loop uses subroutines SENSO, MCBURN and TARGET to perform many of the midcourse calculations along with subroutine JET to pre-target. Subroutine JET uses a variety of conic techniques to determine an initial value of the correction velocity. This velocity is used as the initial guess in the Newton-Raphson
iteration process. The iteration process consists of the following steps in MDCORS:

1. The constraint vector is determined at the target planet. Subroutines SENSO, MCBURN and TARGET are required. SENSO sets up the logic to propagate the state to the target planet and calls MCBURN to apply the current correction. SENSO next calls FOWARD to obtain the state at the target planet and then TARGET to convert the state to the constraint vector.
2. Tests are made on the constraint vector to determine if each component is within tolerances. If the vector is within the desired tolerances, the solution is converged and flow transferred back to subroutine PROTO.
3. The secant matrix is determined if the solution is outside the bounds. The secant matrix is determined by incrementing each of the control variables independently. The constraint vector is found for each control variable in the same manner as described in step 1.
4. The velocity correction is found by inverting the secant matrix and multiplying by the difference of the current constraint vector from the desired constraint vector.
5. Flow is transferred to step 1.

After the midcourse correction is determined, subroutine PROTO calls POST to obtain auxiliary output quantities. These quantities along with the correction and the pre-midcourse state are written on an output device. If more midcourse execution times are to be analyzed, flow is transferred back to the call to MDCORS to evaluate the next correction. After the last correction is determined, the midcourse tape is read and the appropriate displays output.

There are subroutines involved in the midcourse calculation other than those already discussed. Most of these are used to perform calculations necessary in JET, TARGET, MCBURN and POST. A complete description of all the subroutines is presented in Section 5.



Figure 2.3.1

MIDCOURSE GUIDANCE STRUCTURE


## Section 3

## CROSS REFERENCE MAP

### 3.0 Introduction

The MAESTRO system is comprised of more than 90 subroutines. It is difficult to understand the structure of a program of this complexity without some sort of a visual guide. The maps on the following pages are designed to meet this need. These maps present a hierarchy of subroutine calls for each division of the program.

The program is divided into three sections for clarification. The first section consists of the prime control logic. The second division presents the midcourse logic, while the third section consists of the hierarchy of subroutine FOWARD.

### 3.1 Primary Control Logic

The primary control logic consists of those subroutines required for input, initialization and selecting the analysis modes of MAESTRO. The cross reference map of the primary control logic is shown in Table 3.1.

### 3.2 Midcourse Logic

The midcourse guidance section comprises a substantial portion of MAESTRO. The midcourse logic can be entered directly from the MAIN program by a call to PROTO or from subroutine MONTE via calls to MDCORS, SENSO and MCBURN. A cross reference map of the subroutines involved in the midcourse logic is presented in Table 3.2.

### 3.3 Subroutine FOWARD Logic

Subroutine FOWARD's sole aim is to propagate the state forward in time. This subroutine comprises the major portion of MAESTRO and is called from many other subroutines. Table 3.3 presents a cross reference map of the subroutines involved in this portion of the logic.

## MAIN CONTROL LOGIC CROSS REFERENCE MAP



Table 3.2
MIDCOURSE GUIDANCE CROSS REFERENCE MAP


Table 3.3
SUBROUTINE FOWARD CROSS REFERENCE MAP


## COMMON BLOCKS

Currently, there are 25 common blocks established in MAESTRO. Most of these common blocks are required for the numerical integration portion of MAESTRO. The table below presents a list of the current common blocks:

| Used in Numerical $\qquad$ Integration | Used in Ephemeris $\qquad$ | Used in Auxiliary Calculations |
| :---: | :---: | :---: |
| AVG | CETBL2 | ANKOR |
| CNTRL | CETBL 3 | ELMNT |
| CONST | CETBL9 | INPUTS |
| DUM | MOON | INTER |
| FIELDM |  | MCCOM |
| GRAVTY |  | OBSIT |
| INPUT |  | PIT |
| INTVAR |  | SHAD |
| INTVRX |  |  |
| PERT |  |  |
| PLNET |  |  |
| SAVE |  |  |
| STATE |  |  |

The following pages in this section present a description of each of the above common blocks. The descriptions include the usage of the common block, its length, the subroutines which required the common block, and a description of each of the elements that comprise the common block,

## COMMON ANKOR

| Description: |  | This common block contains the current anchor vector |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Length: |  | 6 8-byte words |  |  |  |
| Subroutines Using: |  | CONTRL | MONTE | PROTO | SETUP2 |
| Location | Symbolic <br> Name |  | Descri | ion |  |
| 1 | ANKUE(6) |  | esian pos ent ancho equinox of | on and ve vector in 950, km | city vect arth mean d km/sec |

## COMMON AVG

| Description: | This common block contains the weights and the <br> corresponding abscissa for the Gaussian quad- <br> rature formulae. |
| :--- | :--- |
| Length: | 156 -byte words |
| Subroutines Using: | AVEQNS, BLOCK DATA, MAIN |


| LOCATION | SYMBOLIC NAME | DESCRIPTION |
| :---: | :---: | :---: |
| 1 | WEIGHT (78) | Weights for Gaussian quadra- <br> ture formulae* |
| 79 | ABSCIS (78) | Abscissa for Gaussian <br> quadrature formulae* |

[^0]Description:

Length:
Subroutines Using:

| Location | Symbolic <br> Name |
| :--- | :--- |
| 1 | ICW |
| 2 | ICENTR |

Symbolic
Length:
Subroutines Using:

This common block contains arrays that determine which planets are desired in the disk or tape ephemeris call using subroutine READE

15 4-byte integer words
MAIN GETTAP PLANET READE SETUP2

## Description

Flag indicating the status of the common block CETBL3

Central planet number defined as

| 1 | Mercury | 4 | Mars | 7 |
| :--- | :--- | :--- | :--- | :--- |
| Uranus |  |  |  |  |
| 2 | Venus | 5 | Jupiter | 8 |
| Neptune |  |  |  |  |
| 3 | Earth | 6 | Saturn | 9 | Pluto

10 Sun
11 Moon
12 Oddball
$3 \quad \operatorname{IREQ}(13)$
Array used to determine which planets the ephemeris is desired. The planets 'correspond to the index of the array as described in ICENTR. The value of IREQ is determined from:

$$
\begin{aligned}
\operatorname{IREQ}(J) & =0 & & \text { no ephemeris } \\
& =1 & & \text { position only } \\
& =2 & & \text { position and velocity }
\end{aligned}
$$

| Description: |  | This common block is used to transfer information read from the ephemeris tape to subroutines GETTAP and READE. |
| :---: | :---: | :---: |
| Length: |  | 829 -byte words followed by 205 real 4-byte words |
| Subroutines Using: |  | $\begin{aligned} & \text { MAIN GETTAP PLANET READE } \\ & \text { SETUP2 } \end{aligned}$ |
| Location | $\begin{array}{r}\text { Symbolic } \\ \text { Name } \\ \hline\end{array}$ | Description |
| 1 | TAB3(829) | Array of raw data obtained from the JPL ephemeris tape. Data contains 8 days of information. |
| 830 | NUT(204) | Array of 4-byte real words describing the nutation data obtained from the JPL ephemeris tape. |
| 1034 | CKSUM | A 4-byte word used for checksum. |

## COMMON CETBL9

| Description: |  | This common block contains quantities required in the tape or disk ephemeris calculations. |
| :---: | :---: | :---: |
| Length: |  | 38 -byte words followed by <br> 1 4-byte integer word |
| Subroutines Using: |  | GETTAP READE |
| Location | Symbolic Name | Description |
| 1 | JD1 | Reference julian date of ephemeris call. |
| 2 | TDAY | Time since reference julian date ephemeris is desired, days. |
| 3 | JDIF | Difference in the time of the desired ephemeris minus the time of the ephemeris data read from the tape or disk. |
| 4 | IERR1 | Error return flag from subroutine GETTAP |

## COMMON CNTRL

| Description: |  | CNTRL common is used to store flags that control state propagation. |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Length: |  | 28 4-byte integer words |  |  |  |
| Subroutines Using: |  | MAIN | ACCEL | APROCH | AVEQNS |
|  |  | AVSTRT | BELL | CLOSE | CRASH |
|  |  | DOPLER | DRAG | EQNS | FIELD2 |
|  |  | FIXATG | FOWARD | GETTAP | GRAV |
|  |  | INTEG | INTERP | LUNA | MCBURN |
|  |  | MCVERF | MDCORS | MONTE | MOTORS |
|  |  | MULCON | OBLE | OUTPUT | PLANET |
|  |  | PRINT | PROTO | RKSEVN | SETUP2 |
|  | . | SHADOW | SOL | TARGET | TIMEC |
|  | Symbolic TRIM2 TWELVE |  |  |  |  |
|  |  |  |  |  |  |
| Location | Name | Description |  |  |  |
| 1 | - | Not used |  |  |  |
| 2 | KTHRST | Thrusting flag. Set to 1 when an engine is thrusting. |  |  |  |
| 3-4 | - | Not used |  |  |  |
| 5 | KDIS | Discontinuity flag. Flag equals 1 when the current time is a discontinuity time, otherwise set to 0 . |  |  |  |
| 6 | KHALT | Error return flag. Flag is set to 1 to cause an error return from the numerical integration. |  |  |  |
| 7 | JC | Central planet number. |  |  |  |
| 8 | KREAD | Flag used to determine if the ephemeris tape or disk is to be read at the current time. |  |  |  |
| 9 | KNT | Counter that contains the number of calls to the derivative subroutine, EQNS. |  |  |  |
| 10 | KDOPWT | Doppler write flag. Flag equals zero on the first call to DOPLER. On subsequent calls this flag is set to 1 . |  |  |  |
| 11 | KCA | Co sub nu | $r$ used in the <br> ines CRASH <br> of iteration | st approac IMEC. KC | ation between uals the |

Location | Symbolic |
| :--- |
| Name |$\quad$ Description

12

13

14-28

KFIRST

IEVG
-

First pass flag. Flag equals one on the first step of a numerical integration

Engine number of engine thrusting
Not used

| Description: |  | This common block contains physical constants and unit conversion factors |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Length: |  | 50 8-byte words |  |  |  |
| Subroutines Using: |  | MAIN | APROCH | ARMPIT | AVEQNS |
|  |  | AVERGE | AVSTRT | BLOCK DATA | BURND |
|  |  | CLOSE | CONTRL | DOPLER | DRAG |
|  |  | EQNS | FIELD2 | FIXATG | FOWARD |
|  |  | GRAV | INTEG | INTERP | JET |
|  |  | LUNA | MCBURN | MCSET | MCVERF |
|  |  | MDCORS | MINV | MONTE | MULCON |
|  |  | M50EPM | M50JPM | OBLATE | OBLE |
|  |  | OBLTY | ORBIT | OUTPUT | POST |
|  |  | PRINT | PROTO | QUIKIE | RETDV |
|  |  | SETUP2 | SHADOW | SOL | SOLP |
|  |  | SPER | SUNMIN | TARGET | TIMEC |
|  |  | TOBODY | TRIM | TRIM2 | TRMN |
|  | SYMBOLIC | TWELVE | TWOPIT | VISIB |  |
|  |  |  |  |  |  |
| LOCATION | NAME | DESCRIPTION |  |  |  |
| 1 | RAD | Rad | to degree c | ion factor |  |
| 2 | PI | Pi, |  |  |  |
| 3 | PI2 | Twi |  |  |  |
| 4 | Au | Num | of kilomet | an astronomical | unit |
| 5 | GM(12) | Gravitational coefficients of the planets, $\mathrm{km}^{3} / \mathrm{sec}^{2}$. The planets are ordered in the array as follows, |  |  |  |
|  |  | 1. Mercury 4. Mars 7. Uranus 10. Sun <br> 2. Venus 5. Jupiter 8. Neptune 11. Moon <br> 3. Earth 6. Saturn 9. Pluto 12. Oddball |  |  |  |
| 17 | RE(12) | Equatorial radii of the planets, km . The order is the same as the gravitational coefficients. |  |  |  |
| 29 | WP(12) | Rotation rates of the planets, $\mathrm{rad} / \mathrm{sec}$. The order is the same as the gravitational coefficients. |  |  |  |
| 41 | THS | Hour to second conversion factor |  |  |  |
| 42 | TSH | Second to hour conversion factor |  |  |  |
| 43 | TDS | Day to second conversion factor |  |  |  |
| 44 | TSD | Second to day conversion factor Gravitation acceleration at the surface of the Earth, km/sec ${ }^{2}$ |  |  |  |
| 45 | G |  |  |  |  |

## COMMON DUM

| Description: |  | This common block contains quantities used to propagate the state using the multi-conic scheme. |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Length: |  | 12 8-byte words |  |  |  |
| Subroutines Using: |  | AVERGE | E MULCON | OBLATE | TOBODY |
| Location | Symbolic Name | Description |  |  |  |
| 1 | A1S(3) | Acceleration due to the Moon's indirect term, $\mathrm{kms} / \mathrm{sec}^{2}$ |  |  |  |
| 4 | A2S(3) | Acceleration due to the Sun, $\mathrm{km} / \mathrm{sec}^{2}$ |  |  |  |
| 7 | PM | Spacecraft's mean motion, rad/sec |  |  |  |
| 8 | - | Not used |  |  |  |
| 9 | AM | Value of mean anomaly at end of a compute step, rad |  |  |  |
| 10 | DT | Current multi-conic compute step, sec. |  |  |  |
| 11 | AMO | Value of the mean anomaly at the beginning of the step, rad. |  |  |  |
| 12 | - |  | Not used. |  |  |




## COMMON GRAVTY

Description:

Length:
Subroutines Using:

Location
1

4
Symbolic

POS(3) $\cdot$

VEL(3)

This common block contains vectors used in the determination of the disturbing acceleration when numerically integrating

6 8-byte words

| ACCEL | AVEQNS | DRAG | EQNS |
| :--- | :--- | :--- | :--- |
| FIELD2 | GRAV | OBLE | SOLP |

Description
Position vector with respect to central planet in Earth mean equator and equinox of 1950 , km

Velocity vector in same system, $\mathrm{km} / \mathrm{sec}$

Description:
Length:

Subroutines Using:

This common block contains the data input to MAESTRO.
1000 8-byte real words followed by 100 4-byte integer words

| MAIN | ACCEL | APROCH | ARMPIT |
| :--- | :--- | :--- | :--- |
| AVEQNS | AVSTRT | BELL | BLOCK DATA |
| BURND | CALEND | CLOSE | CONTRL |
| CRASH | DOPLER | DRAG | EQNS |
| FIELD2 | FIXATG | FOWARD | GRAV |
| INPUTF | INTEG | INTERP | JET |
| LUNA | MCBURN | MCSET | MCVERF |
| MDCORS | MINV | MONTE | MOTORS |
| MULCON | OBLATE | OBLTY | OUTPUT |
| OUT1 | PLANET | POST | PRINT |
| PROTO | RETDV | RETRO | RKSEVN |
| SADOUT | SENSO | SETUP2 | SHADOW |
| SOL | SOLP | TABINT | TARGET |
| TIMEC | TRIM | TRIM2 | TWELVE |
| TWOPIT | UPDATE | VISIB |  |


| LOCATION | FORTRAN SYMBOL | USE*** | $\begin{gathered} \text { PRESET } \\ \text { VALUE } \\ \hline \end{gathered}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1 | ERRC | I | - | Error control for automatic integration step size. If $=0$ assume in fixed-step mode. |
| 2 | DELTO | I | - | Initial computer step when in automatic mode. Note: Need not be input when fixed-step. |
| 3 | DELMN | I | - | Minimum compute interval. Note: Not necessary in fixed-step mode. |
| 4 | TF | I | ** | Run stop-time. |

Unit conversion factor for positions. The value is determined such that, when PCON multiplies the input units, they will be converted to KMIS. The output units will be scaled by $1 / \mathrm{PCON}$ so that the output units will be the same units as input.

6

7

8
TOL
I
0.0

Used the same as PCON except that velocities are scaled to KM/SEC.

Corrector convergence tolerance in twelfth-order predictor-corrector

9
EMPTY

* unless otherwise stated, the units of the input quantities are: KM, SEC, KG,degrees ** preset value depends on program operating mode

| $* * *$ | I | integration | V | Midcourse verification |
| :---: | :---: | :--- | :---: | :---: |
|  | A | approach analysis | L | Lifetime analysis |
|  | M | midcourse analysis | P | Post-injection trim |
|  | C | Monte Carlo analysis |  |  |


| docation | FORTRAN SYMBOL | USE | PRESET VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| iô- $\overline{\mathbf{i}} \mathbf{Z}$ | TMETH( ${ }_{\text {a }}$ ) | I | ** | Times for method table. See Kilitit (location 1036) for description. |
| 13-19 | EMPTY | I | - | Launch epoch |
| 20 | XMONL DAYL |  |  |  |
| 22 | YRL |  |  |  |
| 23 | HRL XMINL |  |  |  |
| 24 25 | XMINL |  |  |  |
| 25 |  |  |  |  |
| 26-29 | EMPTY |  |  |  |
| 301 | ELM(6) | 1 | - | Input initial conditions as orbital elements. The order is a, e,f, $\omega, i, \Omega$. The initial conditions may be accepted as position and velocity vectors (see locations 40-45). The coordinate system is defined in KINPT (location 1019). |
| 31 |  |  |  |  |
| 32 |  |  |  |  |
| 33 <br> 34 |  |  |  |  |
| ( 35 |  |  |  |  |
| 36 | EMPTY |  |  |  |
| 37 | DJL |  |  | Modified julian launch date. Not a program input. |
| 38 | wTo | I | 331.40 | Initial weight |
| 39 | ETC | I | - | Ephemeris time correction. If no input, the ephemeris time correction will be calculated from $\mathrm{ETC}=38.66+.0025921 \mathrm{DJ}$ where DJ is the number of days since the julian date of 2440000.0 . |
| 40 | X(3) | I | - | Initial conditions as position and velocity vectors or spherical coordinates. When spherical coordinates are used the order is velocity, flight path elevation angle, flight path azimuth angle, radius, geocentric latitude, geocentric longitude. The input coordinate system is defined by KINPT (location 1019) |
| 41 |  |  |  |  |
| 42 |  |  |  |  |
|  |  |  |  |  |
| 43 | DX(3) | I | - |  |
| 44 |  |  |  |  |
|  |  |  |  |  |
| 46 | DJO |  |  | Modified julian launch date. Not a program input. |
| 47 | RA |  |  | Initial right ascension and declination |
|  |  |  |  | in Earth equator and equinox of 1950 |
| 48 |  |  |  |  |


| LOCATION | FORTRAN <br> SMMBOL | USE | $\begin{gathered} \text { PRESET } \\ \text { VALUE } \\ \hline \end{gathered}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 50 | XMON0 |  |  |  |
| 51 | DAY0 |  |  | Initial time |
| 52 | YR0 |  |  |  |
| 53 | HR0 | I | - | Input is month, day and year., |
| 54 | MIN0 |  |  | hour, min, second (GMT). |
| 55 | SEC0 |  |  |  |
|  |  |  |  | Upper right-hand triangle of the tracking covariance matrix loaded across the rows. Can be input in mean equator and equinox of 1950 or in a local tangent |
| 56-76 | COV | C | $\bigcirc$ | plane. Location 1085 determines the system. When mean of 1950 is used, the order is $X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$. When local tangent plane is input, the order is the position components as $R$, ( $R \times V$ ) $\times R, R \times V$ and then the velocity components along the same axes. |
| 77-99 | EMPTY |  |  |  |
| 100-111 | GM(12) | I | *** | Gravitational coefficient. The order of the planets is defined in locations 1001-1012. |
| 112-123 | RE(12) | I | *** | Planetary radii. |
| 124-135 | WP(12) | I | *** | Planetary rotation rates |
| 136-169 | EMPTY |  |  |  |
|  |  |  |  | Compute interval table when in fixed compute interval mode, (loc ( 1 ) =0.). |
| 170-179 | TCOMP(10) | I | ** | Compute interval $=$ DELT ( I ) . |
| 180-189 | DELT(10) | I | ** | when $\operatorname{TCOMP}(I-1)<T<T C O M P(1)$ or Compute interval $=$ DELT (1) when $T<T C O M P(1)$. Complete compute interval table must be loaded when alterations are made to preset values. |
| 190-192 | XM ¢ $^{\prime}$ N(3) | I | - | Initial position of Moon if osculating elements are used for Moon's position; |
| 193-195 | DXM $\varnothing \varnothing \mathrm{N}(3)$ | I | - | Earth radii and Earth radii/mean solar day. |
| 196 | TMM | I | - | Moon's epoch in modified julian date. |
| 197 | SPRESS | I | $4.7(10)^{-5}$ | Solar pressure at 1 AU , dynes/cm ${ }^{2}$ |

[^1]| LOCATION | FORTRAN SYMBOL | USE | PRESET <br> VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 198 | REFLK | I | 0.2 | Solar pressure reflectivity coefficient |
| 199 | EMPTY |  |  |  |
| 200-219 | TF1(20) | I, A, M, V | *** | Tabular thrust history of Motor 1. TF1(20) are times since ignition and F1(20) are the thrust values (newtons) |
| 260-279 | F1(20) | $\mathrm{I}, \mathrm{A}, \mathrm{M}, \mathrm{Y}$ | *** | at the corresponding times. For the RAE-B application, this motor is used as the midcourse motor. |
| 220-239 | TF2(20) |  | $-\}$ | Thrust table for motor 2. |
| 280-299 | F2(20) |  | - . |  |
| 240-259 | TF3(20) |  | $-\}$ | Thrust table for motor 3. |
| 300-319 | F3(20) |  | - |  |
| 320-329 | TWDOT1(10) | $\mathrm{I}, \mathrm{A}, \mathrm{M}, \mathrm{V}$ | *** | Tabular weight flow rate for motor 1. TWDOT1 is the time past ignition and |
| 350-359 | WDOT1(10) | $\mathrm{I}, \mathrm{A}, \mathrm{M}, \mathrm{V}$ | *** | WDOT is the flow rate at the corresponding time. The flow rate is in KG/sec. |
| 330-339 | TWDOT2(10) | 1 | - | Flow rate table for motor 2. |
| 360-369 | WDOT2(10) | 1 | - |  |
| 340-349 | TWDOT3(10) | I | - | Flow rate table for motor 3. |
| 370-379 | WDOT3(10) | 1 | - |  |
| 380-382 | TIG(3) | I, V | - | Ignition times for motors 1, 2 and 3. |
| 383-385 | TB $\varnothing$ (3) | I, V | - | Burnout times for motors 1, 2 and 3. |
| 386-399 | EMPTY |  |  |  |
| 400 | TFIRE1 | A | - | First and last trial retro firing time on an approach analysis. Time reference |
| 401 | TFIRE2 | A | - | to liftoff epoch. Used when KAPOPT $=3$ (location 1055) |
| 402 | DTFIRE | A | - | Increment in retro firing time. . |
| 403 | RA0 | A | $-1$ | Initial right ascension and declination |
| 404 | DEC0 | A | - | used in the attitude sweep in the approach analysis. If both zero, velocity vector at closest approach is used. |


| LOCATION | FORTRAN <br> SYMBOL | USE | PRESET VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 405 | DELRA | A | - | Increment in right ascension and |
| 406 | DELDEC | A | - | declination in attitude sweep. Used when KAPOPT $=2$ or 3 |
| 407 | EMPTY |  |  |  |
|  | - | --- |  |  |
| 408 | CAR1 | V | 4.0095 D 8 | Telemetry carrier frequency No. 1 |
| 409 | CAR2 | V | 4.0D8 | Telemetry carrier frequency No. 2 |
| 410-419 | OBSLON(10) | A, M, V | *** | Observation site geocentric longitude for sites 1-10. |
| 420 | $\operatorname{PSID}(1)$ | M | 2838. | Lunar radius |
| 421 | PSID(2) | M | 116.5 | inclination w.r.t. target planet equator. <br> Midcourse |
| 422 | PSID(3) | M | . 396 D 6 | $\left.\begin{array}{l}\text { time of flight w.r.t. } \\ \text { liftoff }\end{array}\right\}$analysis <br> desired end |
| 423 | PSID(4) | M | 0.62 | Hyperbolic excess speed conditions |
| 424 | PSID(5) | M | 0.0 | circular excess speed $\quad$ |
| 425-429 | $\operatorname{PSD}(6-10)$ | M |  | Empty |
| 430-432 | DV(3) | I | - | Impulsive velocity of engines 1 to 3. Used when KFM $\varnothing \mathrm{N}$ (location 1047) is set to 2. Velocities are added at ignition times in location (380-382). |
| 433 | SOLARA | I | 13560 | Area of the spacecraft used in solar pressure calculation, square centimeters |
| 434 | DELTMC | M | 7200. | Increment in execution time |
| 435 | STGATM | C, M | 0.7 | One sigma pointing error during midcourse maneuver. |
| 436 | SIGDVM | $\mathrm{C}, \mathrm{M}$ | 0.02 | One sigma percentage error of midcourse velocity. Loaded as percent/ 100 . |
| 437 | SIGATR | C | 0.7 | One sigma pointing error during retro. |


| LCCATION | $\begin{aligned} & \text { FORTRAN } \\ & \text { SYMBOI } \end{aligned}$ | USE | $\begin{gathered} \text { PRESET } \\ \text { VALUE } \end{gathered}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 438 | SIGDVR | C | 0.0003 | One sigma percentage error of retro velocity. Loaded as percent/ 100 . |
| 439 | TMC1 | C | 36000. | Time of first correction |
| 440 | TMC2 | C | 324000. | Time of second correction |
| 441 | ASPMC | A, M, C | 226.0 | Midcourse motor specific impulse |
| 442 | ASPR | $A, M, C$ | 282.5 | Retro motor specific impulse |
| 443 | WRETRO | A, M, C | 71.44 | Mass of retro fuel |
| 444 | RO A | $A, M, C, P$ | 2838. | Desired lunar orbit radius used in trim maneuver |
| $445$ | VC A | $A, M, C, P$ |  | Circular velocity at RO. Program calculation, not to be input. |
| 446 | CIO A | $A, M, C, P$ | 116.5 | Desired lunar orbit inclination w.r.t. lunar equator used in trim maneuver |
| 447 448 | $\operatorname{BVD}(1)$ $\operatorname{BVD}(2)$ | $M$ $M$ | 6000. 6000. | Desired B.T Midcourse analysis <br> desired end conditio: <br> Desired B•R $\quad$Used when IBTR flag <br> set to 1. (loc 1062) |
| 449 | TRINC | $\mathrm{M}, \mathrm{C}, \mathrm{P}$ | 0. | Tolerance band on inclination correction during post-injection trim. If inclination change is less than TRINC, no inclination adjustment is made. |
| 450-459 | TOUT(10) | I | $\begin{aligned} & \operatorname{TOUT}(1)= \\ & \text { 1. } \mathrm{D} 20 \end{aligned}$ | Print table |
| 460-469 | DTOUT(10) | I | $\begin{gathered} \text { DTOUT(1)= } \\ 18000 . \end{gathered}$ | $\begin{aligned} & \text { Print interval = DTOUT(1) when } \\ & \text { TOUT(I-1) }<T<T O U T(I) \text { or print } \\ & \text { interval }=\text { DTOUT(1) when } T<\operatorname{TOUT}(1) \end{aligned}$ |
| 470 | ATFULA | C | 6.0 | Available attitude control fuel and constant to determine attitude control fuel, KG/RAD. |
| 471 | AFUEL | C | 0.1678 | Attitude fuel used $=$ AFUEL * attitude angle change. |
| 472 | FTфT | C | 20.4 | Total midcourse fuel available |


| LOCATION | FORTRAN SXMBOL | USE | PRESET <br> VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 473 | WDROP | A，M，C | 13.77 | Retro drop weight．Weight dropped after retro firing． |
| 474 | CONE | A，M | 5.0 | Cone angle used in approach analysis． The attitude range is this cone angle about the velocity vector or input attitude when KAPOPT＝1（loc 1055）．Also used in midcourse fixed attitude scan． |
| 475 | TRUE | A，M，C | 20 | True anomaly range for trial retro firings in approach analysis，deg．Trial retro firings are made from－True to＋True true anomaly on the approach hyperbola． Used when $K A P O P T=1$ ．Also used as firing range in retro optimization． |
| 476 <br>  <br> 477 | BURNT EMPTY | V | － | Midcourse motor burn time．Used when the initial state is not obtained from a midcourse analysis． |
| 478 | TMC | M | 7200. | Initial execution time |
| 479 | DINK | M | 0.0003 | Secant partial step size in midcourse amalysis also velocity increment when using the fixed attitude mode． |
| 480－489 | OBSLAT（10） | A，M，V | ＊＊＊ | Observation site geocentric latitude for sites 1－10 |
| 490 | T $\phi$ L（1） | M | 10. | B－T |
| 491 | $\mathrm{T} \phi \mathrm{L}(2)$ | M | 10. | B－R ．Midcourse |
| 492 | T $\phi \mathrm{L}(3)$ | M | 10. | time of flight analysis |
| 493 | $\mathrm{T} \phi \mathrm{L}(4)$ | M | ． 0001 | hyperbolic excess speed $\quad$ analysis |
| 494 | $\mathrm{T} \phi \mathrm{L}(5)$ | M | ． 0001 | circular excess speed $\quad$ tolerances on |
| 495 | TめL（6） | M | ． 02 | Total fuel optimization desired end |
| 496 | TめL（7） | M | 5. | closest approach radius desired end |
| 497 | TめL（8） | M | ． 2 | inclination $\quad$ conditions |
| 498－499 | T $\phi \mathrm{L}(9,10)$ | ． |  | Empty |
| 500－511 | RSWTCH（12） | I | ＊＊＊ | Spheres of influence of the planets used to determine the central planet．If the distance from the planet is less than the value in RSWTCH，the planet is the central planet．If none of the planets are central， then the Sun is considered central． |

[^2]| LOCATION | $\begin{aligned} & \text { FORTRAN } \\ & \text { SYMBOL } \end{aligned}$ | USE | PRESET <br> VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 512 | RBURN | A | 20. | Retro burn time |
| 513 | TCATST | I | 0. | Time to begin closest approach testing logic. |
| 516 | SPNRA | I | 12.0 | Spin rate, RPM. |
| 517 | PTI | I | 238.0 | Initial midcourse motor tank pressure, PSIA. |
| 518 | TPI | I | 20. | Initial midcourse motor temperature, ${ }^{\circ} \mathrm{C}$. |
| 519 | FMC | I | 20.4 | Current midcourse motor fuel, ke. |
| 520 | FMULT | I | 1. | Thrust multiplier. |
| 521 | WMULTT | I | 1. | Weight multiplier. |
| 522 | WDOTT | M | . 0183 | Midcourse motor weight flow used with Hamilton Standard thrust in midcourse targeting. |
| 523 | DTM | M | 0. | Compute interval during motor burn in midcourse targeting. Compute interval equals burntime when value is zero. |

 set to 0 when $1029=3$.


LOCATION
$1039-1040$

1041
1042

1043

1044

1045

1046

1047

1048

1049

KOUTPT(2) I

A, M
A, M

KFIRE

MODE
$\mathrm{KD} \varnothing \mathrm{P}$.. V

KAPOUT

KFMOD

KAPSAD

KSADめW

ALL

A
A, M $\quad 20$

Output coordinate systems - Launch and target planets

1. Mean equinox and ecliptic of date
2. True equator and prime meridian
3. Mean Earth equator and equinox of 1950
4. True Earth equator and equinox of date
5. No output

Number of right ascensions and declination angles used in the attitude sweep of the approach analysis. Must be less than 16. Also used in midcourse fixed attitude analysis.

Number of trial retro firing times on approach analysis less than 20

## Program mode

0 Fly to TF or closest approach
1,2 Approach analyses
3 Midcourse analyses
4 MonteCarlo analyses
5 Midcourse verification mode
6 Lunar lifetime mode
7 Post-injection trim
Set to one for doppler analysis
Set to one if retro firing time analysis is to be output for each attitude

Thrusting mode
0 Tabular thrust / weight
2 Impulsive velocity
3 Hamilton Standard subroutine
Set to one for lunar orbit shadow. calculations during approach analysis.
Shadow calculation flag during trajectory propagation:
0 No shadows determined.
1 Shadow time determined by interpolation while numerically integrating trajectory.
2 Osculating orbit used to determine times.
Note: KSADOW $=1$ should not be used when averaging.

| LOCATION | FORTRAN <br> SYMBOL | USE | $\begin{aligned} & \text { PRESET } \\ & \text { VALUE } \end{aligned}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1050 | MCOUT | M | 0 | Extra output flag in midcourse analysis. <br> 1. Outputs targeted solution from PRƠTO <br> 2. Prints $\Delta V$, constraint errors for each iteration. <br> 3. Prints jet iteration information along with 2. |
| 1051 | JMC | M | 10 | Number of midcourse execution times simulated |
| 1052 | KMONTE | C | 2 | Monte Carlo description flag <br> 1. retro only <br> 2. midcourse and retro <br> 3. two midcourses and retro |
|  |  |  |  | If negative, the first correction will be calculated from the nominal assuming that the tracking data is good. |
| 1053 | KMAX | C | 50 | Sample size of Monte Carlo analysis. |
| 1054 | KSTART | C | 17 | Random number starter. |
|  |  |  |  | Approach analysis option flag. <br> KAPOPT $=1$ Firings made between an inpat range of true anomaly about perigee on the approach hyperbola. The attitude range is an input cone angle about the velocity vector at closest approach. |
| 1055 | KAPOPT | A | 1 | KAPOPT $=2$ Firings made from asymptote to asymptote on the approach hyperbola. The attitude range is input as initial right ascension and declination, increment in right ascension and declination, and number of attitude angles. <br> KAPOPT $=3$ Firings made between input times. The attitude range is input as in 2. |
| 1056 | NGRAPH | * |  | If positive, Element set number, for MAESTR graphics data base. <br> If negative, replace \|NGRAPH| element set. |
| 1057 | KREAD | A, V | - | Integer used to obtain initial conditions from a midcourse analysis. KREAD corresponds to the midcourse correction number desired. If KREAD is zero, this option is ignored and the initial state must be input. |
| 1058 | KOUT9 | I, L | ** | Auxiliary peripheral output unit number |


| LOCATION | FORTRAN SYMBOL | USE | PRESET VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
|  |  | - |  | Extra output flag. Used when KOUT9 $\neq 0$ |
| 1059 | KTERM | I, L | 0 | 0 . Orbital elements printed <br> 1. Position and velocity vectors |
| $\because$ | . |  |  | printed <br> 2. Both 0 and 1 |
| 1060 | INPWT | I | 1 | Input array write flag 0 no write |
|  |  |  |  | 1 write |
| 1061 | MCUNIT | A, M, V | 11 | Unit number of auxiliary midcourse output unit. |
|  |  |  |  | Miss vector option flag in Midcourse Analysis |
| 1062 | IBTR | M | 2 | 1. Use $B \cdot T$ and $B \cdot R$ loaded in $B V D$ <br> 2. Use $\operatorname{PSID}(1)$ and $\operatorname{PSID}(2)$ |
|  |  |  |  | Midcourse guidance law |
| 1063 |  |  |  | 1. Minimum fuel |
|  | KGLAW | M | 2 | 2. Fixed time of arrival |
|  |  |  |  | 4. Variable target energy |
|  |  |  |  | 5. Total fuel optimization |
| 1064 | NGROPT | M | 1 | Number of trials for which secant matrix is recomputed in Midcourse Analysis. |
| 1065 | NT | M | 10 | Number of trials allowed in Midcourse convergence. |
| 1066 | JET | M | 1 | If set, preliminary targeting will be done in the Midcourse Analysis. |
| 1067 |  |  |  | Limiting factor in the Midcourse Analysis. |
|  | MCLIM | M | 100 | The midcourse correction is limited to |
|  |  |  |  | MCLIM * DINK (KM/SEC) on each iteration. |


| LOCATION | FORTRAN <br> SYMBOL | USE | PRESET <br> VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1068 | NORD | I, L | 6 | Number of ordinates in averaging less than 16. |
| 1003 | nit | I, L | 3 | Number of intervals in averaging less than 16 |
| 1070 | KPROB | M | 95 | Output probability for midcourse execution error. Second midcourse execution time and errors are input through locations 440, 435, and 436. Negative or zero skips error propagation. |
| 1071 | IBURN | M | 6 | Trajectory propagation method during finite burn of midcourse motor. Impulsive calculations are used when set to zero. |
| 1072 | KROUT | M, C | 0 | Extra output flag during retro optimization calculations. |
| 1073 | KORECT | I | 0 | If set nonzero, the derivative at the end of an integration step will not be calculated. |
| 1074 | KMTOUT | C | 0 | Flag used to output initial state for each sample in a Monte Carlo analysis. |
| 1075 | KMETHP | M | 6 | Trajectory propagation method when generating partial derivatives. |
| 1076 | IFIND | A11 | - | Element set number of the anchor vector to be transferred from the differential correction program. |
| 1077 | KTF | M | 0 | Flag used to determine the type of Midcourse analysis. <br> $=0$ One-dimensional scan of midcourse execution times <br> $>0$ Two-dimensional scan of midcourse execution times and flight times. Flight times are scanned in KTF one-hour steps beginning at the desired flight time in location 422. <br> $<0$ Two-dimensional scan of midcourse. execution times and midcourse impulsive velocity. The impulsive velocity is centered about value, loaded into 426 and varied in -KTF steps of size DINK (location (479). |


| LOCATION | FORTRAN SYMBOI | USE | PRESET <br> VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1078 | IVTI | M | 0 | Overburn option key <br> 0 . in-plane retro antiparallel at periapsis, <br> $\pm 1$. variable inclination procedure approaching above or below desired inclination <br> $\pm 2$. variable periapsis procedure circularizing in-plane before or after periapsis. |
| 1079 | KHIGH | I | 0 | Farthest approach flag. If set, furthest approach will be found. |
| 1080 | NORMIN | M, C | 0 | Retro optimization flag <br> 0 in-plane at periapsis maneuver for underburns, according to IVTI for overburns <br> $\pm 1$. optimize retro to trim inclination in PROTO <br> 2. Same as 1, but in TARGET also. |
| 1081 | NREV | A | 0 | Number of revolutions the transfer orbit completes before stopping at closest approach in an approach analysis |
| 1082 | KAPWT | A | 0 | Flag used to write information from the approach analysis in an auxiliary unit. It is set to the output unit number when the option is desired. |
| 1083 | NAPUNT | P, L | 11 | Unit numbers where initial conditions are stored from a previous approach analysis on post-injection trim analysis. |
| 1084 | KSOLP | I | 0 | Solar pressure flag. <br> 0. no solar pressure <br> 1. pressure in radial direction only ( $\mathrm{s} / \mathrm{c}$ assumed to be sphere) <br> 2. spacecraft assumed to be a cylinder spinning along attitude vector. |
| 1085 | KCOV | C | 0 | Coordinate system of covariance matrix <br> 0. mean Equator and equinox of 1950 . <br> 1. local tangent plane |
| 1086 |  |  |  | Used in PEST version |
| 1087 | IDATT | all | 0 | Element set number of attitude file. Used when attitude is to be input via read from ADP. |


| LOCATION | FORTRAN SYMBOL | USE | PRESET | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| 1089 | NATUNT | al1 | 12 | Init number of attitude data, |
| 1089 | IDSAT | all | 1234567 | Satellite identification number. |
| 1090 | INIT | all | 0 | Initial line number of MAESTRO's space allotted on the director's display. If INIT is zero, no writes are made on the director's display. This option is only used during real-time operations. |
| 1091 | KPIT | P | 0 | Flag used to indicate post-injection trim targeting. |
| 1092 | ISET | all | 0 | Element set number when using subroutine PUTELS to transfer state to GTDS program. |
| 1093 | KPLOT | all | 0 | Set to the plot unit number when plotting. |
| 1094 | NMOD | I, L | 0 | Maximum number of zonals and |
| 1095 | NMOD | I, L | 0 | tesserals used to define the gravity field. |
| 1096 |  |  |  | Used in PEST version. |
| 1097 | KATMOS | I | 0 | Atmosphere drag flag. Set to 3 for Earth Drag. |
| 1098 | KASTRT | all | 0 | Flag used to initiate start-up procedure to obtain average elements from osculating elements. |
| 1099 | KTIST | I | 0 | Flag used to determine thrusters used in Hamilton Standard program <br> 0 both thrusters <br> 1 first thruster only <br> 2 second thruster only |

## COMMON INPUTS

## Description:

This common block contains the saved input array. This common block is required to stack cases.

## Length:

10008 -byte real words followed by
1004 -byte integer words
Subroutines Using:
MAIN INPUTF

This common block is essentially the same as INPUT common.

|  |  | COMMON INTER |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Description: |  | This common block contains quantities used in the interpolation logic of state propagation by numerical integration. |  |  |  |  |
| Length: |  | 130 8-byte words followed by 1 4-byte integer word |  |  |  |  |
| Subroutines Using: |  | CLOSE INTERP TIMEC UPDATE |  |  |  |  |
| LocationSymbolic <br> Name |  |  |  |  |  |  |
| 1 | $\mathrm{X}(10)$ | Table of times corresponding to the saved values |  |  |  |  |
| 11 | $\operatorname{POS}(6,10)$ | Array of saved values of the integration variables at the times corresponding to X . |  |  |  |  |
| 71 | $\operatorname{ACL}(6,10)$ | Array of derivatives of the integration variables at the times corresponding to X . |  |  |  |  |
| 131 | INT | Integer used to indicate the current value of the arrays X, POS and VEL. |  |  |  |  |

Description:

Length:
Subroutines Using:

| Location | Symbolic <br> Name |
| :--- | :---: |
| 2 | X |
| 8 | $\mathrm{Y}(6)$ |
| 8 | RATES(6) |

14

This common block contains the variables used in numerical integration

14 8-byte words

| MAIN | ARMPIT | AVEQNS | AVSTRT |
| :--- | :--- | :--- | :--- |
| CRASH | EQNS | FIELD2 | FOWARD |
| GRAV | INTEG | JET | LUNA |
| MOTORS | OUTPUT | OUT1 | PLANET |
| POST | RKSEVN | SETUP2 | SHADOW |
| SOL | TIMEC | TWELVE | UPDATE |

## Description

Independent variable of the numerical integration, usually time (sec)

Dependent variables of the numerical integration. The quantities depend upon the trajectory propagation technique in use.

Derivatives of the dependent variables with respect to the independent variable

Compute interval is used, same units as X.

## COMMON INTVRY



## COMMON MCCOM

Description:

Length:

Subroutines Using:
Location

1

2

3

4-5
6
XMC(6)

12
DV(3)

15
DVS(3)

18

19
BVD(2)
TMCS

XSUN(3)
150 8-byte words followed by
50 4-byte integer words

Symbolic
Name
ALIMITT

TR

PRX
-

This common block contains variables pertaining to the midcourse correction analysis.

| APROCH | BELL | BURND | FLXATG |
| :--- | :--- | :--- | :--- |
| JET | MCBURN | MCSET | MCVERF |
| MDCORS | MINTF | MONTE | POST |
| PROTO | SENSO | SETUP2 | TARGET |

位

## Description

Maximum allowable change in midcourse velocity on each iteration, $\mathrm{km} / \mathrm{sec}$.

Epoch of target planets state XS (location 32 in MCCOM common), seconds since state epoch.

Probability level of midcourse execution errors, percent.

Not used.
Midcourse pre-maneuver state. Cartesian position and velocity vectors in Earth mean equinox and equator of $1950, \mathrm{~km}$ and $\mathrm{km} / \mathrm{sec}$.

Midcourse impulsive velocity correction vector in Earth mean equator and equinox of $1950, \mathrm{~km} / \mathrm{sec}$

Spherical components of midcourse velocity DV. Magnitude, declination and right ascension, respectively, $\mathrm{km} / \mathrm{sec}$ and deg.

Midcourse correction execution time, seconds since state epoch.

Desired impact parameter vector at target planet, $\mathrm{B} \cdot \mathrm{T}$ and $\mathrm{B} \cdot \mathrm{R}$, respectively, kms.

Vector from the spacecraft to the Sun in Earth mean equator of 1950 .

| Location | Symbolic Name | Description |
| :---: | :---: | :---: |
| 24 | DVRA | Midcourse impulsive velocity oorrection on previous correction, $\mathrm{km} / \mathrm{sec}$. |
| 25 | DVRET | Impulsive velocity of retro motor, $\mathrm{km} / \mathrm{sec}$ |
| 26 | EXFUEL | Expected second midcourse fuel due to first midcourse errors (kg) |
| 27 | TV(3) | Unit vector along the spin-axis at retro maneuver time (true equator and prime meridian of target body) |
| 30 | DVT | Required post-injection trim velocity, $\mathrm{km} / \mathrm{sec}$. |
| 31 | FUELT | Fuel required for the post-injection trim maneuver, kg. |
| 32 | $\mathrm{XS}(6)$ | Cartesian target planet state resulting from current midcourse maneuver in true equator and prime meridian of target planet, km and $\mathrm{km} / \mathrm{sec}$. |
| 38 | PFAC | Midcourse variable target energy guidance constant. Set to |
|  |  | PFAC $=$ ASPR * WRETRO/ASPMC |
|  |  | where |
|  |  | ASPR is the retro motor's specific impulse, sec. <br> ASPMC is the midcourse motor's specific impulse, sec. <br> WRETRO is the retro motor's fuel, kg . |
| 39 | DJDIF | Time from liftoff to state epoch, sec. |
| 40 | SIGOUT(1) | Time, sec. |
| 41 | SIGOUT(2) | Hyperbolic excess speed, errors at <br> $\mathrm{km} / \mathrm{sec}$  |
| 42 | STGOUT(3) | $\left.\begin{array}{l}\begin{array}{l}\text { Arrival circular excess } \\ \text { speed, } \mathrm{km} / \mathrm{sec}\end{array}\end{array}\right\}$target planet due <br> to midcourse |
| 43 | SIGOUT(4) | Required post injection trim fuel, kg . |
| 44 | SIGOUT(5) | Radius, km errors. |
| 45 | SIGOUT(6) | Inclination, deg. |


| Location | Symbolic Name | Description |
| :---: | :---: | :---: |
| 46 | FFIRE | True anomaly of retro-fire on the approach hyperbola (rad) |
| 47 | WTF | Spacecraft weight after midcourse correction, kg. |
| 48 | TMC2 | Time of second midcourse, seconds since state epoch. |
| 49 | EXV2 | Expected magnitude of second midcourse correction, $\mathrm{km} / \mathrm{sec}$ |
| 50 | $\operatorname{DPT}(3,10)$ | Partial derivative matrix. This matrix represents the partial derivative of the output variables with respect to the midcourse correction velocity. The order of the output quantities is the same as the PSI Array starting in location 100. |
| 80 | PSID(1) | Radius, km. |
| 81 | $\operatorname{PSID}(2)$ | Inclination, deg. Desired end conditions |
| 82 | PSID(3) | Time of flight, sec. at target planet. Array |
| 83 | PSID(4) | Hyperbolic excess speed,$\mathrm{km} / \mathrm{sec}$.$\quad$is ten long, only six <br> are currently in use. |
| 84 | PSID(5) | Circular excess speed, $\mathrm{km} / \mathrm{sec}$. |
| 85 | $\operatorname{PSID}(6)$ | Total fuel, kg. |
| 86 | PSID(7) | Central velocity value for fixed- attitude guidance scan (km/sec) |
| 90 | TOL(1) |  |
| 91 | TOL(2) | $B \cdot R, \mathrm{~km}$. Convergence tolerance |
| 92 | TOL(3) | Time of flight, sec. $\quad$ on end conditions. Only |
| 93 | TOL(4) | Hyperbolic excess speed $\mathrm{km} / \mathrm{sec}$$\quad\left\{\begin{array}{l}\text { six locations of a ten } \\ \text { long array are currently }\end{array}\right.$ |
| 94 | TOL(5) | Circular excess speed in use. <br> $\mathrm{km} / \mathrm{sec}$.  |
| 95 | TOL(6) | Minimum gain in total fuel, kg . |



| Location | Symbolic Name | Description |
| :---: | :---: | :---: |
| 163 | NGROPT | Number of trials to recompute secant matrix |
| 164 | KGLAW | Guidance law |
|  |  | $=1$ Minimum fuel |
|  |  | 2 Fixed time of arrival |
|  |  | 3 Fixed target energy |
|  |  | 4 Variable target energy |
|  |  | 5 Minimum total fuel |
| 165 | ICB | Central body number of XMC state in location 6. |
| 166 | ISP | Key to determine if gradient has been computed. |
| 167 | IBTR | Miss vector option key |
|  |  | $=1 \mathrm{~B} \cdot \mathrm{~T}$ and $\mathrm{B} \cdot \mathrm{R}$ |
|  |  | 2 Radius of closest approach and inclination |
| 169 | IPD(3) | Constraint indicator array - constraints pointed out by this array are tested against tolerances for convergence. |
| 172 | KEL(10) | Array of elevation angles of the spacecraft at the midcourse execution time. The array corresponds to observation sites $1-10$, (deg / 10) |
| 182 | KELR(10) | An array similar to KEL except at the retro firing time. |

## COMMON MOON





## COMMON PIT



## COMMON PLNET

Description:

Length;
Subroutines Using:
Location

1 $\frac{$|  Symbolic  |
| :---: |
|  Name  |}{$\operatorname{XP}(6,12)$}

85

89-100

This common block contains quantities that describe the position and velocity of the planets with respect to the central planet.

100 8-byte words

| MAIN | ACCEL | APROCH | ARMPIT |
| :--- | :--- | :--- | :--- |
| AVERGE | CLOSE | CRASH | DRAG |
| FIXATG | GRAV | JET | LUNA |
| MCVERF | MDCORS | MULCON | OBLE |
| OUTPUT | OUT1 | POST | PRINT |
| READE | SETUP2 | SHADOW | SOL |
| SOLP | TARGET | UPDATE | VISIB |

## Description

Position and velocity vectors of the planets with respect to the central planet in Earth mean equator and equinox of $1950, \mathrm{~km}$ and $\mathrm{km} / \mathrm{sec}$. The first index denotes the vectors while the second index denotes the planet number. The order of the planets is,

| 1 | Mercury | 6 Jupiter | 9 | Pluto |
| :--- | :--- | :--- | :---: | :--- |
| 2 | Venus | 6 | Saturn | 10 |
| Sun |  |  |  |  |
| 3 | Earth | 7 Uranus | 11 Moon |  |
| 4 | Mars | 8 Neptune | 12 | Oddball |

Distance from the central planet to the planets, except that DST (JC) is distance to spacecraft where JC is central planet number.
Earth nutation variables. Only available when tape ephemeris is used.

Not used

|  |  | COMMON SAVE |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Description: |  | This common block is principally used to save various quantities for use in restoring the state. |  |  |
| Length: |  | 45 8-byte words |  |  |
| Subroutines Using: |  | MAIN <br> PRINT | CRASH FOWARD <br> SETUP2 TIMEC | OUT1 |
| Location | Symbolic Name | Description |  |  |
| 1 | SAVE(6) | Saved position and velocity vectors in mean equator and equinox of $1950, \mathrm{~km}$ and $\mathrm{km} / \mathrm{sec}$ |  |  |
| 7 | TSAV |  | Epoch of saved variables, time si | epoch, sec |
| 8 | SAVE1(6) |  | Saved integration variables at time to TSAV. | sponding |
| 14 | SAVE2(14) |  | Saved Encke's reference orbit, mea epoch of reference orbit. | tion and |
| 28 | SRATES(6) |  | Saved derivatives of the integration at TSAV. | bles |
| 34-39 | - |  | Not used |  |
| 40 | TOUTL |  | Last time trajectory output was obta numerical integration, sec. since st | during poch. |
| 41 | RSAV |  | Last sine of the flight path angle with planet. Used during closest approa numerical integration. | pect to target ration of the |

## COMMON SHAD




| Location | Symbolic Name | Description |
| :---: | :---: | :---: |
| 20-25 | - | Not used |
| 26 | EJO | Modified ephemeris date of state epoch, days since 2400000 julian date. |
| 27 | TRU | Contains true or mean anomaly according to the setting of the METH flag in location 1013 of INPUT common. $\begin{array}{ll} \text { METH }=4,5 & \text { mean anomaly } \\ \text { METH }=8 & \text { true anomaly plus argument } \end{array}$ |
| 28 | EJT | Current ephemeris time, days since 2400000 julian date. Also valid at intermediate times of integration. |
| 29 | TCA | Time of closest approach, seconds since state epoch. |
| 30-31 | - | Not used |
| 32 | UJT | Current modified julian date, days since 2400000. Also valid at intermediate times of integration. |
| 33 | THRUST | Engine thrust, newtons. |
| 34 | WT | Spacecraft mass at engine ignition, kg. |
| 35 | W | Current spacecraft mass, kg. |
| 36 | SOL | Solar pressure constant. <br> $=$ solar pressure nea $x$ solar pressure at 1 au from the sun $x a u^{2} \times 1(10)^{-8}$. |
| 37 | ACL | Magnitude of thrust acceleration (km/sec ${ }^{2}$ ) |
| 38-40 | - | Not used |

## SECTION 5

## Input/Output Units

This section describes the purpose of the input/output units required for the operation of MAESTRO. A description of these units is shown in Table 5.1. Some of the unit numbers are program inputs. If the unit number is greater than 1000 , the number under unit number specifies the location in the input array where the unit number is to be input.

TABLE 5.1
Input/Output Units

| Unit <br> Number | Record <br> Length (Bytes) | Subroutine <br> Used | Purpose |
| :---: | :---: | :---: | :---: |
| 1 | 3512 | FIND | Directory of the GTDS 24 -hour hold file. Required when input input location 1076 is non-zero. |
| 5 | 36 | INPUTF | Program data inputs. Unit required for all operations. |
| 6 | variable | Many routines | Program primary output device. Unit required for all operations. |
| 10 | 7452 | GETTAP | Ephemeris disk or tape. Direct read used with disk. Input location 1017 used to specify whether disk or tape. Unit required when 1017 equals 3 or 5 . |
| 12 | 280 | SETUP2 | Attitude input unit. Used to pass attitude from Attitude Determination Program to MAESTRO. Required when input location 1087 is non-zero. |
| 13 | 80 | MAIN | Director's display. Used during in-flight operations. Required when input location 1090 is non-zero. |
| 21 | 64 | POST, PROTO | Plot Unit. Used to write tape for post processing. Required when input location 1093 is set to 21 . |
| 26 | 7112 | FIND | State input unit. Used to pass state from GTDS to MAESTRO via the 24-hour hold file. Required when input location 1076 is non-zero. |
| 27 | 7200 | PUTELS | State output unit. Used to transfer state to GTDS program. Required when input location 1092 is non-zero. |


| Unit <br> Number | Record <br> Length (Bytes) | Subroutine <br> Used | OUTPUT <br> PROTO <br> POST |
| :--- | :--- | :--- | :--- |
| variable | POST <br> PROTO | Auxiliary output unit. Required <br> when input location 1058 is non- <br> zero. |  |
|  | 360 |  | Midcourse output unit. Unit <br> number in input location 1061 must <br> be specified when the midcourse <br> analysis mode is used. |

* The unit number is a program input. The number shown is the input location where the unit number is input.


## SECTION 6

## SUBROUTINE DESCRIPTION

This section presents a description of the subroutines which comprise the MAESTRO program. The descriptions include

1. Calling sequence
2. Purpose
3. Common Blocks Required
4. Subroutines Required
5. Input / Output descriptions
6. Theory
7. Description
8. Flow chart where applicable

## MAIN PROGRAM


#### Abstract

Purpose: The main program initializes some common blocks and calls the subroutine (s) which perform the desired MAESTRO analysis.


Common Blocks Required: AVG, CETBL2, CETBL3, CNTRL, CONST, FIELDM, INPUT, INPUTS, INTVAR, PERT, PLNET, SAVE, STATE

Subroutines Required:
APROCH, ARMPIT, CONTRL, FOWARD, MCSET, MCVERF, MONTE, PROTO, INPUTF

Input / Output

| I/O | SYMBOLIC |  |  |  |
| :--- | :---: | :---: | :--- | :--- |
| NAME | DIMENSION | COMMON |  |  |
| 0 | BLOCK | DEFINITION |  |  |
| ICW | 1 | CETBL2(1) | Tape ephemeris error flag |  |
| I | INIT | 1 | INPUT(1090) | Flag used to write <br> director's file |
| I | MODE | 1 | INPUT(1044) | MAESTRO analysis flag |
| $O$ | TAB3 | 829 | CETBL3(1) | Raw ephemeris tape data |

## Description:

The main program initializes some common blocks and calls the appropriate subroutines to perform the desired MAESTRO analysis. No computations are done in this routine. MAIN is also used to define the length of many of the common blocks.

Initially common blocks CNTRL, PLNET and STATE are cleared. Next subroutine INPUTF is called to initialize the input array from program inputs. Writes on unit 13 are made if INIT is not equal to zero. These writes are messages which are displayed on a file known as the director's file and are only used in real-time operations. Next, the saved common is cleared and subroutines CONTRL and MCSET are called to initialize variables for later use. Finally, subroutines APROCH, MONTE, MCVERF, ARMPIT or PROTO are called according to the setting of the MODE flag. These subroutines are used to control the logic for one of the MAESTRO analysis modes. After the analysis is complete, flow returns to the call to INPUTF to initiate the next case.

## SUBROUTINE ACCEL

## Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required:

## CALL ACCEL

This subroutine evaluates the acceleration of the spacecraft.

CNTRL, INPUT, GRAVTY, PERT, PLNET
DVMAG, FIELD2, GRAV, MOTORS, OBLE, PLANET, SOLP

Input / Output

| SYMBOLIC |  |  |  |
| :--- | :---: | :---: | :---: | :--- |
| I/O | NAME | DIMENSION | COMMON |
| BLOCK |  |  |  |$\quad$ DEFINITION

## Description:

This subroutine controls the logic to determine the spacecraft's acceleration. Initially, the acceleration in RCART is cleared. Next, subroutine PLANET is called to obtain the current position of the planets. Subroutines OBLE, FIELD2, MOTORS and SOLP are called to evaluate the acceleration due to Earth oblateness, general gravitational field, engine firing and solar pressure. These subroutines are called only if input flags are set. Finally subroutine GRAV is called to determine the acceleration due to the planets in the system and the central planet if the Cowell trajectory propagation scheme is used.


## STBROTTTNE APROCH

## Calling Sequence:

Purpose:

CALL APROCH
This subroutine performs the analysis to determine the retromotor's firing time and attitude

Common Block Required:

Subroutines Required: CROSS, FOWARD, M50MDT, NUTATE, ORBIT, QUIKIE, PUTELS, TRIM, TRMN, VNORM, VIEW, VISIB

Input / Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINTTION |
| :---: | :---: | :---: | :---: | :---: |
| O | TF | 1 | INPUT(4) | Final time |
| I | DJI | 1 | INPUT(37) | Launch epoch |
| I | WO | 1 | INPUT(38) | Initial S/C mass |
| I | DJO | 1 | INPUT(46) | State epoch |
| 0 | RTAC | 1 | INPUT(47) | Right ascension in retro motor altitude |
| 0 | DECL | 1 | INPUT(48) | Declination of retro motor altitude |
| I | HRO | 1 | INPUT(53) | Hour of state epoch |
| I | XMINO | 1 | INPUT(54) | Minutes of state epoch |
| I | SECO | 1 | INPUT(55) | Seconds of state epoch |


| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | TCOMP | 10 | INPUT(170) | Switching times of compute interval table |
| O | DELT | 10 | INPUT(180) | Compute intervals |
| 0 | TIG | 1 | INPUT(381) | Retro motor ignition time |
| 0 | TBO | 1 | INPUT(384) | Retro motor burnout time |
| I | TFIRE1 | 1 | INPUT(400) | Initial retromotor firing time |
| I | TFIRE2 | 1 | INPUT(401) | Final retro motor firing time |
| 1 | RAO | 1 | INPUT(403) | $\qquad$ |
| I | DECO | 1 | INPUT(404) | Initial declination of retro |
| I | DELRA | 1 | INPUT(405) | Increment in right ascension |
| I | DELDEC | 1 | INPUT(406) | Increment in declination |
| I/O | DELV | 1 | INPUT(407) | Retro motor's impulsive velocity |
| I | ASPMC | 1 | INPUT(441) | Midcourse motor's ISP |
| I | ASPR | 1 | INPUT(442) | Retro motor's ISP |
| I | WRETRO | 1 | INPUT(443) | Mass of retro fuel |
| I | WDROP | 1 | INPUT(473) | Retro drop mass |
| I | CONE | 1 | INPUT(474) | Altitude cone angle |
| I | TRUE | 1 | INPUT(475) | True anomaly firing range |


| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | DTMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | RBURN | 1 | INPUT(512) | Retro motor burn time |
| O | TCATST | 1 | INPUT(513) | Time to begin closest approach testing |
| I | JRA | 1 | INPUT(1041) | Number of trial right ascensions |
| I | JDEC | 1 | INPUT(1042) | Number of trial declinations |
| I | KFIRE | 1 | INPUT(1043) | Number of firing times |
| I. | MODE | 1 | INPUT(1044) | Program mode flag |
| I | KAPOUT | 1 | INPUT(1046) | Firing analysis output flag |
| I | KAPSAD | 1 | INPUT(1048) | Lunar orbit shadow flag |
| I | KAPOPT | 1. | INPUT(1055) | Retro analysis option flag |
| I | KREAD | 1 | INPUT(1057) | Midcourse firing number when initial conditions from midcourse analysis |
| I | MCVNIT | 1 | INPUT(1061) | Midcourse unit number |
| I | NREV | 1 | INPUT(1081) | Number of transfer trajectory revolutions |
| I | KAPWT | 1 | INPUT(1082) | Unit number where approach firings are saved |
| I/O | X | 3 | STATE(1) | S/C position vector |
| 1/O | DX | 3 | STATE (4) | S/C velocity vector |

Input / Output

| 1/0 | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSION | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I/O | T | 1. | STATE(10) | Time since state epoch |
| 0 | ATT | 3 | STATE(11) | Unit vector along S/C altitude |
| 0 | ELMD | 6 | STATE(14) | S/C orbital elements |
| I | UJT | 1 | STATE(32) | Current modified julian date |
| I | WT | 1 | STATE (34) | S/C mass prior to motor ignition |
| I | JC ${ }^{\text {- }}$ | 1 | CNTRL(7) | Central planet number |

## Description:

This subroutine is used to determine the retro firing time and altitude in order to meet specific mission criteria. This version of subroutine APROCH was designed for the RAE-B mission. The most important criterion for this mission is the post injection trim fuel requirement. Other constraints are the spin axis-sun angle at firing, tracking station coverage, shadow times and orbit orientation. The retro conditions are determined in a brute-force manner. Repeated trial retro firings are made at various altitudes and firing times. The mission constraints are determined for each firing and output in displays so that the user can determine the firing time and altitude.

The retro motor can be simulated numerically or with the impulsive velocity approximation. Also, the pre-retro state can be determined numerically or with conic approximations. The MODE flag is used to determine which approximations apply. If the MODE flag is set to one, the entire analysis is performed using numerical techniques. The impulsive velocity and conic approximations are used when the MODE flag equals two.

The first part of the subroutine sets up constants and has logic to pick up the initial conditions from the midcourse tape if desired. The KREAD flag is used for this option. Logic is also included to set the closest approach test start time , TCATST, if more than one orbit of the transfer trajectory is desired. This option was included for Earth orbit missions and does not apply to a RAE-B type of mission. NREV should equal zero. At this point the logic flow divides according to the type of analysis desired - approximate or precise.

Let's discuss the approximate case first. When MODE $=2$, the closest approach flag is set to 2 to cause the propagator to stop at closest approach. Next, subroutine FOWARD is used to propagate the state to the closest approach of the target planet. The coordinate rotation is determined from the integration frame (Earth mean of 1950) to the true equator of the target planet. Subroutines M50MDT and NUTATE are used. The state at the closest approach is rotated to the desired system using MVTRN and orbital elements determined using subroutine ORBIT. These ele$m$ ents are the elements of the approach hyperbola。

The firing time on the approach hyperbola and altitude are varied in one of three ways through the KAPOPT flag as follows:

## KAPOPT $=1$

The true anomaly of the firing is determined from

$$
f_{i}=\left(f_{o}-\text { TRUE }\right)+\sum_{i=1} \quad i *(2 \text { TRUE } / \mathrm{KFIRE})
$$

where
$f_{0}$ is the true anomaly at closest approach
TRUE is the input true anomaly range
KFIRE is the input number of firings

The attitude is determined by

$$
\begin{aligned}
& \mathrm{ra}_{\mathrm{i}}=(\mathrm{RAO}-\mathrm{CONE})+\sum_{\mathcal{H}=1}^{\mathrm{JRA}} \mathrm{i} *(2 \mathrm{CONE} / \mathrm{JRA}) \\
& \mathrm{DEC}_{i}=(\mathrm{DECO}-\mathrm{CONE})+\sum_{i=1}^{\mathrm{JDEC}} \quad \mathrm{i} *(2 \mathrm{CONE} / \mathrm{JDEC})
\end{aligned}
$$

Where
RAO is the input starting right ascension
DECO is the input starting declination
JRA is the input number of right ascensions
JDEC is the input number of declinations
CONE is the input altitude cone angle
If the starting right ascension and declination are not input (equal to zero), the right ascension and declination of the velocity vector at closest approach will be used。

## KAPOPT=2

The starling increment and number of abtitudes must be input. Then the altitude is determined from

$$
\begin{aligned}
\mathrm{RA}_{\mathrm{i}}= & \mathrm{RAO}+\sum_{\mathrm{i}=1}^{J R A} \quad \mathrm{i} * \text { DELRA } \\
& \quad \mathrm{JDEC}^{\mathrm{JDE}}=\mathrm{DECO}+\sum_{i=1} \quad i * \text { DELDEC }
\end{aligned}
$$

The firing true anomaly is determined in a manner similar to the KAPOPT=1 option except that the range in true anomaly is from asymptote to asymptote instead of CONE.

## $\underline{K A P O P T=3}$

The first and last firing point on the approach hyperbola is input via the firing time since liftoff. The firings are made at KFIRE number of constant true anomaly steps between the bounds specified by the times. The altitude is varied as described for KAPOPT=2.

The IF statement just before statement 8 is used to transfer to the logic which sets up the necessary constants to increment the true anomaly and altitude. The logic flow for the precise case, $\mathrm{MODE}=1$, differs slightly from the case discussed above. Instead of flying to closest approach to the target planet, the final time is set to the first firing time, TFIRE1. The KAPOPT flag must be set to 3 for this mode; thus, flow transfers to Statement 11.

At this point, DO LOOPS are established to increment the right ascension, declination and firing time. The range of these loops extends to almost the end of the subroutine. The mission constraints are calculated for each attitude and firing time and saved in arrays for output or output inside the loops. When the $\mathrm{MODE}=2$ is used,' the position and velocity vectors are determined from subroutine ORSIT where the true anomaly is determined by the DO 10 loop. The impulsive velocity of the retro is added to the spacecraft's velocity in the direction specified by the attitude. This calculation is performed at statement 13. Next, the post retro orbit is determined and desired mission constraints calculated using subroutines QUIKIE and TRIM. The firing time display is output if the KAPOUT is set.

Subroutine FOWARD is used to simulate the retro motor when MODE $=1$. When this case is desired, subroutine FOWARD is first called to propagate the spacecraft to the ignition time, next, the compute interval is adjusted and the engine ignition and burnout times set. Finally, FOWARD is used to propagate through the burn. This logic is located between statements 100 and 110. Flow then transfers to the same place where the post retro orbit is determined when $\mathrm{MODE}=2$.

The above logic is repeated until all attitudes and firing times are simulated. When the DO 30 loop is completed, the minimum eccentricity and trim fuel grids are output. This completes the subroutine.


Rotate to lunar equator and




## SUBROUTINE ARMPIT

Calling Sequence:

Purpose:

Common Block Required:

Subroutines Required:

CALL ARMPIT
This subroutine determines the post injection time requirements and outputs the post injection trim displays.

CONST, INPUT, INTVAR, PIT, PLNET, STATE

M50LEQ, ORBIT, PUTELS, ROTATE, TRIM, TRMN, TWOPIT

Input/Output

| 1/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | PROGRAM DIMENSION | $\begin{aligned} & \hline \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | CID | 1 | INPUT(446) | Inclination of desired final orbit w.r.t. target planet equator. |
| I | DJL | 1 | INPUT(37) | Modified Julian launch date |
| I | DJO | 1 | INPUT(46) | Modified Julian date of epoch |
| O | CIF | 1 | PIT (1) | Cosine of desired inclination |
| I | DV1 | 3 | PIT (3) | First trim velocity maneuver |
| I | DV2 | 3 | PIT (6) | Second trim velocity maneuver |
| I | IDSAT | 1 | INPUT(1089) | Satellite ID number |
| I | ISET | 1 | NPPUT(1092) | Element set number when state is saved from GTDS |
| I | JT | 1 | INPUT(1031) | Target planet number |
| I | RD | 1 | INPUT(444) | Radius of final orbit |
| I | TRINC | 1 | NPUT(449) | Inclination of the transfer orbit |
| 0 | SIF | 1 | PIT(2) | Sire of desired inclination |

Input/Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | PROGRAM DIMENSTON | COMMON BLQNK | DEELNTTLION |
| :---: | :---: | :---: | :---: | :---: |
| I | UJT | 1 | STATE(32) | Current modified Julian date |
| I | X | 6 | STATE(1) | Position and velocity of spacecraft |
| I | XP | 6, 12 | PLNET(1) | Positions \& velocities of the planets |
| I | XS | 6 | PIT(12) | Position and velocity of spacecraft just prior to first maneuver |

## Description:

This subroutine determines the post injection maneuver required to achieve a circular orbit with a desired radius and inclination. Two maneuvers are determined. First, a two impulse maneuver with a plane change and, second, a Hohmann transfer maneuver with no plane change. The maneuvers are determined in subroutines TWOPIT and TRIM, respectively.

The initial orbital elements are determined using subroutine ORBIT and supplied as inputs to subroutines TRIM and TWOPIT. The calculated maneuver is passed back to ARMPIT via DV1 and DV2 of PIT common. Also the state just prior to the first maneuver is passed back in XS of PIT common. The first maneuver is added to the state in XS to determine the transfer orbit. Subroutine ORBIT is used to determine the transfer orbit's elements, ELMT. A 180 degree transfer is specified. Thus, the true anomaly of the second maneuver on the transfer orbit is,

$$
\mathrm{F}=\operatorname{ELNT}(3)+\pi
$$

The cartesian state is determined at this true anomaly, the second maneuver applied, and the final orbit, ELMF, determined using ORBIT. The desired output quantities are determined and the appropriate displays presented.

The above logic is surrounded by a loop using the flag ISTEP. The purpose of the loop is to pass through the calculation of the orbits once for the plane change maneuver and a second time for the Hohmann transfer maneuver.

The state after the first trim can be written as a hold file for retrieval by the GTDS Program. The ISET flag is used to key this option. Subroutine PUTELS is used to write this file.


Calling Sequence:
Purpose:

CALL $\operatorname{ATMO}(J C, N, H)$
To compute the atmospheric density for drag calculations.

Common Blocks Required: None
Subroutines Required: None
Inputs/Outputs

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :---: | :---: | :---: | :--- | :--- |
| I | JC | 1 | Call List | Central body number |
| I | N | 1 | Call List | Atmospheric model number |
| I | H | 1 | Call List | Altitude |

## Discussion:

Atmospheric density is computed as an exponential function of altitude for Mercury, Venus, Earth, or Mars when altitude is below $50,000 \mathrm{~km}$. The density data is tabulated as a function of altitude.

$$
\rho=\exp \left[\rho_{k-1}+s\left(\ln h-\ln h_{k-1}\right)\right]
$$

where

$$
h_{k-1} \leq h<h_{k}
$$

and

$$
s=\left(\rho_{k}-\rho_{k-1}\right) /\left(\ln h_{k}-\ln h_{k-1}\right)
$$

## Calling Sequence:

Purpose:

## CALL AVEQNS

To average, by Gaussian quadrature integration, the planetary equations over one revolution of the satellite in its orbit.

Common Blocks Required: INPUT, CNTRL, STATE, AVG, INTVAR, PERT, CONST, GRAVTY

Subroutines Required: TRMN, ORBIT, ACCEL, ROTATE

| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I/O | NAME | DIMENSTON | BLOCK | DEFINITION |
| I | METH | 1 | INPUT(1013) | Trajectory propagation method |
| I | NORD | 1 | INPUT(1068) | Number of ordinates to be used for each interval of Gaussian quadrature integration |
| I | INT | 1 | INPUT(1069) | Number of intervals into which the orbit is divided for Gaussian quadrature integration |
| I | DJ 0 | 1 | INPUT(46) | Modified Julian date at liftoff epoch |
| I | EJ 0 | 1. | STATE(26) | Modified ephemeris date at liftoff epoch |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | PI2 | 1 | CONST (3) | Twice pi |
| I | GM | 12 | CONST (5-16) | Gravitational constants of the planets |
| I | X | 1 | INTVAR(1) | The current time (independent integration variable) |
| I | Y | 6 | INTVAR(2-7) | The current array of dependent integration variables |
| I | WEIGHT | 78 | AVG (1-78) | Weights for Gaussian quadrature formula |
| I | ABSCIS | 78 | AVG (980156) | Abscissa for Gaussian quadrature formula |
| $\emptyset$ | RATES | 6 | INTVAR(8-13) | The time derivatives of the current dependent integration variables |

Input/Output

|  |  |  |  |  |
| :--- | :---: | :---: | :--- | :--- |
| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINTTION |
| I | PI | 1 | CONST(2) | Pi |
| I | F | 1 | STATE(27) | Mean or true anomaly plus <br> argument of perigee |
| 0 | EJT | 1 | STATE(28) | Current ephemeris date |
| $O$ | UJT | 1 | STATE(32) | Current modified julian date |

## Theory:

Reference 1 gives a description of the theoretical basis for the numerical averaging. The planetary equations are written in Gauss's form where the perturbations enter directly as accelerations rather than as the partial derivatives of a potential function. The planetary equations exist in several forms in the program. They are symbolically represented by

$$
\begin{equation*}
\dot{E}_{i}=f_{i}\left(E_{j}, t\right) \quad i, j=1,6 \tag{1}
\end{equation*}
$$

where the $E_{i}$ represent a set of 6 independent osculating orbital elements.
The averaging operation is accomplished numerically rather than analytically and is represented by

$$
\begin{equation*}
\frac{\dot{\mathrm{F}}_{\mathrm{i}}}{}=\frac{1}{\tau} \int_{\mathrm{t}-\tau / 2}^{t+{ }^{\tau} / 2} \mathrm{f}_{\mathrm{i}} \mathrm{dt} \tag{2}
\end{equation*}
$$

where $\tau$ is the orbital period. The mean values of the orbital elements are defined by

$$
\begin{equation*}
\overline{\mathrm{E}}=\frac{1}{\tau} \int_{\mathrm{t}-}^{\tau / 2} \mathrm{E}(\mathrm{t}) \mathrm{dt} \tag{3}
\end{equation*}
$$

The important thing to note is that the numerical averaging method essentially trades the set of 6 differential equations for the osculating orbital elements for a set of 6 differential equations for the mean orbital elements.

The mechanical averaging is done with respect to true anomaly rather than with the time directly. The Keplerian relation

$$
\dot{\mathrm{f}}=\frac{\sqrt{\mu \mathrm{p}}}{\mathrm{r}^{2}}
$$

is used to transform the integrals in (2) from time to true anomaly giving

$$
\begin{equation*}
\overline{\dot{E}}_{i}=\frac{n p^{2}}{2 \pi \sqrt{\mu p}} \int_{f\left(t-{ }^{\tau} / 2\right)}^{\rho^{f(t+\tau / 2)}} \frac{f_{j}\left(E_{j}, f\right)}{(1+e \cos f)^{2}} d f \tag{4}
\end{equation*}
$$

where $\mathbf{f}$ is the true anomaly and the other symbols represent Keplerian elements and constants as given in Reference 1.

Description:
This subroutine is a modified version of a variable-order, variable interval Gaussian quadrature integrator. The weights and abscissae for the quadrature formula are stored sequentially in AVG common, in the arrays called WEIGHT and ABSCIS. The values stored correspond to quadrature formulae of order 2-16 as given in reference 2. The single weight required for a two-point quadrature is stored in WEIGHT(1), the two weights required for a three point quadrature in WEIGHT(2) and WEIGHT(3), and the two weights required for a four point quadrature in WEIGHT(4) and WEIGHT(5). The pattern continues up through and including the eight weights required for a 16 point quadrature stored in WEIGHT (64-71). The corresponding abscissae are stored in the corresponding locations of the ABSCIS array.

The order (N $\varnothing$ RD) of the quadrature is specified by the user as well as the number of subintervals (INT) to be used in the integration. The full range of the independent variable (true anomaly) is divided into INT intervals and the current (mean) value of the true anomaly is calculated from the mean mean anomaly and stored. The coding near statements 50 and 51 establishes the proper limits of integration and performs the transformation from mean anomaly to true anomaly for either the Method 5 or Method 8 variables. Throughout the subroutine, appropriate tests are made to distinguish between the two methods.

Certain quantities that are calculated only once are established before the quadrature summation begins at the DO 507 loop. This outer loop is for the summation of the subinterval quadrature formula. The actual Gaussian quadrature algorithm starts at statement

150 where the time and the sine and cosine of the true anomaly are calculated for each required evaluation of the integrands for the 6 differential equations to be numerically averaged. The values of the integrands are calculated as shown in reference 1, multiplied by the appropriate weight, and the results are stored in the array SUM.

The results of each subinterval quadrature are accumulated in the array called ANS until all subintervals are completed at statement 507. The 6 stored integrals are then multiplied by the constants outside the integral signs of the averaged equations and the results are loaded into the RATES array of INTVAR common for use in the numerical solution of the averaged differential equations of motion.

## References

1. Uphoff, C., Numerical Averaging in Orbit Prediction, Presented at AIAA/AAS Astrodynamics Conference, Palo Alto, Calif., AIAA preprint No. 72-934, September 1972.
2. Abramowitz, Milton and Stegun, Irene A., Handbook of Mathematical Functions, published by The National Bureau of Standards, Applied Mathematics Series No. 55, May 1968.

Calling Sequence:
Purpose:

CALL AVERGE (KSET)
AVERGE calculates the average perturbing acceleration due to the Sun and the Moon's indirect term over one step of the multiconic trajectory scheme.

Common Block Required: DUM, STATE, PLNET, CONST
Subroutines Required: DVMAG

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| I/O | A1S | 3 | DUM(1) | Earth-Moon acceleration <br> at beginning of step |
| I/O | A2S | 3 | DUM(4) | Sun acceleration at <br> beginning of step |
| I | DT | 1 | DUM(10) | Step time interval |
| I | GM | 12 | CONST(5) | Gravitational constants <br> of planets |
| I | KSET | 1 | CALLING <br> OPERAND | Flag for adding or subtracting <br> accelerations |
| I/O | X |  | STATE(1) | Position and velocity of <br> spacecraft |
| I | XP | 72 | PLNET(1) | Positions and velocities of <br> planets wrt central planet |

## Theory:

AVERGE corrects the position and velocity of the spacecraft during each step in order to account for perturbing accelerations that are not included in the conic assumptions.

These perturbing accelerations are caused by the Sun and also by the effect of the Moon on the planet. Let $\overline{A 1}$ be the acceleration vector due to the EarthMoon term at the present time, and $\overline{A 1 S}$ be the same vector from the previous time (T-DT). Similarly let $\overline{\mathrm{A} 2}$ and $\overline{\mathrm{A} 2 \mathrm{~S}}$ be the Sun term accelerations corresponding to T and $\mathrm{T}-\mathrm{DT}$.

Then $\quad \overline{\mathrm{A} 1}=\mathrm{GM}_{\mathrm{m}} * \overline{\mathrm{R}}_{\mathrm{em}} /\left|\overline{\mathrm{R}}_{\mathrm{em}}\right|^{3}$
And $\quad \overline{\mathrm{A} 2}=\mathrm{GM}_{\mathrm{s}} *\left(\overline{\mathrm{R}}_{\mathrm{es}} /\left|\overline{\mathrm{R}}_{\mathrm{es}}\right|^{3}+\overline{\mathrm{R}}_{\mathrm{sV}} /\left|\overline{\mathrm{R}}_{\mathrm{sV}}\right|^{3}\right)$
Where subscripts m, e, s, svrefer to Moon, Earth, Sun and space vehicle respectively and $\overrightarrow{\mathrm{R}}_{\mathrm{es}}$ is the vector from the Earth to the Sun, etc.

The average accelerations on the spacecraft with respect to the Earth over the interval DT are:

$$
\begin{aligned}
& \overline{\mathrm{A} 1}_{\mathrm{a}}=-(\overline{\mathrm{A} 1}+\overline{\mathrm{A} 1 \mathrm{~S}}) / 2 \\
& \overline{\mathrm{~A}}_{\mathrm{a}}=-(\overline{\mathrm{A} 2}+\overline{\mathrm{A} 2 \mathrm{~S}}) / 2 .
\end{aligned}
$$

If $\bar{X}$ and $\overline{D X}$ represent the position and velocity of the spacecraft respectively, then the corrected position and velocity are

$$
\begin{aligned}
& \overline{\mathrm{X}}=\overline{\mathrm{X}}+1 / 2 *(\overline{\mathrm{~A} 1}+\overline{\mathrm{A} 2}) * \mathrm{DT}^{2} \\
& \overline{\mathrm{DX}}=\overline{\mathrm{DX}}+(\overline{\mathrm{A} 1}+\overline{\mathrm{A} 2}) * \mathrm{DT}
\end{aligned}
$$

If KSET . LIT. 0 , then the orbit is to be retraced, in which case the following equations are substituted for those above:

$$
\begin{aligned}
& \overline{\mathrm{X}}=\overline{\mathrm{X}}-1 / 2 *(\overline{\mathrm{~A} 1}+\overline{\mathrm{A} 2}) * \mathrm{DT}^{2} \\
& \overline{\mathrm{DX}}=\overline{\mathrm{DX}}-(\mathrm{A} 1+\mathrm{A} 2) * \mathrm{DT}
\end{aligned}
$$

## Description:

AVERGE is used in connection with the multiconic scheme. It calculates the acceleration of the Earth due to the Moon at the present time and adds it to the acceleration at the last time and divides by two for the average acceleration over the step. The present acceleration is then saved for future use.

The same procedure is followed for the calculation of the acceleration due to the Sun on the Earth and on the spacecraft.

Once these two average acceleration vectors have been calculated, the position and velocity of the spacecraft are updated to reflect the accelerations. The changes in position and velocity are either added or subtracted depending on the flag KSET. If KSET. GE. 0, then forward propagation is desired and the terms are added. Otherwise, the terms are subtracted and the orbit path is retraced.


## Calling Sequence:

Purpose:

CALL AVSTRT
To average the input osculating elements over one revolution of the satellite in its orbit. This procedure yields the mean orbital elements at a time halfway through the first revolution and these elements are used to start the numerical averaging techniques for long-term orbit prediction.

Common Blocks Required: CONST, INTVRX, STATE, INPUT, INTVAR, CNTRL
Subroutines Required: INTEG, ORBIT, EQNS (indirectly)
Reference:
Uphoff, C., Numerical Averaging in Orbit Prediction, AAS/AIAA Preprint No. 72-934, Palo Alto, California, September 1972

Input / Output

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSIONS | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :---: |
| I/O | XX | 6 | STATE(l) | Position and velocity (osculating on <br> entry ${ }_{2}$ mean on exit) |
| I/O | TT | I | STATE(10) | TIME (advanced by one-half revolution <br> on exit) |

## Theory:

In the reference, it was suggested that the numerical averaging method should be started with mean orbital elements and an algorithm for performing the start-up was presented. The averaging technique is properly started from osculating elements by performing a one-revolution integration of the actual equations of motion and, at the same time, forming the integrals

$$
\vec{E}_{i}=\frac{1}{t} \int_{0}^{\infty} E_{i} d t
$$

At each step of the integration, the period $\bar{\tau}$ given by the running mean semi-major axis is calculated. The averaging start-up is terminated when the time $t$ is equal to this running mean value of the period. The procedure is implemented by a simple linear search when $t$ is found to be greater than $\bar{\tau}$. The mean values of the elements are then calculated from

$$
\bar{E}_{i}=\frac{1}{\tau} \int_{0}^{\tau} E_{i} \mathrm{dt}
$$

and are associated with the mean value of the time $\bar{\tau} / 2$ 。

## Description:

The averaging startup is invoked by setting KAVST (location 1098) to 1. This single value of the flag will cause a call to AVSTRT at the beginning of subroutine FOWARD, the position and velocity will be changed to values corresponding to the mean elements at time $-\bar{\tau} / 2$ and the time is advanced accordingly.

The startup will be performed in FOWARD only if KAVST is equal to 1. The associated logic in subroutine EQNS, however, will be in effect for any non-zero value of KAVST so that the startup may be called from other subroutines before entrance to FOWARD.

To use the subroutine, put the osculating state (position and velocity) into STATE common and the corresponding time into STATE(10). Set KAVST non-zero and call AVSTRT. The averaging integration will be performed using METH $=7$ with an internally set automatic computing interval. On return from AVSTRT, the position and velocity are changed to represent averaged values and the time is advanced by $1 / 2$ revolution. All other quantities except ELM (STATE(14-19)) are restored to the values they had before the call to AVSTRT.

The integration of the six additional equations is accomplished simultaneously with the one revolution propagation of the state. Subroutine INTEG is used to integrate forward in time using Method 7 with the automatic computing interval control logic. The
additional integrals are accumulated in INTVRX common and, at the end of each step, the current time is tested against the running mean orbital period. When $t \geq T$ the computing interval is changed to take the state from its value at the previous step to the desired time when $t=\tau$, as obtained by linear interpolation on the current and previous values of $t$ and $\tau$. The tolerance on this simple hunting procedure is set at 10 seconds.

AVSTRT


## SUBROUTINE BCONIC

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Called:

CALL BCONIC (UIN, X1, X2, TIN, $\varnothing \mathrm{UT}, \mathrm{V} \emptyset, \mathrm{DV}$ ) BCONIC uses Lambert's Theorem to solve for the Keplerian conic trajectory which connects two given radius vectors in a specified transfer time.
None
DOT, CROSS

Input/Output

| I/ $/$ ¢ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | PROGRAM DIMENSION | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | UIN | 1 | Call List | GM ( $\mathrm{km}^{3} / \mathrm{sec}^{2}$ ) of the central body |
| I | X1 | 6 | Call List | Position ( km ) and velocity ( $\mathrm{km} / \mathrm{sec}$ ) vectors at the first terminal |
| I | X2 | 6 | Call List | Position ( km ) and velocity ( $\mathrm{km} / \mathrm{sec}$ ) vectors at the second terminal |
| I | TIN | 1 | Call List | Transfer time (sec) |
| I | ФUT | 1 | Call List | Iteration's tolerance on transfer time error (fraction of transfer time) |
| $\emptyset$ | Vø | 6 | Call List | Terminal velocity vectors ( $\mathrm{km} / \mathrm{sec}$ ) |
| $\emptyset$ | DVø | 6 | Call List | Terminal excess velocity vectors (km/sec) |

Theory:
BCONIC implements the Schmidt-Bjorkman method to solve Lambert's problem, which is to find the conic section connecting two terminal radii in a specified trans- . fer time. The implemented method bears similarities to several of the methods described in Ref. 1.

The conic parameters are not explicit functions of transfer time, although the converse is true, setting up the requirement for iteration to solve the problem. The conic sections connecting the two terminal radii can be parameterized by a single independent variable (in which flight time is a monotonic function). The independent variable of the $\mathrm{S} / \mathrm{B}$ method is flight path angle, $\gamma$, at the first terminal, $\mathbf{R}_{1}$. The necessary equations for iterating to a solution are derived as follows. The distance from the force center is given by

$$
\begin{equation*}
r=\frac{p}{1+e \cos f} \tag{1}
\end{equation*}
$$

where $p$ is semi-latus rectum, $e$ is eccentricity, and $f$ is true anomaly none of which is known a priori. We presume to know the two terminal radii, $r_{1}$ and $r_{2}$, and the transfer angle, $\psi$. Since $r_{1}$ and $r_{2}$ are points on the same conic section,

$$
\begin{align*}
& r_{1}=\frac{p}{1+e \cos f_{1}}  \tag{2}\\
& r_{2}=\frac{p}{1+e \cos \left(f_{1}+\psi\right)}=\frac{p}{1+e \cos f_{1} \cos \psi-e \sin f_{1} \sin \psi} \tag{3}
\end{align*}
$$

The relationship of $f_{1}$ to $\gamma$ at $r_{1}$ is shown by

$$
\begin{equation*}
e \sin f_{1}=\frac{p}{r_{1}} \tan \gamma \tag{4}
\end{equation*}
$$

and we can eliminate $e$ and $f_{1}$ from equations (2) and (3) and solve for $p$ as a function of $\gamma$ and the geometry.

$$
\begin{equation*}
p=\frac{r_{1} r_{2}(1-\cos \psi)}{\left(r_{1}-r_{2} \cos \psi\right)+r_{2} \sin \psi \tan \gamma}=\frac{d_{3}}{d_{1}+d_{2} \tan \gamma} \tag{5}
\end{equation*}
$$

The potential problems in computation of Eq. (5) are eliminated by properly limiting the range of $\gamma$. This range will be discussed later. The velocity at $r_{1}$ can be calculated by

$$
\begin{equation*}
v=\frac{\sqrt{\mu \mathrm{p}}}{\mathrm{r}_{1} \cos \gamma} \tag{6}
\end{equation*}
$$

where $\mu$ is the force center's gravitational constant. The semi-major axis of the transfer is given by

$$
\begin{equation*}
a=\mu /\left(\frac{2 \mu}{r_{1}}-v^{2}\right) \tag{7}
\end{equation*}
$$

One might worry about parabolic transfers when studying Eq. (7); however, the special treatment of parabolic transfers will be described later.

At this point, we launch into a discussion of the calculation of transfer time. BCONIC's formulation of transfer time is written in terms of incremental eccentric anomaly, $\varphi$, on the conic section connecting $r_{1}$ and $r_{2}$.

$$
\begin{equation*}
n t=\sigma\left[(\varphi-S \varphi)+\frac{r_{1}}{a} S \varphi+\frac{r_{1} v \sin \gamma}{\sqrt{\mu \sigma a}}(1-C \varphi)\right] \tag{8}
\end{equation*}
$$

In this equation, n is the mean motion $\left(\sqrt{\mu /(\sigma \mathrm{a})^{3}}\right)$ while $\sigma, \mathrm{S} \varphi$, and $\mathrm{C} \varphi$ are defined by the following table.

|  | Elliptical | Hyperbolic |
| :--- | :---: | :---: |
| $\mathrm{S} \varphi$ | $\sin \varphi$ | $\sinh \varphi$ |
| $\mathrm{C} \varphi$ | $\cos \varphi$ | $\cosh \varphi$ |
| $\sigma$ | +1 | -1 |

It has been shown in Ref. 2 that:

$$
\begin{align*}
& \sin \psi=\frac{\sigma \sqrt{\mu \mathrm{p}}}{\mathrm{n} \mathrm{r}_{1} \mathrm{r}_{2}}[\beta(1-\mathrm{C} \varphi)-\alpha \mathrm{S} \varphi]  \tag{9}\\
& \cos \psi=\frac{\mathrm{r}_{1}}{\mathrm{r}_{2}}+\frac{\sigma \mathrm{a}^{2}}{\mathrm{r}_{1} \mathrm{r}_{2}}\left[\left(\alpha^{2}-\sigma \beta\right)(1-\mathrm{C} \varphi)+\alpha \beta \mathrm{S} \varphi\right] \tag{10}
\end{align*}
$$

where $\beta=\frac{\mathrm{r}_{1}}{\mathrm{a}}$ and $\alpha=\frac{\mathrm{r}_{1} \mathrm{v} \sin \gamma}{\sqrt{\mu \sigma \mathrm{a}}}$. It is easy to solve these equations for $\mathrm{S} \varphi$ and ( $1-\mathrm{C} \varphi$ ), rendering Eqs. (11) and (12) in which $d_{1}=r_{1}-r_{2} \cos \psi$ and $\mathrm{d}_{2}=\mathrm{r}_{2} \sin \psi$,

$$
\begin{align*}
\mathrm{S} \varphi & =-\frac{1}{\mathrm{r}_{1}}\left[\sqrt{\frac{\sigma \mathrm{a}}{\mathrm{p}}}\left(\alpha^{2}-\sigma \beta\right){\left.d_{2}+\alpha d_{1}\right]}_{1-\mathrm{C} \varphi}=\frac{1}{\mathrm{a}}\left[\sqrt{\frac{\sigma a}{p}} \alpha d_{2}+d_{1}\right]\right. \tag{11}
\end{align*}
$$

If the transfer orbit is elliptical, $\varphi=2 \tan ^{-1}\left(\frac{\mathrm{~S} \varphi}{1-\mathrm{C} \varphi}\right)$, while if it is hyperbolic, $\varphi=\ln (\mathrm{S} \varphi+\mathrm{C} \varphi)$. It should be noted that $\varphi$ is the only transcendental function to be evaluated in computing the transfer time.

## Iteration:

Equations (5) through (12) provide the recipe for computing transfer time as a function of flight path angle at the first terminal. The discussion which follows is concerned with the iteration to compute the unique flight path angle which renders the "desired" transfer time between the two terminals. BCONIC uses a simple Newton-Raphson iteration which invariably provides rapid convergence.

$$
\begin{equation*}
\gamma_{n+1}=\gamma_{n}+\left(t_{d}-t_{n}\right)\left(\frac{\gamma_{n}-\gamma_{n-1}}{t_{n}-t_{n-1}}\right) \tag{13}
\end{equation*}
$$

Flight time is a monotonic-increasing function over the allowable range of $\gamma$.

Limits:
The allowable transfer orbit regions are sketched in Fig. 1 for $\psi<180^{\circ}$ and for $\psi>180^{\circ}$.


$$
\psi>180^{\circ}
$$



No orbits traveling from $R_{1}$ to $R_{2}$ are permitted in the shaded regions of the figure. The upper limit on flight path angle is, in either case, the path angle of the escape parabola $P_{2}$ at $R_{1}$. This flight path angle is derivable in terms of the geometry if we solve Eqs. (2), (3), and (4) for $\gamma$ with the conditions that $\mathrm{e}=1$ and $\mathrm{f}_{1}=2 \gamma$.

$$
\begin{align*}
& \mathrm{d}_{1} \cos 2 \gamma+\mathrm{d}_{2} \sin 2 \gamma=\mathrm{r}_{2}-\mathrm{r}_{1}  \tag{14}\\
& \gamma_{\mathrm{p}}=\frac{1}{2}\left[\cos ^{-1}\left(\frac{\mathrm{r}_{2}-\mathrm{r}_{1}}{\sqrt{\mathrm{~d}_{1}^{2}+\mathrm{d}_{2}^{2}}}\right)+\tan ^{-1}\left(\frac{\mathrm{~d}_{2}}{\mathrm{~d}_{1}}\right)\right] \tag{15}
\end{align*}
$$

If the second term in Eq. (15) is evaluated with regard for quadrant in the range of $0^{\circ}$ to $360^{\circ}$, the arccosine ambiguity still leads to two solutions for $\gamma_{p}$. The larger solution is the flight path angle of the escape parabola. The smaller solution is the flight path angle of the prograde transfer parabola connecting $R_{1}$ and $\mathrm{R}_{2}$. The transfer time for this prograde parabola is computed according to

$$
\begin{equation*}
p_{p}=2 r_{1} \cos ^{2} \gamma_{p 1} \tag{16}
\end{equation*}
$$

$$
\begin{equation*}
t_{p}=\frac{\frac{1}{2}}{\frac{p_{p}^{3}}{\mu}}\left\{\tan \left(\gamma_{p 1}+\frac{\psi}{2}\right)+\frac{\tan ^{3}\left(\gamma_{p 1}+\frac{\psi}{2}\right)}{3}-\tan \gamma_{p 1}-\frac{\tan ^{3} \gamma_{p 1}}{3}\right\} \tag{17}
\end{equation*}
$$

If the desired transfer time is equal to the parabolic transfer time (within tolerance), no iteration is required and the potential parabolic singularity in Eq. (7) is avoided. Otherwise, the desired transfer time will be less than $t_{p}$ (hyperbolic solution) or greater than $t_{p}$ (elliptical solution). Thus, the flight path angle limits for the ensuing iteration may be set as follows:

Elliptical


The lower limits on the hyperbolic solution correspond to the straight-line path if $\psi \leq 180^{\circ}$ and to the path along $-R_{1}$ and then along $R_{2}$ if $\psi>180^{\circ}$. Equation (5) has no singularities within the above limits.

## Starting:

If the solution is expected to be hyperbolic, the initial $\gamma$ is taken to be a little less than $\gamma_{p 1}$, the prograde parabolic value. The amount less is chosen simply as $1 \%$ of the allowed range of $\gamma$. The starting value for $\gamma$ when the solution is expected to be elliptical is the first-terminal flight path angle of the minimum-period ellipse joining $R_{1}$ and $R_{2}$. The minimum-period ellipse is characterized by the following semi-major axis and semi-latus rectum (Ref. 3).

$$
\begin{align*}
& a_{\min }=\frac{1}{4}\left(r_{1}+r_{2}+\sqrt{d_{1}^{2}+d_{2}^{2}}\right)  \tag{18}\\
& p_{\min }=2\left(2 a_{\min ^{-}}-r_{1}\right)\left(2 a_{\min ^{-}}-r_{2}\right) / \sqrt{d_{1}^{2}+d_{2}^{2}} \tag{19}
\end{align*}
$$

The flight path angle is then found by substituting $p_{\min }$ into Eq. (5).

$$
\begin{equation*}
\gamma=\tan ^{-1}\left(\frac{d_{3}-p_{\min } d_{1}}{p_{\min } d_{2}}\right) \tag{20}
\end{equation*}
$$

## I/O Computations:

BCONIC is entered with a 6 -vector of Cartesian position and velocity components for each terminal. The magnitudes of the position vectors, $R_{1}$ and $R_{2}$, are $r_{1}$ and $r_{2}$, respectively. The velocity vector, $V_{1}$, at the first terminal is used in determining whether $\psi$ is less or greater than $180^{\circ}$. The cosine of $\psi$ is given by

$$
\begin{equation*}
\cos \psi=\frac{R_{1} \cdot R_{2}}{r_{1} r_{2}} \tag{21}
\end{equation*}
$$

but the sine of $\psi$, computed as $\sqrt{1-\cos ^{2} \psi}$, has a sign ambiguity. This ambiguity is resolved by comparing the cross-product, $R_{1} \times V_{1}$, with the cross-product $R_{1} \times R_{2}$. If these cross-products have a negative dot product, $\sin \psi$ must be negative to exclude retrograde orbits with respect to $V_{1}$ at $R_{1}$.

$$
\begin{aligned}
& \left(R_{1} \times V_{1}\right) \cdot\left(R_{1} \times R_{2}\right)>0=\sin \psi=\sqrt{1-\cos ^{2} \psi} \\
& \left(R_{1} \times V_{1}\right) \cdot\left(R_{1} \times R_{2}\right) \leq 0 \quad=\sin \psi=-\sqrt{1-\cos ^{2} \psi}
\end{aligned}
$$

The transfer angle is then defined by

$$
\begin{equation*}
\psi=\tan ^{-1}\left(\frac{\sin \psi}{\cos \psi}\right), \quad 0<\psi \leq 360^{\circ} \tag{22}
\end{equation*}
$$

At the conclusion of the iteration, BCONIC computes the vector velocity, $\mathrm{V}^{*}$, of the transfer orbit at each terminal. If the transfer angle is not $180^{\circ}$ or $360^{\circ}, R_{1}$ and $\mathrm{R}_{2}$ define a unique plane for the transfer orbit. Then the velocities are

$$
\begin{equation*}
\mathrm{V}_{1}^{*}=\mathrm{c}_{1} \mathrm{R}_{1}+\mathrm{c}_{2}\left(\mathrm{R}_{1} \times \mathrm{R}_{2}\right) \times \mathrm{R}_{1} \tag{23}
\end{equation*}
$$

and

$$
\begin{equation*}
\mathrm{V}_{2}^{*}=\mathrm{c}_{3} \mathrm{R}_{1}+\mathrm{c}_{4} \mathrm{~V}_{1}^{*} \tag{24}
\end{equation*}
$$

where

$$
\begin{align*}
& c_{1}=v \sin \gamma / r_{1}  \tag{25}\\
& c_{2}=v \cos \gamma /\left(d_{2} r_{1}^{2}\right)  \tag{26}\\
& c_{3}=-\sqrt{\mu \sigma a} \operatorname{se} /\left(r_{1} r_{2}\right) \tag{27}
\end{align*}
$$

and

$$
\begin{equation*}
c_{4}=\left(r_{1} C \varphi+\sigma \alpha a S \varphi\right) / r_{2} \tag{28}
\end{equation*}
$$

If $|\sin \psi|<10^{-6}$, the transfer orbit plane is ill-defined by $R_{1}$ and $R_{2}$, so the plane of $R_{1}$ and $V_{1}$ is chosen. In this case,

$$
\begin{equation*}
V_{1}^{*}=c_{1} R_{1}+c_{2}\left(R_{1} \times V_{1}\right) \times R_{1} \tag{29}
\end{equation*}
$$

where

$$
\begin{equation*}
c_{2}=v \cos \gamma /\left|\left(R_{1} \times v_{1}\right) \times R_{1}\right| \tag{30}
\end{equation*}
$$

The other calculations are unchanged. BCONIC also computes the terminal velocity differences for output.

$$
\begin{align*}
& \Delta \mathrm{V}_{1}=\mathrm{V}_{1}^{*}-\mathrm{V}_{1}  \tag{31}\\
& \Delta \mathrm{~V}_{2}=\mathrm{V}_{2}^{*}-\mathrm{V}_{2} \tag{32}
\end{align*}
$$



## REFERENCES

1. Battin, R. H. ; Astronautical Guidance, McGraw-Hill, 1964.
2. "Programer's Manual for Quick-Look Mission Analysis Program," PhilcoFord WDL TR-2217, January 1964.
3. "Design Parameters for Ballistic Interplanetary Trajectories - Part I: One-Way Transfers to Mars and Venus," JPL TR-32-77, January 1963.

## SUBROUTINE BELL

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:

## CALL BELL

BELL computes statistical estimates of end constraint errors and second-midcourse requirements based on estimated first-midcourse execution error statistics.

CNTRL, INPUT, MCCOM, STATE
CROSS, DOT, DVMAG, FOWARD, MCBURN, MVTRN, RETDV, SENSO, VNORM

Input / Output

| 1/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | JC | 1 | CNTRL(7) | Central body number of state |
| I | TFINAL | 1 | INPUT(4) | Trajectory stop-time (sec). Input value matters not. |
| I | WTO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | SIGAT | 1 | INPUT(435) | Attitude execution error statistic (rad) |
| I | SIGDV | 1 | INPUT(436) | Proportional velocity execution error statistic ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | PRX | 1 | MCCOM $(3)$ | Scale factor for printed statistics |
| I | XMC | 6 | MCCOM $(6)$ | Pre-midcourse state ( km , $\mathrm{km} / \mathrm{sec}$ ) |
| I | DV | 3 | MCCOM(12) | Midcourse velocity impulse (km/sec) |
| I | DVMG | 1 | MCCOM(15) | Magnitude of DV (km/sec) |
| I | TMCS | 1 | MCCOM(18) | Midcourse time (seconds from anchor epoch) |
| 0 | DVB4 | 1 | MCCOM(24) | Velocity impulse magnitude expended prior to current maneuver ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | EXFUEL | 1 | MCCOM(26) | Expected fuel required at next maneuv to correct errors of this one (kg) |
| 0 | SIGOUT | 6 | MCCOM(40) | Expected end constraint error statistics ( ${ }^{\mathrm{f}_{\mathrm{f}}} \mathrm{sec}, \mathrm{v}_{\mathrm{\infty}} \mathrm{~m} / \mathrm{sec},{ }^{\mathrm{r}} \mathrm{c}-\mathrm{m} / \mathrm{sec}, \mathrm{f}-\mathrm{kg}, \mathrm{r}_{\mathrm{p}-\mathrm{km}}$ ) |


| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | WTF | 1 | MCCOM(47) | Weight (kg) after the first <br> midcourse maneuver |
| I | TMC2 | 1 | MCCOM(48) | Time(sec) of the next midcourse <br> maneuver relative to anchor epoch |
| O | EXV2 | 1 | MCCOM(49) | Expected next-maneuver velocity <br> magnitude (m/sec) |
| I | DPT | 3,10 | MCCOM(50) | Sensitivity matrix (transposed) of <br> end constraints to midcourse <br> velocity variations |
| O | KBURN | 1 | MCCOM(154) | Midcourse burn option key |
| O | IR | 1 | MCCOM(158) | Sensitivity option key |
| I | KGLAW | 1 | MCCOM(164) | Guidance law selection key |
| O | ICB | 1 | MCCOM(165) | Central body number <br> at midcourse |
| $O$ | $X$ | 6 | STATE(1) | State (km, km/sec) relative to JC |
| $O$ | T | 1 | STATE(10) | Time (sec) to which <br> X corresponds |

Theory:
Errors in the miss vector, $B$, are assumed to be entirely attributable to errors in executing the required midcourse maneuver: Deviations in $B$ are assumed linearly related to deviations in the corrective velocity impulse, $\Delta \mathrm{V}$, by

$$
\begin{equation*}
\delta \mathrm{B}=\frac{\partial \mathrm{B}}{\partial \Delta \mathrm{~V}} \delta(\Delta \mathrm{~V}) \tag{1}
\end{equation*}
$$

Deviations in $\Delta V$ are, in turn, related to errors along $\Delta V$ (cut off error, $\epsilon_{v}$ ) and normal to $\Delta V$ (pointing errors, $\varepsilon_{e}$ and $\epsilon_{\mathrm{n}}$ ) by

$$
\delta(\Delta V)=\frac{\partial \Delta V}{\partial\left(\epsilon_{v}, \epsilon_{e}, \epsilon_{n}\right)}\left\{\begin{array}{l}
\epsilon_{v}  \tag{2}\\
\epsilon_{e} \\
\epsilon_{n}
\end{array}\right\}
$$

The pointing errors $\varepsilon_{e}$ and $\varepsilon_{n}$, are components of the error vector eastward and northward, respectively, from $\Delta V$ on the celestial sphere as shown in the figure.


The partial derivative in Equation (2) is therefore written as

$$
\frac{\partial \Delta V}{\Delta\left(\epsilon_{\mathrm{v}}, \epsilon_{\mathrm{e}} ; \epsilon_{\mathrm{n}}\right)}=\left[\begin{array}{lll}
\mathrm{V} & \mathrm{E} & \mathrm{~N} \tag{3}
\end{array}\right]
$$

V, E, and N are unit column vectors defined by

$$
\begin{align*}
& \mathrm{V}=\Delta \mathrm{V} /|\Delta \mathrm{V}|  \tag{4a}\\
& \mathrm{E}=\mathrm{K} \times \mathrm{V} /|\mathrm{K} \times \mathrm{V}| \quad\left(\mathrm{K}=\left[\begin{array}{l}
0 \\
0 \\
1
\end{array}\right]\right)  \tag{4b}\\
& \mathrm{N}=\mathrm{V} \times \mathrm{E} \tag{4c}
\end{align*}
$$

The partial derivative in Equation (1) is the gradient needed for computing the required midcourse maneuvers and is therefore available. It is computed by the secant method in the vicinity of the solution. Because $B$ has six components (time of flight, $v_{\infty}, v_{c}$, total fuel, $r_{p}$, i) $\partial B / \partial \Delta V$ is a $6 \times 3$ matrix.

Defining P by

$$
\begin{equation*}
P=\frac{\partial B}{\partial \Delta V} \quad \frac{\partial \Delta V}{\partial\left(\epsilon_{\mathrm{v}}, \epsilon_{\mathrm{e}}, \epsilon_{\mathrm{n}}\right)} \tag{5}
\end{equation*}
$$

we can combine Equations (1) and (2) to obtain

$$
\delta B=P\left\{\begin{array}{l}
\epsilon_{\mathrm{v}}  \tag{6}\\
\epsilon_{\mathrm{e}} \\
\epsilon_{\mathrm{n}}
\end{array}\right\}
$$

The covariance matrix of miss vector errors, $C$, is given by

$$
\left.\left.\left.C=E\left[\delta B \delta B^{T}\right]=P E\left\{\begin{array}{l}
\epsilon_{v}  \tag{7}\\
\epsilon_{e} \\
\epsilon_{n}
\end{array}\right\} \right\rvert\, \begin{array}{lll}
\epsilon_{v} & \epsilon_{e} & \varepsilon_{n}
\end{array}\right)\right] P^{T}
$$

where $E$ is the expectation operator and where $P$, being deterministic, is not affected by E . Assuming no correlation among $\varepsilon_{\mathrm{v}}$, $\epsilon_{\mathrm{e}}, \epsilon_{\mathrm{n}}$,

$$
E\left[\left\{\begin{array}{l}
\epsilon_{\mathrm{v}}  \tag{8}\\
\epsilon_{\mathrm{e}} \\
\epsilon_{\mathrm{n}}
\end{array}\right\}\left(\begin{array}{lll}
\epsilon_{\mathrm{v}} & \epsilon_{\mathrm{e}} & \left.\epsilon_{\mathrm{n}}\right)
\end{array}\right]=\left[\begin{array}{ccc}
\sigma_{\mathrm{v}}^{2} & 0 & 0 \\
0 & \sigma_{\mathrm{e}}^{2} & 0 \\
0 & 0 & \sigma_{\mathrm{n}}^{2}
\end{array}\right]\right.
$$

If we input values for $\sigma_{v}, \sigma_{e}$, and $\sigma_{n}$, we can then calculate the covariance matrix of miss vector errors by Equation (7).

Then the standard deviations of the miss-vector components are computed as uncorrelated errors by simply square-rooting the appropriate diagonal element of C. These are then multiplied by PRX to be displayed as "Probability $P$ " values.

The problem of appropriately supplying execution error statistics now arises. If we assume cut-off error to be comprised of independent resolution and proportional errors,

$$
\begin{equation*}
\epsilon_{\mathrm{v}}=\epsilon_{\text {res }}+\epsilon_{\text {prop }}|\Delta \mathrm{V}| \tag{10}
\end{equation*}
$$

then

$$
\begin{equation*}
\sigma_{\mathrm{v}}^{2}=\sigma_{\mathrm{res}}^{2}+\sigma_{\text {prop }}^{2}|\Delta \mathrm{v}|^{2} \tag{11}
\end{equation*}
$$

The resolution error is coded into the program as a uniformly distributed random variable on a $.1 \mathrm{~m} / \mathrm{s}$ interval centered at zero. The proportional error is normally
distributed with input standard deviations.
Pointing errors are conveniently specified as normally-distributed along orthogonal axes with the same variance on each axis. The distributions and axis orientations would more appropriately be related to the plane of the attitude maneuver, but this is more difficult to simulate. The pointing error is input as a single number which represents the standard deviation of equal normal distributions along the E and N axes.

## Second Maneuver Requirements

We now consider the requirements of a second maneuver, executed at $t_{2}$, to correct the errors made in executing the first maneuver at $t_{1}, t_{1}<t_{2}$. We must develop a mapping, $M$, of first midcourse execution errors, $\delta\left(\Delta \mathrm{V}_{1}\right)$; into a second-midcourse velocity impulse, $\Delta \mathrm{V}_{2}$. The end constraint errors, $\delta \mathrm{B}$ related to $\delta(\Delta \mathrm{V})$ by Equation (1) are the very ones to be corrected by $\Delta V_{2}$, so

$$
\begin{equation*}
\delta \mathrm{B}=\frac{\partial \mathrm{B}}{\partial \Delta \mathrm{~V}_{1}} \quad \delta\left(\Delta \mathrm{~V}_{1}\right)=-\frac{\partial \mathrm{B}}{\partial \Delta \mathrm{~V}_{2}} \quad \Delta \mathrm{~V}_{2} \tag{12}
\end{equation*}
$$

$$
\begin{equation*}
\Delta V_{2}=-\left(\frac{\partial \mathrm{B}}{\partial \Delta \mathrm{~V}_{2}}\right)^{-1} \quad \frac{\partial \mathrm{~B}}{\partial \Delta \mathrm{~V}_{1}} \quad \delta\left(\Delta \mathrm{~V}_{1}\right)=\mathrm{M} \delta\left(\Delta \mathrm{~V}_{1}\right) \tag{13}
\end{equation*}
$$

The particular constraint error set to be nulled by $\Delta \mathrm{V}_{2}$ depends on the guidance law invoked. We assume that the guidance law for the second maneuver is the same as for the first. If the guidance lawis FTA, FTE, VTE, or MTF, B has three components and, assuming the independence of their gradients with respect to $\Delta \mathrm{V}_{2}$, the indicated inverse exists. For the MFG law where there are only two end constraints, $\partial \mathrm{B} / \partial \Delta \mathrm{V}_{2}$ is made invertible by adjoining a row perpendicular to the two rows it has. The third column of the inverse is then zeroed before the multiplication, $\frac{\partial B^{-1}}{\partial \Delta V} \frac{\partial B}{\partial \Delta V_{1}}$, takes place because a third-constraint error must have no influence on the MFG solution. The expected value of the second-midcourse velocity is computed as the square root of the trace of the following matrix.
$E\left(\Delta V_{2} \Delta V_{2}^{T}\right)=M E\left(\delta\left(\Delta V_{1}\right) \delta\left(\Delta V_{1}\right)^{T}\right) \quad M^{T}$

$$
=M P\left[\begin{array}{ccc}
\sigma_{\mathrm{v}}^{2} & 0 & 0  \tag{14}\\
0 & \sigma_{\mathrm{e}}^{2} & 0 \\
0 & 0 & \sigma_{\mathrm{n}}^{2}
\end{array}\right]
$$

This expected value is then scaled by PRX to bring it up to a specified probability (univariate Gaussian) level for printout.

It would be desirable, perhaps, to present the expected spherical probable secondmidcourse velocity, but the computation of this measure is too complex and timeconsuming to be merited. The expected fuel is computed from the expected velocity by means of the rocket equation.

The sensitivity matrix, $\quad \partial \mathrm{B} / \partial \Delta \mathrm{V}_{1}$, is usually available, since BELL is called right after the first-midcourse maneuver has been calculated. If not, however, it is computed by finite differences by calling SENSO. The second-midcourse sensitivity matrix, $\frac{\partial \mathrm{B}}{\partial \Delta \mathrm{V}_{2}}$, is computed at $\mathrm{t}_{2}$ by finite differences about the trajectory which would result from a perfectly-executed maneuver at $t_{1}$.

Compute the transformation [VEN] from local tangent of $\Delta V$ to EE50 coordinates

Call SENSO to compute DPT, the sensitivity of end conditions to midcourse variations


## SUBROUTINE BIGMAT

Calling Sequence:
Purpose:

CALL BIGMAT (A, VALU, EVE, NN, NEIG, NVEC) BIGMAT calculates eigenvalues and eigenvectors of a symmetric array using Householder's method.

Common Blocks required: None

Subroutines required: None

Reference:
Householder, A. S., and Bauer, F. L., "On Certain Methods for Expanding the Characteristic Polynomial", Numerische Mathematik 1. Bank, 1. Heft, p. 29. (1959).

Input / Output

| SYMBOLIC |  |  | COMMON |  |  |
| :--- | :---: | :---: | :---: | :--- | :---: |
| I/O | NAME | DIMENSION | BLOCK | DE FINITION |  |
| I/O | A | 21 | Calling <br> Operand | Symmetric Array * |  |
| O | EVE | 6,6 | Calling <br> Operand | Eigenvectors |  |
| I | NEIG | 1 | Calling <br> Operand | Number of eigenvalues desired |  |
| I | NN | 1 | Calling <br> Operand | Dimension of array * |  |
| I | NVEC | 1 | Calling <br> Operand | Number of eigenvectors desired |  |
| O | VALU | 6 | Calling <br> Operand | Eigenvalues |  |

* The square symmetric array $B$ has dimensions NN x NN, however, it is loaded into BIGMAT as A, a column vector that includes only diagonal and lower diagonal elements. Thus $A(N)$ corresponds to $B((N-1) / N N+1, \operatorname{MOD}(N-1, N N)+1)$.

Theory:
BIGMAT uses the Householder method of calculation. This method involves a codiagonalization of the original matrix before the roots are calculated. See the above reference for details of the method.

## Description:

Given the column vector $A$, its dimension $N N$ (see *), the number of eigenvalues desired

NEIG, and the number of eigenvectors desired NVEC, BIGMAT will calculate the corresponding number of eigenvalues and eigenvectors. The eigenvalues will be output largest first in the vector VALU. The eigenvectors will be output in the matrix EVE so that the elements in the ith column correspond to the $i$ th eigenvalue. Since the eigenvalue is needed for the calculation of the corresponding eigenvector, NVEC must be less than or equal to NEIG. It is inportant to note that the elements in A are changed in the subroutine and the values output are in general not the same as those input. If NVEC is not equal to zero, then there will be output written on Unit 6 as follows: first a value "EPS" will be printed and then several lines indicating the eigenvalues and their corresponding eigenvectors. The value of EPS represents the accuracy of the eigenvalue calculation. For good calculations of the eigenvectors, EPS must be small compared to the difference of any two eigenvalues.

## BLOCK DATA

| Calling Sequence: |  | Preloaded |  |
| :---: | :---: | :---: | :---: |
| Purpose: |  | BLOCK DATA initializes various constants in the program. |  |
| Common Blocks Required: |  | AVG, CONST, INPUT |  |
| Subroutines Required: |  | None |  |
| Reference: |  | Scarborough, James Bo, NUMERICAL MATHEMATICAL ANALYSIS, The Johns Hopkins Press, 1930. |  |
| Input / Output |  |  |  |
| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | DEFINITION |
| 0 | A | 1000 | Real portion of input array |
| 0 | ABSCIS | 78 | Abscissa for Gaussian quadrature formula |
| 0 | CONST | 50 | Program consta.nts |
| 0 | KOPT | 100 | Integer portion of input array |
| 0. | WEIGHT | 78 | Weights for Gaussian quadrature formula |

## Description:

The arrays A and KOPT contain initialized values for variables which control program operations and may be changed through input. These initialized values are listed in Tables I and II. For a detailed description of these variables, see write $-u p$ of INPUT common.

The arrays WEIGHT and ABSCIS contain the weights and corresponding abscissa for the Gaussian quadrature formula. The values stored in these arrays are listed in Tables III and IV. For a description of the Gaussian quadrature and the use of these values, see the above reference.

The array CONST contains general mathematical constants in addition to constants pertaining to the planets. BLOCK DATA initializes only the mathematical constants. These constants and their preset values are as follows:

| LOCATION | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PRESET VALUE | DESCRIPTION |
| :---: | :---: | :---: | :---: |
| 1 | RAD | 57.29577951308232 | Degrees per radian |
| 2 | PI | 3.141592653589793 | $\pi$ |
| -3 | PI2 | 6.283185307179586 | 2 \% |
| 4 | AU | 149597893. | Kilometers per astronomical unit. |
| 41 | THS | 3600, | Seconds per hour |
| -42 | TSH | 2.7777777777778D-4 | Hours per second |
| 43 | TDS | 86400. | Seconds per day |
| 44 | TSD | 1.1574074074074D-5 | Days per second |
| 45 | G | . 0098066 | Average surface gravity on Earth (KM/sec ${ }^{2}$ ) |


| T, |  | Tocnminot | 1 vntue | Incamtor | $\pm$ Unter |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | 540000.000 | 5 | 1.09007000 | 6 | 1.70000000 |
| 8 | 0.1000000000-30 | 300 | $0.10009000052 ?$ | 30 | 337.400000 |
| 100 | 2. 16355300 | 107 | 32.1935 .400 | 179 | 390500.900 |
| 103 | 42915.5150 | 104 | 125770603. | 105 | 37910699.0 |
| 109 | $0.13271545001 ?$ | 110 | 490 ?.77800 | 1.1? | 2329.31600 |
| 113 | 6093.63600 | 1111 | 6378.7 .4000 | 115 | 3490.53000 |
| 115 | 7.1422 .0000 | 117 | 57505.0009 | 118 | 25434.0000 |
| 119 | 24933.0000 | 12.0 | 6345.07000 | 1?? | 705070.000 |
| 122 | 1733.09000 | 1260 | 0.7292123571104 | 1.7 | 0.79 98200000-04 |
| 128 | $0.2758549005-03$ | 1340 | $0.266369950 \mathrm{D-05}$ | 197 | 0. 1500000000 D 0 - |
| 193 | 0.200000050 | 20.1 | 909.000000 | $20 ?$ | 20000.0000 |
| 221 | 2.20000000 | 22.2 | 25.0000000 | 260 | 48.5000000 |
| 261 | 26.6892000 | 262 | 26.6892000 | 237 | -9357. 32000 |
| 231 | -9367.12000 | 32.1 | 900.000000 | $32 ?$ | 20000.0000 |
| 331 | 20.1000000 | 33.2 | 25.0000000 | 357 | $0.3201000000-0$. |
| 351 | 0.120200000 D 01 | 352 0 | ก. 2202000000-0 | 360 | 3.55559000 |
| 361 | 3.55550000 | 3300 | 0.100000000 D 21. | 381 | 0.100000000023. |
| 332 | 0.100000000 D 2 ? | 33.30 | 0.100000000 D 21 | 394 | 0.100000000 D 27 |
| 335 | 0.100000000 D 27 | 4070 | 0.530000000 | 405 | 126000000. |
| 409 | 400000000. | 410 | 27.7072309 | $4!7$ | 47. 377380 |
| 412 | 113.716 .372 | 413 | 148.956967 | 114 | -117.5 71732 |
| 415 | -32. 3760930 | 416 | -76.8429650 | 417 | -70.6555190 |
| 420 | 2333.00000 | 421 - | -59.0000000 | $4 ? 3$ | 396000.000 |
| 423 | 0.620000000 | 433 | 13550.0000 | 434 | 7300.00009 |
| 435 | 0.700000000 | 4360 | 0.2000000001-07 | 437 | $0.7300300008-01$ |
| 438 | $0.300000000 \mathrm{D}-03$ | 439 | 367)0.0000 | 440 | 259200.000 |
| 441 | 226.000000 | 442 | 232.500000 | 44? | 71. 1700000 |
| 444 | 2333.00000 | 446 | 59.0000000 | 817 | 6000.00000 |
| 448 | 6000.00000 | 450 | 0.100000000 D 21 | 460 | 18000.0000 |
| 470 | 6.00000000 | 4710 | 0.167800000 | 472. | 20.4000000 |
| 473 | 13.7700000 | 474 | 5.00000000 | 4.75 | 20.0000000 |
| 473 | 7200.00000 | 4790 | $0.3000000005-0 ?$ | 430 | $-27.7357020$ |
| 431 | -13.9032240 | 482 | -24.7576070 | 483 | -35.4430090 |
| 434 | 64.0236430 | 485 | 35.0133150 | 486 | 38.9103660 |
| 487 | -32.9759380 | 490 | 10.0000000 | 491. | 10.0000000 |
| 492 | 10.0000000 | 4930 | $0.100000000 \mathrm{D}-03$ | 494 |  |
| 495 | $0.200000000 \mathrm{D}-01$. | 496 | 5.00000000 | 497 | 0.300000000 |
| 501 | 616000.000 | 502. | 925000.000 | $50 ?$ | 565000.000 |
| 504 | 43000000.0 | 505 | 50000000.0 | 506 | 51000000.0 |
| 507 | 86000000.0 | 508 | 33000000.0 | 509 | 0.100000000011 |
| 510 | 110000.000 | 512 | 20.0000000 | 5?4 | 2.00002000 |
| 515 | 90.0000000 | $5 ? .6$ | 12.0090000 | 517 | 239.000009 |
| 513 | 20.0000000 | 519 | 20.4100000 | 520 | 1.00000000 |

```
    TABIE II
    INTEGER PORTION OT INPUT ARRAY
```

| LOCATION | VAJIJE | LOCATION | VALUE | SOCATION | VALUE |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 3 | 2 | 10 | 1 | 11. | 2 |
| 14 | 3 | 1.5 | 3 | 17 | 5 |
| 18 | 1 | 19 | 1 | 31 | $? 7$ |
| 32 | 2 | 35 | 1 | 35 | 4 |
| 39 | 3 | 40 | 2 | 41. | 10 |
| 42 | 15 | 43 | . 21 | 48 | 1 |
| 51 | 10 | 52 | -. 2 | 53 | 50 |
| 54 | 17 | 55 | 1 | 60 | $?$ |
| 61 | 11 | 62 | 2 | 63 | 2 |
| 64 | 1 | 65 | 10 | 66 | 1 |
| 67 | 100 | 68 | 6 | 69 | 3 |
| 70 | 95 | 71 | 6 | 75 | 6 |
| 83 | 1.1 | 88 | 12 | 89 | 1234567 |


| I,OCATION | VALUE | LOCNTION | VALUS | $\underline{\text { rocantoy }}$ | TATTTE |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $?$ | 1.09007009 | 2 | 0.398939383 | 3 | 7. 5555555.56 |
| 4 | 0.652145755 | 5 | ก.317951945 | 6 | ก.560989089 |
| 7 | 0.473629670 | 2 | 0.735926385 | 9 | 2.167973035 |
| 10 | 0.350751573 | 11 | n.1.71.3).149? | 19 | 0.4770 .50104 |
| 1.3 | $0.38383005 ?$ | 14 | ก.27970530? | 95 | 0.790401955 |
| 15 | ?. 352583793 | 1.7 | ก.213705¢ 15 | 19 |  |
| 19 | 0.101228536 | 20 | 0.330239355 | 27 |  |
| $2 . ?$ | 0.250690696 | 23 | 0. 3 ¢n¢ 43151 | 2 ? | 0.99.71300cn-01 |
| 25 | 0.295521325 | 25 | ท. 2 ¢0 265710 | 27 |  |
| 28 | 9.149451349 | 29 |  | 30 | 0.372925007 |
| $3 ?$ | 0.262904545 | 32. | ก.233792765 | 3.3 |  |
| 34 | 0.125592353 | 35 | 0.555635671 | 23 |  |
| 37 | 0.233492537 | 33 | 0.203157427 | 39 | 0. 0 60278329 |
| 40 | 0.106939326 | 41 | ก. $271753354 \mathrm{D}-0$ ? | 12 | ?. 232553553 |
| 4.3 | 0.226293180 | 44 | 0.207916049 | 45 | 0.970145001 |
| 46 | 0.138873510 | 47 | ท. 2 212149090-01 | 43 |  |
| 49 | 0.215263953 | 50 | 0.205193454 | 53 | 2. 735530397 |
| 5 ? | ग. 757203157 | 53 | n. 121519571. | $5 \wedge$ | $0.0)$ ¢509072ก-01 |
| 55 | 0. $3511.945035-01$ | 57 | ก.20257224? | 57 | ก.939431125 |
| 5 \% | 0.106751000 | 59 | ก.156.599006 | 60 | 9. 239570578 |
| 61 | 0.1971592 .20 | 62 | 0.70 3К69175n-91 | 6.3 | ก.3า753? 1 20n-01 |
| 6.4 | 0.199450610 | 65 | 0.700 ¢0 3415 | 65 | n. 369 ? 56579 |
| ¢7 | 0.119595999 | 62 | ก.124593971 | 69 | 0.951585117D-0? |
| 79 | 0.672.5352390-01 | 71 | ก.2715?4594D-0? | 72 | $9.176740071 \mathrm{r}-\mathrm{7}$ |
| 72 |  | 74 | 0.6767201830-01 | 7.5 | 2. $032757415 \Gamma-01$ |
| 76 | 0.101930120 | 77 | 0.11819453 ? | 78 | ก.131508530 |

F3SCTSSA APPAY

| LOCAITON | VAIUE | TOCATION | $\xrightarrow{\text { VAT, } T J E}$ | LOCATION | y vaide |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 0.577350269 | 2 | 0.0 | 3 | 0.774595669 |
| 4 | 0.339931044 | 5 | 0.861136312 | 6 | 3.0 |
| 7 | 0.538469310 | 9 | 0.906179846 | $\bigcirc$ | 0.239679186 |
| 10 | 0.651209336 | 11. | 0.932469514 | $1 ?$ | 0.0 |
| 13 | 0.405845151 | 14 | 0.741531136 | 15 | 0.949107912 |
| 16 | 0.133131642 | 1.7 | 0.525532 .410 | 19 | ?.796565177 |
| 19 | 0.960289 .956 | 20 | 0.0 | 27 | 0.324253423 |
| 22 | 0.613371433 | 23 | 0.936031107 | 24 | $\bigcirc .969150240$ |
| 25 | 0.148874339 | 25 | 0.133395394 | 27 | 0.679403568 |
| 28 | 0.865063367 | 29 | 0.973906579 | 30 | 0.0 |
| 31 | 0.269543156 | 32 | 0.519096129 | 33 | 0.730152005 |
| 34 | 0.887062600 | 35 | 0.9782 .8659 | 36 | 0.7257 .33409 |
| 37 | 0.367831499 | 38 | 0.587317954 | 39 | 0.759902674 |
| 40 | 0.9041172 .56 | 41 | 0.981560634 | 42 | 0.0 |
| 43 | 0.2304583 .16 | 44 | 0.443492751 | 45 | 0.642349339 |
| 46 | 0.901578091. | 47 | 0.917598399 | 4.8 | 0.90 .4183055 |
| 49 | 0.108054949 | 50 | 0.319112369 | 51 | 0.575248536 |
| 52 | 0.687292905 | 53 | 0.827201315 | 54 | 0.978434984 |
| 55 | 0.935283809 | 56 | 0.0 | 57 | 0.901194094 |
| 58 | 0.394151 .347 | 59 | 0.570972173 | 60 | 0.724417731 |
| 61 | 0.848206583 | 62 | 0.937273392 | 63 | 0.987992519 |
| 64 | $0.9501250980-0]$. | 65 | 0.781603551 | 65 | 0.459016778 |
| 67 | 0.617876244 | 68 | 0.75540440 ? | 69 | 0.06563170 ? |
| 70 | 0.941575023 | 7.2 | 0.9894009 .35 | 72 | 0.093128593 |
| 73 | 0.963971927 | 74 | 0.912334428 | 75 | 0.030116972 |
| 76 | 0.746331906 | 77 | 0.636053681 | 78 | $0.51096700 ?$ |

## FUNCTION BURND

Calling Sequence: $\quad T=$ BURND(DVMG)
Purpose:
BURND computes burn duration as a function of incremental velocity.

Common Blocks Required: C $\quad$ NST, INPUT, MCC $\varnothing M$
Subroutines Called:
None

## Input/Output

| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :--- |
| I | DVMG | 1 | Argument <br> List | Midcourse correction velocity (km/sec) |
| I | G | 1 | CめNST(45) | Earth's surface gravity (km/sec ${ }^{2}$ ) |
| I | WT $\varnothing$ | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | TWD | 10 | INPUT(320) | Times for weight flow rate table (sec) |
| I | WD | 10 | INPUT(350) | Weight flow rate tabular values (kg/sec) |
| I | ASPMC | 1 | INPUT(441) | Specific impulse (sec) |
| I | DW | 10 | MCC $\varnothing M$ | Weight expenditure at TWD (kg) |
| $\varnothing$ | BURND | 1 | Call | Burn duration (sec) |

## Method:

Validity of the rocket equation is assumed. The fuel weight, FUEL, corresponding to the velocity increment, DVMG, is

$$
\text { FUEL }=W T \phi\left(1-e^{-\frac{D V M G}{G \cdot A S P M C}}\right)
$$

A piecewise-linear weight flow rate is assumed. The fuel expended as a function of time is computed under this assumption in MCSET and stored in DW. That is,

$$
D W(I)=\text { fuel expended at } T W D(I)
$$

FUEL is tested against DW until FUEL $\leq \mathrm{DW}(\mathrm{I}+1)$. Then the burn duration is $\mathrm{TWD}(\mathrm{I})$ plus the time since $T W D(I)$, which would cause the fuel to increase by FUEL-DW(I).


Weight flow rate and expended weight versus burn duration

On any segment of the burn history where the weight flow rate, $\dot{w}$, is linear, the slope of the weight flow rate, $\ddot{w}$, is constant. We define the following:

$$
\begin{aligned}
\delta \mathrm{t} & =\mathrm{t}-\mathrm{TWD}(\mathrm{I}) \\
\delta \mathrm{f} & =\mathrm{FUEL}-\mathrm{DW}(\mathrm{I}) \\
\dot{\mathrm{w}}_{\mathrm{o}} & =\mathrm{WD}(\mathrm{I}) \\
\ddot{\mathrm{w}} & =\frac{\mathrm{WD}(\mathrm{I}+1)-\mathrm{WD}(\mathrm{I})}{\operatorname{TWD}(\mathrm{I}+1)-\operatorname{TWD}(\mathrm{I})}
\end{aligned}
$$

The fuel expenditure is a quadratic function of time.

$$
\delta f=\dot{w}_{o} \delta t+\frac{1}{2} \ddot{w} \delta t^{2}
$$

The usual solution of the quadratic,

$$
\delta t=\frac{-\dot{w}_{o} \pm \sqrt{\dot{w}_{o}^{2}+2 \ddot{\mathrm{w}} \delta \mathbf{f}}}{\ddot{\mathrm{w}}}
$$

is ill-defined when $\ddot{w}=0$, so we multiply numerator and denominator by $-\dot{w}_{o} \mp \sqrt{\dot{w}_{o}^{2}+2 \ddot{w} \delta f}$. The result is

$$
\delta t=\frac{2 \delta f}{\dot{w}_{o}+\sqrt{\dot{w}_{o}^{2}+2 \ddot{w} \delta f}}
$$

The sign ambiguity on the radical is removed by observing that for $\ddot{w}=0$, the solution must reduce to

$$
\delta t=\frac{\delta f}{\dot{w}_{\mathrm{o}}}
$$

It may be noted that

$$
\overline{\dot{w}}=\frac{1}{2}\left(\dot{w}_{o}+\sqrt{\dot{w}_{0}^{2}+2 \ddot{w} \delta \mathbf{f}}\right)
$$

is the constant weight flow rate which also would render of in the time interval,万t.

Calling Sequence:

Purpose:

CALL BVE(X, V, U, B, BTR, C3,STHET, CTHET, E, S, RP, T, R, KK)

To compute the miss-vector components of the orbit, relative to the target body, from the Cartesian state.

None
CRめSS, RめTAIT
Input/Output

| I/ $\varnothing$ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PROGRAM DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | X | 3 | Call List | Position vector (km) |
| I | V | 3 | Call List | Velocity vector ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | U | 1 | Call List | Gravitational constant ( $\mathrm{km}^{3} / \mathrm{sec}^{2}$ ) |
| $\emptyset$ | B | 3 | Call List | Miss vector (km) |
| $\emptyset$ | BTR | 2 | Call List | Miss vector (miss-plane components (km) |
| $\emptyset$ | C3 | 1 | Call List | Energy $\left(\mathrm{V}^{2}-\frac{2 \mu}{\mathrm{r}}\right)\left(\mathrm{km}^{2} / \mathrm{sec}^{2}\right)$ |
| $\emptyset$ | STHET | 1 | Call List | Sine of the true anomaly |
| $\emptyset$ | CTHET | 1 | Call List | Cosine of the true anomaly |
| $\emptyset$ | E | 1 | Call List | Eccentricity |
| $\varnothing$ | S | 3 | Call List | Unit asymptote vector |
| $\emptyset$ | RP | 1 | Call List | Radius of periapsis (km) |
| $\emptyset$ | T | 3 | Call List | Equatorial miss-axis |
| $\emptyset$ | R | 3 | Call List | Zenith miss-axis |
| I | KK | 1 | Call List | Asymptote indicator |

## Method:

Subroutine BVE computes the target miss vector, B, from Cartesian input vectors $X$, position, and $V$, velocity.

For hyperbolic orbits, $B$ is the vector from the body center to either the approach or departure asymptote depending on a key set by the user, (see Fig. 1a). For elliptic orbits, the $B$ vector is always the semi-latus rectum for the departing orbit, (see Fig. 1b).

a - Hyperbolic orbit

Figure 1 - B Vector

b-Elliptic orbit
$S$ is a unit vector along the asymptote for the hyperbolic case, and is directed toward periapsis from the body center in the elliptic case.
$T$ is a unit vector taken as

$$
T=\frac{\vec{k} \times \vec{S}}{|k \times S|}
$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the orthogonal unit vectors of the Cartesian input position and velocity vectors X and V .
$R$ is the unit vector given by

$$
\mathrm{R}=\overrightarrow{\mathrm{S}} \times \overrightarrow{\mathrm{T}}
$$

The vectors $B, R$, and $T$ lie in a plane normal to the unit vector $S$ (see Fig. 2).


Figure $2-\mathrm{S}$ Normal to Plane Containing $\overrightarrow{\mathrm{R}}, \overrightarrow{\mathrm{B}}, \overrightarrow{\mathrm{T}}$
For visualization purposes, if $\vec{X}$ and $\vec{V}$ were given to the subroutine with the $i$ and $j$ vectors in the moon's equatorial plane, then the $\vec{T}$ and $\vec{R}$ vectors would be as shown in Fig. 3. The $\vec{R}$ vector doen not lie along the polar axis but, rather, normal to $\vec{S}$ and $\vec{T}$. For lunar approach from Earth, the $\vec{R}$ vector is roughly in the direction of the polar axis, so if one had a trajectory such that $B \cdot R=0$, the plane of the trajectory around the moon should lie close to the equator.


Figure 3 - Miss-Plane Geometry

## Theory:

Given $\vec{X}$ and $\vec{V}$, the position and velocity vectors of the vehicle relative to the central body, and $U$, the gravitational constant for the central body, compute the following:

The semi-latus rectum, $\mathbf{P P}$

$$
\begin{equation*}
P P=|\vec{X} \times \vec{V}|^{2} / U \tag{i}
\end{equation*}
$$

The magnitude of the position vector, $\vec{X}$

$$
\begin{equation*}
\text { RMAG }=|\vec{X}| \tag{2}
\end{equation*}
$$

The energy, C3

$$
\begin{equation*}
\mathrm{C} 3=\overrightarrow{\mathrm{V}} \cdot \overrightarrow{\mathrm{~V}}-2 \mathrm{U} / \mathrm{RMAG} \tag{3}
\end{equation*}
$$

The eccentricity, E

$$
\begin{equation*}
E=[1+(C 3) P P / U]^{\frac{1}{2}} \tag{4}
\end{equation*}
$$

The radial component of velocity

$$
\begin{equation*}
\mathrm{RDOT}=(\overrightarrow{\mathrm{X}} \cdot \overrightarrow{\mathrm{~V}}) / \mathrm{RMAG} \tag{5}
\end{equation*}
$$

The sine and cosine of the true anomaly, $\theta$

$$
\begin{align*}
& \mathrm{CTHET}=\cos \theta=(\mathrm{PP}-\mathrm{RMAG}) /(\mathrm{E} \text { RMAG })  \tag{6}\\
& \mathrm{STHET}=\sin \theta=\operatorname{RDOT}(\mathrm{PP} / \mathrm{U})^{\frac{1}{2}} / \mathrm{E} \tag{7}
\end{align*}
$$

Define a unit vector $\overrightarrow{U X}$ by

$$
\begin{equation*}
\overrightarrow{\mathrm{UX}}=\overrightarrow{\mathrm{X}} / \text { RMAG } \tag{8}
\end{equation*}
$$

and the unit angular momentum vector by

$$
\begin{equation*}
\overrightarrow{U W}=\frac{\vec{x} \times \vec{v}}{|\vec{X} \times \vec{v}|} \tag{9}
\end{equation*}
$$

As shown in Fig. 4, a unit vector $\overrightarrow{U Y}$ normal to $\overrightarrow{U X}$ and $\overrightarrow{U W}$ lying in the orbital plane, and in the direction of rotation is

$$
\begin{equation*}
\overrightarrow{U Y}=\overrightarrow{U W} \times \overrightarrow{U X} \tag{10}
\end{equation*}
$$

The rotation of $\overrightarrow{U X}$ and $\overrightarrow{U Y}$ through $(-\theta)$ yields $\vec{P}$ and $\vec{Q}$, where $\vec{P}$ is a unit vector toward periapsis, and $\vec{Q}$ is a unit vector in the same direction as the semi-latus rectum of the orbit.


Figure 4 - In-Plane Orbit Direction Vectors
Rotate $\overrightarrow{U X}$ and $\overrightarrow{U Y}$ through $(-\theta)$ to $\vec{P}$ and $\vec{Q}$

$$
\begin{align*}
& \overrightarrow{\mathrm{P}}=\overrightarrow{\mathrm{UX}} \cos \theta-\overrightarrow{\mathrm{UY}} \sin \theta \\
& \overrightarrow{\mathrm{Q}}=\overrightarrow{\mathrm{UX}} \sin \theta+\overrightarrow{\mathrm{UY}} \cos \theta \tag{11}
\end{align*}
$$

The $\overrightarrow{\mathrm{S}}$ and $\overrightarrow{\mathrm{B}}$ vectors are computed as follows:
For elliptical orbits

$$
\begin{align*}
& \overrightarrow{\mathrm{S}}=\overrightarrow{\mathrm{P}} \\
& \vec{B}=\vec{Q}|\vec{B}|=\vec{Q} \text { BMAG }=\vec{Q} P P \tag{12}
\end{align*}
$$

For hyperbolic orbits, $\vec{Q}$ and $\vec{P}$ are rotated through an angle of magnitude $(90-\alpha)$ to obtain $\vec{S}$ and $\overrightarrow{U B}$, a unit vector in the direction of the $\vec{B}$ vector. The angle $\alpha$ is the half-angle between asymptotes and is computed by

$$
\begin{align*}
& \text { CALP }=\cos \alpha=1 / \mathrm{E}=\sin (90-\alpha) \\
& \text { SALP }=\sin \alpha=\left(1-1 / \mathrm{E}^{2}\right)^{\frac{1}{2}}=\cos (90-\alpha) \tag{13}
\end{align*}
$$

In the case of the approach hyperbola, $\vec{Q}$ and $\vec{B}$ must be rotated counterclockwise through ( $90-\alpha$ ), while for departure, clockwise. Thus, for approach, with STHET negative (see Fig. 5)

$$
\begin{align*}
& \overrightarrow{\mathrm{S}}=\overrightarrow{\mathrm{Q}} \sin \alpha+\overrightarrow{\mathrm{P}} \cos \alpha \\
& \overrightarrow{\mathrm{UB}}=-\overrightarrow{\mathrm{Q}} \cos \alpha+\overrightarrow{\mathrm{P}} \sin \alpha \tag{14}
\end{align*}
$$

Figure 5 - Hyperbola and Approach Asymptote

For departure, with STHET positive

$$
\begin{align*}
& \overrightarrow{\mathrm{S}}=\sin \alpha \overrightarrow{\mathrm{Q}}-\cos \alpha \overrightarrow{\mathrm{P}} \\
& \overrightarrow{\mathrm{UB}}=\cos \alpha \cdot \overrightarrow{\mathrm{Q}}+\sin \alpha \overrightarrow{\mathrm{P}} \tag{15}
\end{align*}
$$

Finally, the $\vec{B}$ vector is

$$
\begin{equation*}
\overrightarrow{\mathrm{B}}=\mathrm{BMAG} \overrightarrow{\mathrm{UB}} \tag{16}
\end{equation*}
$$

where

$$
\begin{aligned}
& \mathrm{BMAG}=(\mathrm{U} / \mathrm{C} 3+\mathrm{RP}) \sin \alpha \\
& \mathrm{U} / \mathrm{C} 3=\text { the semi-major axis } \\
& \mathrm{RP}=\mathrm{PP} /(1+\mathrm{E})=\text { radius of periapsis }
\end{aligned}
$$

$\vec{B} \cdot \vec{T}$ and $\vec{B} \cdot \vec{R}$ are computed and returned by the subroutine as $\operatorname{BTR}(1)$ and BTR(2) respectively. As seen in Fig. 2, BTR(1) represents BMAG $\cos \psi$ and $\operatorname{BTR}(2)$ represents BMAG $\cos (90-\psi)$.

Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required:

CALL CALEND(T, JYR, JDY, MO, NHR, MIN, SEC)

This subroutine converts the current MAESTRO time to calendar date.

INPUT

None

Input / Output

| SYMBOLIC |  |  |  |  |
| :--- | :--- | :---: | :---: | :--- |
| I/O | NAME | DIMENSION |  |  |
| COMMON |  |  |  |  |
| BLOCK |  |  |  |  |$\quad$ DEFINITION

## Description:

The calendar date is obtained from the modified julian date of state epoch and time since state epoch. Output is year, day, month, hour, minute and seconds.

## SUBROUTINE CLOSE

## Calling Sequence:

Purpose:

CALL CLOSE
Subroutine CLOSE determines the spacecraft's central planet and transforms the state to the central planet if a new planet becomes central.

Common Blocks Required: CNTRL, CONST, INPUT, INTER, PLNET, STATE Subroutines Required: DVMAG, INTEG, ORBIT, TRMN, UPDATE

## Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I/O | DST | 12 | PLNET(73) | Distances from s/c <br> to planets |
| I/O | DX | 3 | STATE(4) | Velocity of s/c |
| O | ELM | 6 | STATE(14) | Orbital elements of s/c |
| I | GM | 12 | CONST(5) | Planet gravitational <br> constants |
| O | INT | 1 | INTER(131) | Interpolation counter |
| I/O | JC | 1 | CNTRL(7) | Central planet number |
| O | KDIS | 1 | CNTRL(5) | Discontinuity flag |
| I | KP | 12 | INPUT(1001) | Planets in the system |
| I | METH | 1 | INPUT(1013) | Propagator method |
| I | RSWTCH | 12 | INPUT(500) | Distances when planets <br> become central |
| I/O | X |  | 3 | STATE(1) | | Position of s/c |
| :--- |
| I/O |

## Description:

There are two major parts to this subroutine. In the first part CLOSE determines which planet's sphere of influence the spacecraft resides in. The sphere of influence is denoted by RSWTCH. If the spacecraft's distance from planet $J$ is less than RSWTCH $(J)$, then planet $J$ is central. If that planet is already the central planet, then CLOSE returns. If that planet is different than the central planet, then that planet becomes the central planet.

The second part of the subroutine takes care of the changes in the system due to the change in central planet. The position and velocity vectors of the planets and the spacecraft are changed to correspond to position and velocity relative to the new central planet. KDIS is set to one to denote a discontinuity in the system. Depending on the propagation scheme (METH), new orbital elements are calculated if needed, and INTEG is called to set up new values and derivatives of integration variables. INT is set to zero and then UPDATE is called so that the interpolation table is cleared and a new one is started.


## SUBROUTINE CONTRL

## Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required:

CALL CONTRL
This subroutine initializes the compute interval table and other constants before initiation of the program options

ANKOR, CONST, INPUT, STATE
DVMAG, PRINT, SETUP2

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| 0 | ANKVEC | 6 | ANKOR(1) | Initial anchor vector position and <br> velocity vectors |
| 0 | ATT | 3 | STATE(11) | Unit vector along the <br> spacecraft's centerline |
| 0 | DECO | 1 | INPUT(48) | Initial declination |
| 0 | DELT | 10 | INPUT(180) | Table of compute intervals |
| I | KMETH | 1 | INPUT(1036) | Initial trajectory propagation <br> indicator |
| I | RAD | 1 | CONST(1) | Radian-degree conversion factor |
| O | RAO | 1 | INPUT(47) | Initial right ascension |
| O | TCOMP | 10 | INPUT(170) | Array of switching times used in <br> compute interval table. |
| I | X | 6 | STATE(1) | Initial position and velocity <br> vectors |

## Description:

This subroutine initializes program constants and flags before initiation of the program options. Subroutine SETUP2 is called at the beginning of this routine. Most of the initialization is performed in SETUP2. The compute interval table is also established if one is not already input. The compute interval table is established according to the
initial spacecraft radius, $R$, as shown in the following table.

$$
\begin{array}{ll}
\mathrm{R}<20,000 & \\
\operatorname{DELT}(1)=300 . & \operatorname{TCOMP}(1)=3600 . \\
\operatorname{DELT}(2)=1800 . & \operatorname{TCOMP}(2)=18000 . \\
\operatorname{DELT}(3)=18000 . & \operatorname{TCOMP}(3)=1(10)^{20} \\
20000<\mathrm{R}<40000 & \\
\operatorname{DELT}(1) & =1800 . \\
\operatorname{DELT}(2) & =18000 .
\end{array}
$$

If the initial spacecraft attitude is not input, the attitude is set in the same direction as the initial velocity vector. The subroutine next calls subroutine PRINT in order to obtain a list of the initial input array.

## SUBROUTINE COVERT

Calling Sequence:
Purpose:

CALL COVERT (COV, X, Q, KCOV)
This subroutine transforms a covariance matrix from a local tangent plane coordinate system to the coordinate system of the input state which defines the local system.

None
CROSS, VNORM

Input/Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| I | COV | 6,6 | Calling <br> Operand | Input covariance matrix |
| I | KCOV | 1 | Calling <br> Operand | Transformation flag. If KCOV is not <br> zero, the covariance matrix will be <br> transformed |
| O | Q | 6,6 | Calling <br> Operand | Output covariance matrix |
| I | X | 6 | Calling <br> Operand | Position and velocity vectors of <br> the state |

## Description:

The local tangent plane coordinate system is defined as follows:

> X-axis Positive along the position vector

Y-axis Lies in the plane formed by the position and velocity vectors and normal to the position vector positive in the direction of the velocity. It is represented vectorially by

$$
(\overline{\mathrm{X}} \times \overline{\mathrm{V}}) \times \overline{\mathrm{X}}
$$

Z-axis Normal to the plane formed by the position and velocity vectors. It is represented vectorially by

$$
\overline{\mathrm{x}} \times \overline{\mathrm{v}}
$$

This subroutine determines the rotation matrix from the local tangent plane to the coordinate system of the state $\overline{\mathrm{X}}$ and $\overline{\mathrm{V}}$. To calculate the transformation matrix, T , it is first necessary to establish the vectors

$$
\begin{aligned}
& \hat{S}=\bar{X} \times \bar{V} /(|\bar{X}||\bar{V}|) \\
& \hat{u}=\bar{S} \times \bar{x} /|\bar{x}|
\end{aligned}
$$

Then the first column of the matrix is a unit vector along $\overline{\mathrm{X}}$. The second column is composed of vector $\hat{u}$, while the last column consists of vector $\hat{S}$.

The covariance matrix ts transformed to the new coordinate system using the following relationship,

$$
[\mathrm{Q}]=[\mathrm{T}][\mathrm{COV}][\mathrm{T}]^{\mathrm{t}}
$$

If KCOV is zero, no transformation is performed and the output matrix, $Q$, is set equal to the input matrix, COV.

SUBROUTINE CRASH

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required;

CALL CRASH
Subroutine CRASH determines the time of closest approach to the target planet.

CNTRL, INPUT, INTVAR, PLNET, SAVE, STATE

DVMAG, INTERP, PLANET

Input/Output

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |

## Theory:

The time of closest approach is determined by an iterative process using the flight path angle as the dependent variable and time as the independent variable. The time is adjusted in order to drive the flight path angle to zero. A Newton-Raphson type iteration is employed to determine the time. The sine of the flight path angle at time, t , is determined from :

$$
\begin{equation*}
R 3=(\overline{\mathrm{x}} \cdot \overline{\mathrm{v}}) /|\overline{\mathrm{x}}| \quad|\overline{\mathrm{v}}| \tag{1}
\end{equation*}
$$

where $\bar{X}$ and $\stackrel{\rightharpoonup}{V}$ are the position and velocity vectors at time $t$ determined from subroutine INTERP.

The derivative of this sine with respect to time is numerically determined by calculating the sine at two times as described in equation (1) and dividing their difference by the difference in time as,

$$
\begin{equation*}
\text { DERIV }=(\mathrm{R} 2-\mathrm{R} 3) / \text { DELT } \tag{2}
\end{equation*}
$$

where R2 is the sine of the flight path angle at $T+$ DELT. The change in time to drive the sine of the flight path angle to zero is calculated from:
DEL = R3 / DERIV

The sine of the flight path angle is determined at time $T=D E L$. The iteration is assumed to converge if the flight path angle is within a small tolerance around zero. If not, the derivative is recalculated in equation (2) with $T=T 2+D E L$ and the process repeated. A limit of 20 iterations are allowed. A limit to the size of the step, DEL, is also employed to help assure convergence.

## Description:

The time of closest approach is determined through a double iteration using this routine along with subroutines INTERP and TIMEC. There is an iteration inside subroutine CRASH to determine the time of closest approach using subroutine

INTERP. In this iteration the spacecraft's state at time, T, is determined using the interpolation logic in subroutine INTERP. T is adjusted until the spacecraft's flight path angle is within a specified tolerance. A Newton-Raphson type iteration discussed in the theory is employed to determine this time. The second iteration loop is associated with subroutines TIMEC and CRASH.

The time of closest approach is transfered to TIMEC. The time of closest approach is considered a discontinuity time in subroutine TIMEC. Thus, if the current time is greater than the time of closest approach, the state is restored to the values at the last step and the compute interval is adjusted to integrate to the time of closest approach. The time of closest approach is recalculated in subroutine CRASH at the discontinuity time. If the new time of closest approach is within a tolerance of the last time of closest approach, it is assumed that convergence has been achieved on the time of closest approach. If not, the new time of closest approach is used as a discontinuity time in TIMEC and the iteration repeated. The iteration between TIMEC and CRASH usually converges in one or two iterations. This iteration would not be necessary if the interpolation logic exactly matched the numerical integrator. KCA is a counter used to determine the number of TIMEC-CRASH iterations. A limit of 7 iterations are allowed.


Determine time of closest approach using
Newton-Raphson scheme, equations (2) and (3)



## Description:

The vector cross product is determined by this subroutine. The cross product is determined by

$$
\begin{aligned}
& \mathrm{Y}_{1}=\mathrm{X}_{2} \cdot \mathrm{Y}_{3}-\mathrm{X}_{3} \mathrm{Y}_{2} \\
& \mathrm{Y}_{2}=\mathrm{X}_{3} \mathrm{Y}_{1}-\mathrm{X}_{1} \mathrm{Y}_{3} \\
& \mathrm{Y}_{3}=\mathrm{X}_{1} \mathrm{Y}_{2}-\mathrm{X}_{2} \mathrm{Y}_{1}
\end{aligned}
$$

## SUBROUTINE DATE

Calling Sequence:
Purpose:

CALL DATE (YEAR, DAY, QM, HR, DJO)
DATE converts a calendar date to its julian date.

Common Blocks Required: None
Subroutines Required: None

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINTION |
| :--- | :---: | :---: | :--- | :--- |
| I | DAY | 1 | CALLING <br> OPERAND | Day of the month |
| 0 | DJO | 1 | CALLING <br> OPERAND | Julian date |
| I | HR | 1 | CALLING <br> OPERAND | Hour of the day |
| I | QM | 1 | CALLING <br> OPERAND | Month |
| I | YEAR | 1 | CALLING <br> OPERAND | Year |

## Description:

Date calculates the number of days since 1900 and adds it to the modified julian date of 1900 to get the actual modified julian date.

## SUBROUTINE DOPLER

Calling Sequence:
Purpose:

## CALL DOPLER

This subroutine determines the velocity away from the visible tracking sites and determines the doppler frequency shift.

Common Blocks Required: CNTRL, CONST, INPUT, OBSTT, STATE
Subroutines Required: DVMAG, M50EPM, ROTATE, ORBIT, INTEG, UPDATE

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | CAR1 | 1 | INPUT(408) | Spacecraft's primary carrier frequenct |
| I | CAR2 | 1 | INPUT(409) | Spacecraft's secondary carrier <br> frequency |
| I | DJ0 | 1 | INPUT(46) | Modified julian date of <br> state epoch |
| I | DJ1 | 1 | INPUT(37) | Modified julian date of liftoff epoch |
| I | DOBS | 10,2 | OBSIT(1) | Velocity of the tracking stations due to <br> the Earth's rotation |
| I | DX | 3 | STATE(4) | Spacecraft's velocity vector |
| I | HR | 1 | INPUT(53) | Hour of state epoch <br> I <br> KWTDOP |
| I | OBSLAT | 10 | CNTRL(10) | Doppler first pass flag. Zero on <br> first pass |
| I | OBSLON | 10 | INPUT(480) | Latitudes of the tracking stations |
| I | SEC | 1 | INPUT(4I0) | Longitudes of the tracking stations |
| I | T | 1 | INPUT(55) | Seconds of state epoch |
| I | TBO | 1 | STATE(10) | Seconds since state epoch |
| I | THRUST | 1 | STATE(33) | Engine thrust |


| SYMBOLIC |  |  | COMMON |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: |
| I/O NAME DIMENSION BLOCE DEFINITION <br> I TIG 1 INPUT(380) Ignition time of engine 1. <br> I X 3 STATE(1) Spacecraft's position vector <br> I XMIN 1 INPUT(54) Minutes of state epoch <br> I XOBS 10,3 OBSTT(21) Vectors from center of Earth to trackin <br> stations in Earth equator and Greenwic <br> I UJT 1 STATE(32) Current modified julian date <br> I W 1 STATE(35) Current spacecraft mass |  |  |  |  |  |

## Description:

The spacecraft's velocity with respect to a rotating Earth must be determined in order to calculate the velocity away from a tracking station. Subroutine M50EPM is used to determine the transformation to the Earth equator and Greenwich. The vector from the observation site to the spacecraft is established from

$$
\begin{equation*}
\overline{X O B}=\overline{X E}-\overline{X O B S} \tag{1}
\end{equation*}
$$

where $\overline{\mathrm{XE}}$ is the position vector of the spacecraft in the Earth equator and Greenwich, and
$\overline{\mathrm{XOBS}}$ is the vector to the tracking site obtained from OBSIT common.

The spacecraft is visible from the tracking station if

$$
\begin{equation*}
\overline{\mathrm{XOB}} \cdot \overline{\mathrm{XOB}}>0 \tag{2}
\end{equation*}
$$

On the first pass through this subroutine (KWTDOP $=0$ ), KSTAT (I), where I corresponds to the tracking station, is set to one if the spacecraft is visible. This array is used to output the information only for the visible tracking stations. The velocity relative to the tracking station is obtained from

$$
\begin{equation*}
\overline{\mathrm{V}}=\overline{\mathrm{DX}} \mathrm{E}-\overline{\mathrm{DOB}} \mathrm{~S} \tag{3}
\end{equation*}
$$

where $\overline{\mathrm{DXE}}$ is the velocity of the spacecraft in the Earth equator and Greenwich, and
$\overline{\mathrm{DOBS}}$ is the velocity of the tracking station from OBSIT common.

Note: The $z$-component of $\overline{\mathrm{V}}$ is equal to z -component of $\overline{\mathrm{DXE}}$ since the z-component of $\overline{\mathrm{DOBS}}$ is zero.

Finally, the velocity away from the tracking station is determined from

$$
\begin{equation*}
\overline{\mathrm{RDOT}}=(\overline{\mathrm{V}} \cdot \overline{\mathrm{XOB}}) /|\overline{\mathrm{XOB}}| \tag{4}
\end{equation*}
$$

and the doppler shift is obtained from

$$
\begin{aligned}
\text { DOP }= & \left(\text { CAR }^{2}-\mathrm{CAR1}^{2}\right) \text { RDOT } / \mathrm{C} \\
\text { where } & \text { CAR1 and CAR2 are the spacecraft carrier frequencies, and } \\
& \mathrm{C} \text { is the velocity of light }
\end{aligned}
$$

The spacecraft's velocity away from each of the visible tracking stations and the corresponding doppler shift along with the current thrust and mass are output on unit 6. The KWTDOP flag is used to control the writing of the heading for the writes.




Calling Sequence:
Puxpose:

Common Blocks Required:
Subroutines Required:
$\mathrm{Z}=\operatorname{DOT}(\mathrm{X}, \mathrm{Y})$
This function performs the vector dot product

None

None

Input / Output

| SYMBOLIC |  |  |  | COMMON |  |  |
| :--- | :--- | :--- | :---: | :--- | :---: | :---: |
| I/O | NAME | DIMENSION | BLOCK | DEFINITION |  |  |
| $O$ | DOT | 1 | FUNCTION <br> NAME | Vector dot product |  |  |
| I | X | 3 | CALLING <br> OPERAND | Input vector $X$ |  |  |
| I | Y | 3 | CALLING <br> OPERAND | Input vector $Y$ |  |  |

## Description:

The vector dot product is obtained from

$$
\mathrm{DOT}=X_{1} Y_{1}+X_{2} Y_{2}+X_{3} Y_{3}
$$

where

$$
X_{i}, Y_{i} \quad i=1,3 \text { are the components of the } X \text { and } Y \text { vectors, respectively }
$$

## SUBROUTINE DRAG

## Calling Sequence:

## Purpose:

Common Blocks Required:

Subroutines Required:

CALL DRAG
This subroutine calculates the spacecraft acceleration due to atmospheric drag.

CNTRL, CONST, GRAVTY, INPUT, PERT, PLNET.

ATMO, VNORM.

Inputs/Outputs

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | DST | 12 | PLNET (73) | The JC element is the <br> spacecraft distance from the <br> central planet. |
| I | JC | 1 | CNTRL (07) | Central planet number |
| I | KATMOS | 1 | INPUT (1097) | Drag flag |
| I | POS | 3 | GRAVTY (1) | Position vector from central <br> planet. |
| I/O | RCART | 3 | PERT (1) | Spacecraft perturbing <br> acceleration |
|  | RE | 12 | CONST (17) | Equatorial radius of the <br> planet |

## Description:

This subroutine determines the acceleration due to atmospheric drag. The magnitude of the acceleration is obtained from:

$$
a=\frac{1}{2} \rho V^{2} C_{D}(10)^{6}
$$

where $\rho$ is the density in $\mathrm{g} / \mathrm{CM}^{3}$
V is spacecraft velocity in $\mathrm{km} / \mathrm{sec}$
$C_{D}$ is the drag-area-mass coefficient
and the multiplier (10) ${ }^{6}$ is required to convert units. The acceleration is applied along the velocity vector. The density is obtained from the function ATMOS while the drag coefficient is set at . 03 .

| Calling Sequence: | $Y=$ DVMAG (X) |
| :--- | :--- |
| Purpose: | This function determines the magnitude of <br> an input vector |
| Common Blocks Required: | None |
| Subroutines Required: | None |

Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \\ & \hline \end{aligned}$ | DIMENSION | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | DVMAG | 1 | $\begin{aligned} & \text { FUNCTION } \\ & \text { NAME } \\ & \hline \end{aligned}$ | Magnitude of the input vector |
| I | X | 3 | CALLING OPERAND | Input vector |

## Description:

The magnitude of a vector is determined from
DVMAG $=\sqrt{\mathrm{x}_{1}^{2}+\mathrm{x}_{2}^{2}+\mathrm{x}_{3}^{2}}$
where

$$
X_{i} \quad i=1,3 \text { are the components of the input vector } X
$$

## SUBROUTINE EQNS

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:

CALL EQNS
This subroutine calculates the derivatives of the variables being numerically integrated.

CNTRL, CONST, GRAVTY, INPUT, INTVAR, INTVRX, PERT, STATE.

ACCEL, AVEQNS, ORBIT, GRAV

Input / Output
-

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | DJO | 1 | INPUT(46) | Julian date of state epoch |
| I | GM | 12 | CONST(5) | Gravitational constants |
| I | KP | 12 | INPUT(1001) | Planets in the system |
| I | METH | 1 | INPUT(1013) | Trajectory propagator indicator |
| O | RATES | 6 | INTVAR(8) | Derivatives of integration <br> variables |
| I | RCART | 3 | PERT(1) | Disturbing acceleration <br> I <br> X |
| Y | 1 | INTVAR(1) | Current independent variable |  |
| I | UJT | 1 | INTVAR(2) | Current dependent or <br> integration variables |

Theory:
This subroutine calculates the derivatives of the integration variables at the current time when propagating the state using any of the following trajectory propagation methods,

1. Cowell
2. Encke
3. NICE/True
4. NICE/Mean
5. NICE/e $\cos \omega, \mathrm{e} \sin \omega$.

The equations which define the derivatives of each of these methods are presented below.

## Cowell

Let the origin of the cartesian coordinate system be located at the central planet of mass $M$. The disturbing planets are denoted by $M_{i}$, and the spacecraft by $m$. Then the perturbing acceleration becomes,

$$
\begin{equation*}
\vec{R}=\sum_{i=1}^{n} G M_{i}\left[\frac{\bar{X}_{i}^{\prime}-\bar{X}_{i}}{\rho^{3}}-\frac{\bar{X}^{\prime}}{\left|\vec{X}^{\prime}\right|}\right]+\bar{a}_{S P}+\bar{a}_{T}+\bar{a}_{O B} \tag{1}
\end{equation*}
$$

where
$\bar{X}^{\prime}$ is the vector from the central planet to the disturbing planet,
$\overline{\mathrm{X}}$ is the vector from the central planet to the spacecraft,
$\overline{\mathrm{a}}_{\mathrm{SP}}$ is the acceleration due to solar pressure,
$\overline{\mathrm{a}}_{\mathrm{T}}$ is the acceleration due to engine thrusting,
$\overline{\mathrm{a}}_{\mathrm{OB}}$ is the acceleration due to an oblate planet, and
$\rho$ is $\left|\bar{X}^{\prime}-X\right|$
The total acceleration acting on the spacecraft is the sum of the perturbing acceleration and the acceleration due to the central planet; thus the total acceleration becomes,

$$
\begin{equation*}
\overline{\mathrm{a}}=\frac{G M \bar{X}}{\mathbf{r}^{3}}+\overline{\mathrm{R}} \tag{2}
\end{equation*}
$$

The above vector equation denotes the acceleration, which is numerically integrated to obtain the velocity. The velocity is numerically integrated to obtain the position.

## Encke's Method

In this method an attempt is made to utilize the knowledge that the motion is very nearly two-body with respect to the central planet. Thus, only the motion which deviates from the two-body motion is integrated. This motion is added to the twobody motion to obtain the position and velocity of the spacecraft.

Let $\bar{X}_{0}$ be the position vector of the spacecraft obtained from two-body motion, and $\bar{\xi}$ be a vector describing the deviation from the two-body orbit. The position vector of the spacecraft is then obtained by,

$$
\begin{equation*}
\overline{\mathrm{x}}=\overline{\mathrm{x}}_{0}+\bar{\xi} \tag{3}
\end{equation*}
$$

The acceleration of the disturbing vector, $\bar{\xi}$, is obtained as follows: If the quantity $f$ is defined as

$$
\begin{equation*}
f=1-(1+2 q)^{-3 / 2} \tag{4}
\end{equation*}
$$

where

$$
\begin{equation*}
q=\left[\sum_{i=1}^{3}\left(\vec{x}_{0_{i}}+\frac{1}{2} \vec{\xi}_{i}\right) \vec{\xi}_{i}\right]\left(\frac{1}{\left|\vec{x}_{0}\right|^{2}}\right) \tag{5}
\end{equation*}
$$

then the disturbing acceleration is

$$
\begin{equation*}
\vec{\xi}=\frac{\mu}{\left|\mathrm{x}_{0}\right|^{3}} \quad(\mathrm{f} \overrightarrow{\mathrm{X}}-\bar{\xi})+\overline{\mathrm{R}} \tag{6}
\end{equation*}
$$

where $\vec{R}$ is obtained from Equation (1). This vector is numerically integrated to obtain the deviation from the two-body orbit. The reference orbit is updated (rectified) whenever

$$
\begin{equation*}
\frac{|\vec{x}|^{2}}{|\vec{x}|_{0}^{2}}>0.001 \tag{7}
\end{equation*}
$$

These equations are much more complicated than Cowell's equations. Also, to obtain the position along the reference orbit, one must calculate the true anomaly from the mean anomaly. This involves an iterative solution and is time consuming.

## NICE Methods

The three NICE methods involve the numerical integration of the classical orbital elements to obtain the orbit of the spacecraft as a function of time.

The orbital elements integrated are,

1. semilatus rectum, $\mathbf{p}$
2. eccentricity, e
3. true or mean anomaly, for M
4. argument of perigee, $\omega$
5. inclination, i
6. longitude of ascending node, $\Omega$
or, alternately
7. p
8. e $\cos \omega$
9. e $\sin \omega$
10. $\mathrm{f}+\omega$
11. i
12. $\Omega$

The derivatives of the above quantities are determined and numerically integrated to determine the instantaneous orbital elements. The derivatives of the orbital elements are derived in many texts and reports and only the results will be presented here. The derivatives of the orbital elements are

$$
\begin{align*}
& \dot{p}=\left(2 r \sqrt{\frac{p}{\mu}}\right) C  \tag{8}\\
& \dot{e}=\sqrt{\frac{p}{\mu}}\left\{R \sin f+\frac{r}{p}\left[2 \cos f+e\left(1+\cos ^{2} f\right)\right] C\right\}
\end{align*}
$$

$$
\begin{aligned}
& \dot{\omega}=\sqrt{\frac{p}{\mu}}\left\{\frac{\sin f}{e}\left(1+\frac{r}{p}\right) C-\left(\frac{\cos f}{e}\right) R-\left(\frac{r}{p} \sin u \cot i\right) w\right\} \\
& \dot{\Omega}=r \sin u W /(\sin i \sqrt{\mu p}) \\
& i=\left(\frac{r}{\sqrt{p \mu}} \quad \cos u\right) W \\
& \stackrel{f}{f}=\frac{\sqrt{\mu p}}{2}+\frac{1}{e} \sqrt{\frac{p}{\mu}}\left\{\cos f R-\sin f\left(1+\frac{r}{p}\right) C\right\} \\
& \dot{M}=n+\sqrt{p \frac{\left(1+e^{2}\right)}{\mu}}\left\{\left(\frac{\cos f}{e}-\frac{2 r}{p}\right)-R \frac{\sin f}{C}\left(1+\frac{r}{p}\right) \quad C\right\} \\
& (e \sin \omega)=\sqrt{\frac{p}{\mu}}\left\{-\cos \mu R+\left[\left(1+\frac{r}{p}\right) \sin u+e \frac{r}{p}\right] C\right. \\
& \left.+\frac{r}{p} \text { e cos } \omega \sin u \cot i \quad W\right\} \\
& (e \cos )=\sqrt{\frac{p}{\mu}}\left\{\sin u R+\left[\left(1+\frac{r}{p}\right) \cos u+e \frac{r}{p} \sin \omega\right] C\right. \\
& \left.+e \frac{r}{p} \sin u \cot i w\right\} \\
& \dot{u}=\frac{\mu}{r^{2}} \sqrt{\frac{p}{\mu}}-\frac{r}{p} \sqrt{\frac{p}{\mu}} \sin u \cot i \quad W
\end{aligned}
$$

where
$n$ is the mean motion
$\mu$ is the gravitational potential
$\mathbf{u}=\boldsymbol{\omega}+\mathbf{f}$
$r$ is the radius
$R, C, W$ are the perturbing accelerations

The perturbing accelerations are written with respect to the orbit plane. They are in the radial direction, circumferential direction and normal to the orbit plane. These accelerations are obtained from the perturbing acceleration derived in Equation (1) by rotation to the orbit plane, as

$$
\left\{\begin{array}{l}
R  \tag{9}\\
C \\
W
\end{array}\right\}=[A]\left\{\begin{array}{l}
R_{1} \\
R_{2} \\
R_{3}
\end{array}\right\}
$$

where
$[A]\left[\begin{array}{lll}\cos \Omega \cos u-\sin \Omega \sin u \cos i & \sin \Omega \cos u+\cos \Omega \sin u \cos i & \sin u \sin i \\ -\cos \Omega \sin u-\sin \Omega \cos u \cos i & -\sin \Omega \sin u+\cos \Omega \cos u \cos i & \cos u \sin i \\ \sin \Omega \sin i & -\cos \Omega \sin i & \cos i\end{array}\right]$

The difference between the NICE/Mean and NICE/True methods is that the NICE/Mean method uses the mean anomaly equation of (8) while in the NICE/TRUE method, the true anomaly is used. When the last of the NICE methods are used, the last 3 equations of equation (8) are used instead of the equations for $\dot{e}, \dot{\boldsymbol{\omega}}$ and $\dot{\mathbf{f}}$.

## Description:

The derivatives of the integration variables are as described in the above equations. The set of equations used is determined by the METH flag as follows: .

$$
\begin{aligned}
\text { METH }= & \text { 1. } \text { Cowell } \\
& \text { 2. Encke } \\
& \text { 3. } \text { NICE/True } \\
& \text { 4. NICE/Mean } \\
& \text { 7. }
\end{aligned}
$$

The current set of integration variables and the independent variable, time, is brought into the subroutine via INTVAR common. This common block is initiated in subroutine INTEG. The Encke reference orbit is input via PERT common. The disturbing acceleration is calculated in subroutine ACCEL and transferred to EQNS via PERT common.

The logic flow consists of testing the METH flag to determine which set of equations to calculate. The derivatives are determined in a straightforward manner using the equations described in the theory.





Calling Sequence:
CALL FIELD2
Purpose:
FIELD2 uses the chain rule to calculate the partial derivatives of the gravitational potential.

Common Blocks Required: CNTRL, CONST, FIELDM, GRAVTY, INPUT, INTVAR, PERT, STATE

Subroutines Required: M50JPM, ROTATE, SPNM
Reference:
Gulick, L.J., "A Comparison of Methods for Computing Gravitational Potential Derivative," ESSA TECHNICAL REPOKI C \& GS 40, 1970.

Input/Output
$\left.\begin{array}{|ll|l|l|l|}\hline \hline \text { I/O } & \begin{array}{c}\text { SYMBOLIC } \\ \text { NAME }\end{array} & \text { DIMENSION } & \text { COMMON } \\ \text { BLOCK }\end{array}\right]$ DEFINITION

| I | UJT | 1 | STATE(32) | Modified Julian date |
| :--- | :---: | :---: | :---: | :--- |
| I | WP | 12 | CONST(29) | Planet spin rates |
| I | XJ | 16 | FIELDM(1) | Zonal coefficients |

## Theory:

The gravitational potential of the central planet in terms of spherical harmonics may be expressed as

$$
\begin{aligned}
V= & \frac{G M}{r}\left\{1+\sum_{n=1}^{\infty}\left(\frac{R}{r}\right)^{n} \sum_{m=0}^{n} P_{n}^{m}(\sin \beta) x\right. \\
& \left(C_{n}^{m} \cos (m \lambda)+S_{n}^{m} \sin (m \lambda)\right\}
\end{aligned}
$$

where GM is the gravitational constant, $R$ is the equatorial radius of the central planet, $C_{n}{ }^{m}$ and $S_{n}{ }^{m}$ are the coefficients representing the mass distribution, $P_{n}^{m}(\sin \beta)$ is the associated Legendre polynomial of degree $n$ and order $m$, and $\beta, \lambda$, and $\mathbf{r}$ are the body-fixed latitude, longitude, and radius of the point where the disturbing force is to be evaluated.

The force due to this potential is the partial derivative of $V$ with respect to inertial cartesian coordinates. One method of evaluating this force is by using the chain rule to evaluate the derivatives with respect to the body cartesian coordinates and then rotating to the integration (inertial) frame. This chain rule method incorporates the following recursion relationship for $P_{n}^{\prime m}(x)$

$$
\frac{d P_{n}^{m}(x)}{d x}=-\frac{\operatorname{mxP}_{n}^{m}(x)}{1-x^{2}}+\frac{P_{n}^{m+1}(x)}{\left(1-x^{2}\right)^{\frac{1}{2}}}
$$

See Gulick (referenced above) for the derivation of this relationship and for further notes on the chain rule method.

## Description:

The main purpose of FIELD2 is to calculate the partial derivatives of the gravitational potential. These derivatives are used to evaluate the acceleration of the $\mathrm{S} / \mathrm{C}$ at any given position. To use FIELD2 for this purpose, KVAR should be set to zero. FIELD2 will calculate the acceleration of the $\mathrm{S} / \mathrm{C}$ due to the central planet using the input field ( CC and XJ ), the position of the $\mathrm{S} / \mathrm{C}$, and the time. The resulting acceleration vector will be rotated to the integration frame using the transformation matrix SELNEQ, and then added to RCART.

In addition to this function, FIELD2 may also be used to calculate the gradient of the force field and the partial derivatives of the force with respect to the spherical har-monic coefficients of the field. To use this mode of FIELD2, KVAR must be set to one, in which case the acceleration (described above) will still be output, along with the matrices D2 VDX2 and DFDCX.

The symmetric matrix $\mathrm{D}^{2}$ VDX2 contains the second partials of the potential with respect to the body-centered cartesian coordinates. For a description of the use of this Jacobian matrix in linear variational theory, see subroutine SHIMMY.

The matrix DFDCX contains the explicit partial derivatives of the force with respect to the NB different harmonic coefficients to be estimated. The maximum value of NB is 100 . The input vector IX determines which coefficients are to be studied, and must be set up in the following way. The $i^{\text {th }}$ element of IX indicates the $i^{\text {th }}$ coefficient to be studied, and $\operatorname{IX}(i)$ must be equal to the position of the $i^{\text {th }}$ coefficient in FIELDM common. The order of the values in IX is crucial since the derivatives are calculated in the same order as the coefficients are used in FIELD2. This order is (( $\left.C_{i j}, S_{i j}, j=0, i\right), i=1$, NMOD). The element $C_{i j}$ occupies position $16 \times j+i$ of FIELDM common and $S_{i j}$ occupies position (i+1) $\times 16+j$ 。 Thus, if $\mathrm{C}_{20}, \mathrm{C}_{22}, \mathrm{C}_{31}$, $\mathrm{C}_{32}, \mathrm{C}_{41}, \mathrm{~S}_{21}, \mathrm{~S}_{32}, \mathrm{~S}_{33}$ are the coefficients to be studied, then $\mathrm{NB}=8$ and IX would be $2,49,34,19,35,66,67,20$ since the calculations would be made in the order $\mathrm{C}_{20}, \mathrm{~S}_{21}, \mathrm{C}_{22}, \mathrm{C}_{31}, \mathrm{C}_{32}, \mathrm{~S}_{32}, \mathrm{~S}_{33}$ and $\mathrm{C}_{41}$. If $\mathrm{C}_{33}$ were to be studied also, then NB would be 9 and 51 would have to be inserted between 66 and 67 in the vector IX. The order that the derivatives are output in DFDCX is the same as the order in IX.


Calling Sequence: CALL FIND (IDSAT, ISET, \$)
Purpose:
This subroutine reads a file from the GTDS 24 -hour hold file to retrieve the state from the GTDS program.

Common Blocks Required: ELMNT
Subroutines Required: None

Input / Output

|  |  |  |  |  |
| :--- | :---: | :---: | :---: | :--- |
| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| I | IDSAT | 1 | CALLING <br> OPERAND | Satellite identification number |
| I | ISET | 1 | CALLING <br> OPERAND | Element set number of <br> desired data |
| $O$ | $\$$ | - |  | Statement number to transfer <br> to if error return. |

## Description:

This subroutine is used to retrieve the state and covariance matrix from the 24-hour hold file written by the GTDS program. The data is read from unit 26 using a direct read. The record number used in the direct read is determined from the element set number brought in via the argument list and variables defining the 24-hour hold file obtained from a read to unit 1.

The data is read into a working array and later transferred to the SET array of ELMNT common. ELMNT common is used to transfer the data to other subroutines in MAESTRO. The SET array is defined as follows:

Location
1

2

Definition
Date of state in year, month and day written as YYMMDD.

Time of state in hours, minutes and seconds written as HHMMSS. SSS.

| Location | Definition |
| :--- | :--- |
| $3-5$ | Cartesian position vector. |
| $6-8$ | Cartesian velocity vector. |
| $9-14$ | Keplerian orbital elements. <br> $15-35$ |
| Upper triangle of the state covariance <br> matrix. |  |
| 36 | Start time of fitted data, (year, month, day) |
| 37 | End time of fitted data, (year, month, day) |
| 38 | End time of fitted data, (hour, minute, second) |
| 39 | Root mean square of fit |
| 40 | Reference coordinate system of state |
| 41 | Central body indicator |
| 42 | Element set number |
| 43 |  |

## SUBROUTINE FLXATG

## Calling Sequence:

Purpose:
Common Blocks Required:

Subroutines Required:

CALL FIXATG
FIXATG controls the fixed-attitude guidance.logic.
CNTRL, CONST, INPUT, MCCOM, STATE
FOWARD, SENSO, POST

Input / Output

| I/O | $\begin{array}{l}\text { SYMBOLIC } \\ \text { NAME }\end{array}$ | DIMENSION | $\begin{array}{l}\text { COMMON } \\ \text { BLOCK }\end{array}$ | DEFINITION |
| :--- | :--- | :---: | :--- | :--- |
| I | JC | 1 | CNTRL(7) | Central body number |
| I | UT | 1 | CONST(15) | GM of the Moon (km ${ }^{3} / \mathrm{sec}^{2}$ ) |
| I | TFINAL | 1 | INPUT(4) | Trajectory stop tine (sec) |
| I | WO | 1 | INPUT(38) | Initial weight of the spacecraft (kg) |$\}$| Central value of right |
| :--- |
| ascension, (deg) |


| SYMBOLIC |  |  | COMMON DEFINITION |  |
| :---: | :---: | :---: | :---: | :---: |
| I/O | NAME | DIMENSION |  |  |
| 0 | DV | 3 | $\operatorname{MCCOM}(12)$ | Midcourse velocity impulse ( $\mathrm{km} / \mathrm{sed}$ ) |
| I | TMCS | 1 | MCCOM(18) | Midcourse time (sec after anchor epoch) |
| I | DVRET | 1 | MCCOM(25) | Retro velocity impulse (km/sec) |
| I | DJDIF | 1 | MCCOM(39) | Anchor-launch epoch difference (sec) |
| I | WTF | 1 | MCCOM(47) | Weight after midcourse burn (kg) |
| I | PSID | 10 | MCCOM $(80)$ | Desired end constraints, except $\operatorname{PSID}(7)$ is the central value of impulse magnitude ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | PSI | 10 | $\operatorname{MCCOM}(100)$ | Constraint error vector |
| 0 | IR | 1 | $\mathrm{MCCOM}(158)$ | Return key for SENSO |
| 0 | KDV | 1 | MCCOM(161) | Counter for delta-V steps taken |
| O | ICB | 1 | MCCOM(165) | Midcourse central body number |
| 1 | X | 6 | STATE(1) | State vector ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| I | T | 1 | STATE(10) | Time (sec) |
| I | ATT | 3 | STATE (11) | Unit thrust ( $\Delta \mathrm{V}$ ) vector (equator, equinox of 1950.0 ) |

## Description:

FIXATG varies the velocity impulse magnitude to scan the end conditions achievable with a midcourse burn of fixed thrust direction. The direction may also be systematically varied to ascertain the effects of attitude errors. FIXATG is called by PROTO within a loop in which midcourse execution (ignition) time is varying, but, for any particular entry, ignition time is fixed. The anchor vector state is first propagated to midcourse ' time by a call to FOWARD. The pre-midcourse state is saved in XMC. SENSO is then called to propagate the state through the burn and on to the target, then to compute end conditions (TARGET) from the arrival state. The following quantities are printed out at each step of the scan.

| DVM | Midcourse velocity impulse magnitude $(\mathrm{m} / \mathrm{sec})$ |
| :--- | :--- |
| RTA | Right ascension of the thrust $(\mathrm{deg})$ |
| DEC | Declination of the thrust $(\mathrm{deg})$ |
| RCA | Radius at closest approach $(\mathrm{km})$ |
| INC | Inclination (deg) |
| TFLT | Time of flight to closest approach (hours past launch) |
| ROPA | Radius at opposite apsis (km, opposite RCA after retro) |
| FCP | Fuel to circularize at periapsis $(\mathrm{kg})$ |
| TCF | Total correction fuel $(\mathrm{kg})$ |

Computation of the last three quantities assumes variable attitude for the retro and trim maneuvers.


## SUBROUTINE FOWARD

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:

CALL FOWARD (KSET)
This subroutine establishes certain constants to propagate the state forward in time. Calls are made to subroutines which propagate the state

CNTRL, CONST, INPUT, INTVAR, SAVE, SHAD, STATE

INTEG, MULCON, ORBIT, OUTPUT, PLANET, PRINT, TIMEC, TRMN

Input/Output

| 1/0 | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DE FINITION |
| :---: | :---: | :---: | :---: | :---: |
| O | DSAD | 3,5 | SHAD(1) | Array of back distances to the shadow cone |
| 1/O | DX | 3 | STATE(4) | Spacecraft's velocity vector |
| I | EJO | 1 | STATE(26) | Ephemeris date corresponding to state epoch |
| 0 | ELM | 6 | STATE(14) | Osculating orbital elements to be integrated |
| 0 | KDIS | 1 | CNTRL(5) | Discontinuity flag |
| 0 | KFIRST | 1 | CNTRL(12) | First pass flag |
| 0 | KHALT | 1 | CNTRL(6) | Error return flag |
| I | KMETH | 3 | INPUT(1036) | Trajectory propagation indicator table |
| I | KOUT | 1 | INPUT(1030) | Output frequency flag |
| I | KSET | 1 | Calling Operand | If non-zero, more constants are initialized |
| 0 | RSAV | 1 | SAVE(41) | Last flight path angle used in closest approach calculation. |
| I | T | 1 | STATE(10) | Seconds since state epoch |
| 0 | TCA | 1 | STATE(29) | Time of closest approach |


| SYMBOLIC |  |  | COMMON |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: |
| I/O |  |  |  |  |  |
| NAME |  |  |  |  |  | DIMENSION | BLOCK |
| :--- |$\quad$ DEFINITION

## Description:

This subroutine sets up constants to propagate the state in time, establishes the integration variables and calls the proper subroutine to propagate the state. The KHALT and KWTDOP flags are initialized to zero and METH set to KMETH (1). If the KSET flag is non-zero, the following constants are initialized:

1. TOUTL $=T$
2. DSAD and TSAD arrays to zero
3. KNTRL (1-6 and $8-10$ ) to zero
4. KDIS and KFIRST to one
5. TCA to a large number

The integration array is established according to the trajectory propagation technique. If Cowell is to be used, METH=1, the position and velocity arrays in STATE are used as the integration variables. However, if any other method is used, the ELM array is set to the integration variables. Subroutines ORBIT and TRMN are used to establish the proper set of orbital elements when any of the 'NICE" methods or averaging is used. After the integration array is established, subroutine INTEG is used to determine initial derivatives of the state. The initial state is output using subroutine OUTPUT if the output frequency flag is greater than 0. Finally, the state is propagated in time using subroutine TIMEC when numerical integration is desired at MULCON when the multi-conic algorithm is used.

Calling Sequence:
Purpose:

CALL GETTAP
This subroutine reads the ephemeris tape and sets up CETBL3 common for use in subroutine READE

Common Blocks Required: CETBL2, CETBL3, CETBL9, CNTRL
Subroutines Required: None
Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON |  |
| :--- | :---: | :---: | :---: | :--- |
| BLOCK | DEFINITION |  |  |  |
| 0 | ICW | 1 | CETBL2(1) | Flag indicating status of common <br> block CETBL3 |
| 0 | IERR1 | 1 | CETBL9(4) | Error flag |
| 0 | JDIF | 1 | CETBL9(3) | Time from beginning of ephemeric blocl <br> of data to current time |
| I | JD1 | 1 | CETBL9(1) | Reference ephemeris julian date |
| I | KREAD | 1 | CNTRL(8) | Ephemeris tape read flag |
| $O$ | NUT | 204 | CETBL3(830) | Nutation output |
| $O$ | TAB3 | 829 | CETBL3(1) | Planetary and Lunar ephemeris raw <br> data from tape |
| I | TDAY | 1 | CETBL.9 (2) | Time from reference ephemeris <br> Julian date |

## Description:

This subroutine reads the ephemeris tape and sets up the TAB3 and NUT arrays for use in subroutine. READE. If the KREAD flag is zero, these arrays are already established and no read is performed. Most of the logic in this subroutine is involved in searching through the ephemeris tape to find the desired record. The time of the desired ephemeris data, JDP, is determined from the sum of TDAY and JD1. The next record of the ephemeris tape is read to establish its current time, TAB3(1)。 The difference between the current time and desired time is determined. If the difference is negative the tape
must be backspaced, while the tape is advanced when the difference is positive. The number of records to advance or backspace the tape is determined by dividing the difference by eight,since eight days of data are stored in each record. When the proper record is determined,JDIF and ICW are set and the subroutine terminates. Another version of GETTAP is available for use at Goddard Space Flight Center. This version uses the ephemeris data stored in disk form and the direct read feature is used to retrieve the data.


## SUBROUTINE GRAV

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required:

CALL GRAV
GRAV calculates the disturbing accelerations due to external bodies.

CNTRL, CONST, GRAVTY, INPUT, INTVAR, PLNET, PERT

DVMAG

Input/Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | PROGRAM DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | DST | 12 | PLNET (73) | Distance from central planet to other planets |
| I | GM | 12 | CONST(5) | Planet gravitational constants |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | JMN | 1 | INPUT(1017) | Ephemeris flag |
| I | KP | 12 | INPUT(1001) | Planets in system |
| I | METH | 1 | INPUT(1013) | Method of integration |
| I | POS | 3 | GRAVTY(1) | Spacecraft position |
| I/O | RATES | 6 | INTVAR(8) | Derivatives of integration variables |
| I/O | RCART | 3 | PERT(1) | Accelerations of S/C |
| I | XP | $\mathrm{XP}(6,12)$ | PLNET(1) | Positions and velocities of planets in system |

Theory:
The acceleration of a body in space due to the presence of another body in space is given by

$$
\vec{A}=\frac{\mathrm{GM} \overrightarrow{\mathrm{R}}}{|\overrightarrow{\mathrm{R}}|^{3}}
$$

where GM is the gravitational constant and $R$ is the vector from the first body to the second body.

The net acceleration of a $\mathrm{S} / \mathrm{C}$ with respect to a central planet is given by the difference between the spacecraft's acceleration and the planets acceleration.

$$
\vec{A}_{s / c}=G M_{j}\left[\frac{\left(\vec{R}_{j}-\vec{R}_{s} / c\right)^{\prime}}{\left|\vec{R}_{j}-\vec{R}_{s / c}\right|^{3}}-\frac{\vec{R}_{j}}{\left|\vec{R}_{j}\right|^{3}}\right]
$$

where j represents the $j$ th planet, $\mathrm{S} / \mathrm{C}$ represents spacecraft, and all vectors are with respect to the central planet.

## Description:

GRAV calculates the disturbing acceleration on the spacecraft due to all external planets in the system. The planets in the system are determined by the vector KP, i.e., if $K P(i)$ is not equal to zero, then the net acceleration due to Planet $i$ will be calculated. The accelerations are added and stored in the vector RCART.

If the moon is the central planet, and osculating elements are used for the moon $(J M N=4)$, then the net acceleration on the spacecraft due to the Sun is the acceleration on the S/C due to the Sun minus the acceleration on the Earth due to the Sun.

If Cowell's method of integration is to be used, then the acceleration due to the central planet is added to the other acceleration and the total is loaded into RATES (4-6) for use in the integration step.


## SUBROUTINE HSDTHR

This subroutine determines the thrust and weight characteristics of the midcourse motor used on the RAE-B spacecraft. This subroutine was supplied by Hamilton-Standard Corp., who is the builder of the motor. Any questions concerning this subroutine should be directed to Mr. Charles Newman of NASA Goddard Space Flight Center.

## SUBROUTINE INPUTF

Calling Sequence:
Purpose:

CALL INPUTF

This subroutine reads the input data cards and stores the information in the proper common blocks.

Common Block Required: FIELDM, INPUT, INPUTS
Subroutines Required:

Input / Output

| 1/0 | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSION | $\begin{array}{r} \text { COMMON } \\ \text { BLOCK } \end{array}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | A | 1000 | INPUT(1) | Real part of the input array |
| 0 | AS | 1000 | INPUTS(1) | Real part of the saved input array |
| 0 | C | 16,16 | FIELDM(1) | Cosine coefficient of spherical harmonic potential term. |
| O | KMOD | 16,16 | FIELDM(525) | Array of flags used to indicate if a harmonic is input |
| O | KOPT | 100 | INPUT(1001) | Integer portion of the input array |
| 0 | KOPTS | 100 | INPUTS(1001) | Integer portion of the saved input array |
| 0 | MMOD | 1 | FIELDM(514) | Number of tesserals used |
| I | MODLEM | 1 | INPUT(1035) | Flag used to determine the type of gravitational field. |
| 0 | NMMOD | 1 | FIELD(515) | Highest zonal for which a tesseral is desired |
| 0 | NMON | 1 | FIELDM(513) | Number of zonals used |
| O | S | 16,16 | FTELD(257) | Sine coefficient of the spherical harmonic potential term |

## Description:

This subroutine reads the input data cards and establishes the working input arrays. These arrays consist of the A and the KOPT arrays of INPUT common. These arrays are set equal to the saved input arrays of INPUTS common before the case is initiated. The saved input arrays consist of the accumulation of all previous input including inputs from previous cases. Thus, only the inputs which differ from case to case need be input. The cases are separated by a blank card.

The saved input array is initialized to preset values in the BLOCKDATA subroutine. However, some of the values of preset inputs are dependent on the program MODE flag input via location 1044. Hence, logic is incorporated in this subroutine to preset those inputs dependent on the program mode. The inputs preset and their respective values are presented in Table I.

A special set of inputs are also included to set up the gravitational field. These inputs are designated by input locations greater than 2000. The data cards with input locations in the $2000^{\prime}$ 's must be placed after the normal 1000 series inputs.

The inputs for the gravity field can either define the entire field, modify one of the preset fields, or use the field set up in the previous case. The MODLEM flag is used to determine the field as follows:

$$
\left.\begin{array}{rlrl}
\text { MODLEM } & =1,2,3 & & \left.\begin{array}{l}
\text { Modify or use the L1, Earth } J_{2}, \text { or } \\
\\
\\
\end{array}\right)=5 P L 15 \times 8 \text { field. }
\end{array}\right] \begin{aligned}
& \text { New field input. Field is comprised of only } \\
& \text { the } 2000 \text { series inputs of the current case. }
\end{aligned}
$$

Subroutine OBSET is used to initialize the field when MODLEM $=1,2$ or 3 .



## Calling sequence: CALL INTEG (LOPT)

Purpose: | This subroutine sets up arrays for use in |
| :--- |
| subroutine EQNS where the derivatives of |
| the integration variables are calculated. |
| This subroutine also calls EQNS and / or |
| the numerical integration routines. |

Common blocks required:

Subroutines required:

CONST, CNTRL, INPUT
INTVAR, PERT, STATE

EQNS, ORBIT,
RKSEVN, TWELVE

| 1/0 | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PROGRAM DIMENSIONS | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DESCRLPTION |
| :---: | :---: | :---: | :---: | :---: |
| I | DELTO | 1 | INPUT (2) | Initial compute interval in automatic compute interval mode. |
| I | DOM | 3 | PERT (25) | Position and velocity vectors on reference orbit when using Encke. |
| 1/0 | DX | 3 | STATE (4) | Spacecraft velocity |
| 0 | D2X | 3 | STATE (7) | Spacecraft accelerations |
| I/O | ELM | 6 | STATE (14) | Spacecraft osculating orbital elements on Encke variables |
| I | GM | 12 | CONST (5) | Gravitational constants |
| 1/0 | H | 1 | INTVAR (14) | Compute interval |
| I | JC | 1 | CNTRL (7) | Central planet |
| 1/0 | KDIS | 1 | CNTRL (5) | Discontinuity flag |
| I | KHALT | 1 | CNTRL (6) | Error return flag |
| I | KINT | 1 | INPUT (1014) | Numerical integration scheme indicator |
| I | KORECT | 1 | INPUT (1073) | Flag not to calculate derivatives at the end of the step |
| I | LOPT | 1 | Calling | Positive to integrate a step Negative to calculate derivatives only |
| I | METH | 1 | INPUT (1013) | Trajectory propagation method indicator |
| I | RATES | 6 | INTVAR (8) | Derivatives of state |
| 1/O | T | 1 | STATE (10) | Time corresponding to state in STATE common |
| 0 | TIME | 1 | INTVAR (1) | Time corresponding to variables to be integrated in INTVAR common |
| I | TRU | 1 | STATE (27) | True anomaly of state when using METH $=4$ or 5 , or true anomaly plus argument of perigee when METH $=8$ |
| I/O | X | 3 | STATE (1) | Spacecraft position |
| 1/0 | Y | 6 | INTVAR (2) | Variables to be integrated |

## Description:

The primary function of this subroutine is to set up the integration array before integration and restore the new state after integration is completed. If LOPT is negative, numerical integration is not executed and this routine is only used to determine the derivatives at a specified time. The initial state is brought into the routine via X and DX or ELM in STATE common. When Cowell is used (METH=1), the position and velocity vectors are in $X$ and $D X$. When the other trajectory propagation methods are used, the various sets of orbital elements or Encke's variables are in ELM. Whichever method is used, the state is loaded into the integration array,' Y.

If LOPT is zero or positive, the appropriate numerical integration subroutine is called to propagate the state over the time step $H$.

After the state is propagated, the state array is restored to the integration array in order to set the state to the values after integration. The position and velocity vectors are also calculated at the end of integrations for use in other subroutines.

SUBROUTINE INTEG



## SUBROUTINE INTERP

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL INTERP (T, Z)
To determine the position and velocity of the spacecraft at times other than compute times.

CNTRL, CONST, INPUT, INTER, PERT
ORBIT, TRMN
Input / Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PROGRAM DIMENSION | $\begin{gathered} \text { COMMON } \\ \quad \text { BLOCK } \\ \hline \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ACL | 6, 10 | INTER(81) | Back values of the derivative of the interpolation quantities |
| I | GM | 12 | CONST(5) | Gravitational constant |
| I | JC | 1 | CNTRL(7) | Central planet |
| I | INT. | 1 | INTER(131) | Number of back values stored |
| I | METH | 1 | INPUT(1013) | Trajectory propagation indicator |
| I | PI 2 | 1 | CONST (3). | Twice pi |
| I | POS | 6, 12 | INTER(11) | Back values of the interpolation quantities |
| I | T | 1 | CALLING OPERAND | Independent variable used in interpolation |
| I | X | 10 | INTER(1) | Back values of the independent variable. |
| O | Z | 6 | CALLING <br> OPERAND | Interpolated position and velocity vectors |

Theory:
The interpolation formula used is of the form

$$
P(X)=\sum_{j=0}^{n} h_{0, j}(X) f_{i}+h_{1, j}(X) f_{i}^{\prime}+h_{2, j}(X) f_{i}^{\prime \prime}
$$

where
$\mathrm{P}(\mathrm{X}) \quad$ is the interpolated value
X is the independent variable
$h_{0, j}, h_{1, j}, h_{2, j}$ are functions in $X$
$f, f^{\prime}, f^{\prime \prime} \quad$ are the back values of the quantity to be interpolated and its first and second derivatives.

For the cases where only the first derivatives are available, the interpolating polynomial shown in equation (1) reduces to a Hermitian polynomial and the functions $h_{i, j}$ become,

$$
\begin{align*}
& h_{0, j}(X)=\left[1-2\left(X-X_{j}\right) \lambda_{j}^{\prime}\left(X_{j}\right)\right] \lambda_{j}^{2}(X)  \tag{2}\\
& h_{i, j}(X)=\left(X-X_{j}\right) \lambda_{j}^{2}(X)
\end{align*}
$$

where $\lambda_{j}$ is the Lagrange interpolating coefficient defined by

$$
\lambda_{j}(X)=\underset{\substack{i=0 \\ i \neq j}}{n} \quad\left[\left(X-X_{i}\right) /\left(X_{j}-X_{i}\right)\right]
$$

also

$$
\lambda_{j}^{\prime}\left(X_{j}\right)=\sum_{\substack{i=0 \\ i \neq j}}^{n} \frac{1}{X_{i}-X_{i}}
$$

When both the first and second derivatives of the interpolation quantities are available, the functions in the interpolating polynomial become

$$
h_{k, j}(X)=\phi_{k, j}(X) \quad \lambda_{j}^{3}(X) \quad K=0,2
$$

The functions $\varnothing_{\mathrm{k}, \mathrm{j}}$ are defined by

$$
\begin{align*}
& \emptyset_{0, \mathrm{j}}(\mathrm{X})=1+6\left(\mathrm{X}-\mathrm{X}_{\mathrm{j}}\right)^{2}\left[\left(\left(\lambda_{\mathrm{j}}^{1}\left(\mathrm{X}_{\mathrm{j}}\right)^{2}-\frac{1}{4} \lambda_{\mathrm{j}}^{\prime \prime}\left(\mathrm{X}_{\mathrm{j}}\right)\right]-3\left(\mathrm{X}-\mathrm{X}_{\mathrm{j}}\right) \lambda_{\mathrm{j}}^{\prime}\left(\mathrm{X}_{\mathrm{j}}\right)\right.\right. \\
& \emptyset_{1, \mathrm{j}}(\mathrm{X})=\left(\mathrm{X}-\mathrm{X}_{\mathrm{j}}\right)-3\left(\mathrm{X}-\mathrm{X}_{\mathrm{j}}\right) \lambda_{\mathrm{j}}^{\prime}\left(\mathrm{X}_{\mathrm{j}}\right)  \tag{4}\\
& \emptyset_{2, \mathrm{j}}(\mathrm{X})=\frac{1}{2}\left(\mathrm{X}-\mathrm{X}_{\mathrm{j}}\right)^{2}
\end{align*}
$$

Where the second derivative of the Lagrange interpolating coefficient is given by

$$
\lambda_{j}^{\prime \prime}\left(X_{j}\right)=\sum_{\substack{i=1 \\ i \neq j}}^{n}\left(\frac{1}{X_{j}-x_{i}}\right)^{2}-\sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{1}{\left(X_{j}-x_{i}\right)^{2}}
$$

The first derivative of equation (1) defines an interpolating polynomial for the first derivative of the interpolation quantity. Differentiating equation (1) yields

$$
\begin{equation*}
P^{\prime}(X)=\sum_{j=0}^{k} h_{0, j}^{\prime}(X) f_{j}+h_{1, j}(X) f_{j}^{\prime}+h_{2, j}^{\prime}(X) f_{j}^{\prime \prime} \tag{5}
\end{equation*}
$$

where

$$
h_{i, j}^{\prime}(X)=\gamma_{i, j}(X) \lambda_{j}^{3}\left(X_{j}\right) \quad i=0,2
$$

and

$$
\gamma_{i, j}(X)=3 \emptyset_{i, j}(X) \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{1}{X-X_{i}}+\phi_{i, j}(X)
$$

The functions $\emptyset^{\prime}$ are given by

$$
\begin{align*}
& \emptyset_{0, j}^{\prime}(X)=12\left(X-X_{j}\right)\left[\left(\lambda_{j}^{\prime}\left(X_{j}\right)\right)^{2}-\frac{1}{4} \lambda_{j}^{\prime \prime}\left(X_{j}\right)\right]-3 \lambda_{j}^{\prime}\left(X_{j}\right) \\
& \emptyset_{1, j}^{\prime}(X)=1-6\left(X-X_{j}\right) \lambda_{j}^{\prime}\left(X_{j}\right)  \tag{6}\\
& \emptyset_{2, j}\left(X_{j}\right)=\left(X-X_{j}\right)
\end{align*}
$$

## Description:

This subroutine determines the position and velocity vectors at any arbitrary time by interpolating on saved values of the state and its derivatives. The saved values of the state and its derivatives are brought into the subroutine via INTER common. This array is set up by subroutine UPDATE. The values in this array are the integration parameters. Thus if METH $=1$, this array contains the position and velocity vectors, whereas it contains the orbital elements if METH=3. If METH $=1$ or 2 , the state and its first and second derivatives are contained in INTER. Only the state and its first derivative are available when METH is other than 1 or 2.

When METH equals 1 or 2 , the position (or position from reference orbit) is obtained using equations (1), (3) and (4). The velocity (or velocity from reference orbit) is obtained using equations (5) and (6). When METH is other than 1 or 2 , the state is obtained from equations (1) and (2).

The position and velocity vectors are determined from the interpolated state. These quantities are set in the $Z$ array and returned through the argument list.


Determine $\varnothing_{i, j}$ and $\emptyset^{\prime}$ using equations (4) and (6)

Interpolate for position and velocity or Encke variables using equations (1) and (5)

## SUBROUTINE JET

## Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Called:

## CALL JET

JET computes a first-guess translunar midcourse velocity impulse using patched-conic assumptions.

CONST, INPUT, INTVAR, MCCOM, PLNET, STATE BCONIC, CROSS, DOT, DVMAG, MVTRN, M50LEQ, ORIENT, PLANET, RETDV, ROTAIT, TRMN, VNORM

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :--- | :--- | :--- | :--- |
| I | RTD | 1 | CONST(1) | Radians to degrees conversion factor |
| I | GME | 1 | CONST(7) | GM(km ${ }^{3} / \mathrm{sec}^{2}$ ) for the Earth |
| I | GMM | 1 | CONST(15) | GM(km ${ }^{3} / \mathrm{sec}^{2}$ ) for the Moon |
| I | WTO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | DJO | 1 | INPUT(46) | Julian date (days) of anchor <br> vector epoch |
| I | JT | 1 | INPUT(1031) | Target body number (11) |
| I | MCOUT | 1 | INPUT(1050) | Extra output key (prints if .GE. 3) |
| O | TINT | 1 | INTVAR(1) | Time for anchor epoch (sec) |
| O | DVMC | 3 | MCCOM(12) | Midcourse velocity impulse (km/sec) |
| I | BVD | 2 | MCCOM(19) | Desired miss-vector (km) <br> if IBTR =1 |
| I | DVB4 | 1 | 1 | MCCOM(24) | | Velocity expended previously (km/sec) |
| :--- |
| I |
| PRD |
| I |
| OINC |


| I/O | $\begin{aligned} & \text { SYMBOLI } \\ & \text { NAME } \end{aligned}$ | DIMENSION | COMIMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I | VINFD | 1 | MCCOM(83) | Desired hyperbolic excess velocity $/ \mathrm{km} / \mathrm{sec}$ |
| I | DCEV | 1 | $\mathrm{MCCOM}(84)$ | Desired circular excess velocity $(\mathrm{km} / \mathrm{sec})$ after retro |
| I/O | NGROPT | 1 | MCCOM $(163)$ | Gradient re-computation key |
| I | KGLAW | 1 | MCCOM(164) | Guidance law key |
| I | IBTR | 1 | MCCOM(167) | End constraint type key <br> ( 1 for BVD, 2 for PRD and OINC) |
| O | XP | 6,12 | PLNET 1 ( | Ephemeris state of body I (km, $\mathrm{km} / \mathrm{sec}$ ) |
| 1 | X 1 | 6 | STATE (1) | Pre-maneuver state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| I | TIME | 1 | STATE(10) | Time(sec) of X1 referred to anchor epoch |
| I | EJO | 1 | STATE(26) | Ephemeris time (days) at anchor epoch |
| O | EJT | 1 | STATE(28) | Ephemeris time (days) |

## Introduction:

The method to be described has been fruitfully applied to the determination of midcourse correction maneuvers for the RAE-B mission. It has been used as a pre-targeting device to provide "first-guess" midcourse corrections to a precise differential-correction-type targeting scheme. In this role, the method has eliminated the need for non-linear targeting measures such as control limiting and gradient re-computations and greatly reduced the number of trajectory calculations required.

A gross description of the method would be "patched-conic targeting with constrained end conditions." Refinements which contribute to the success of the method are primarily found in the use of the Jacobian energy for targeting and in the use of radius of closest approach and inclination as terminal condition parameters.

## Transfer Phase



Schematic for a Transfer Trajectory

The transfer phase is treated as a single conic section relative to the central body. The first terminal of this phase is the point of the midcourse correction on the uncorrected transfer trajectory at radius $R_{1}$, velocity $V_{1}$, at time $t_{1}$. The second terminal is on the sphere of influence of the target at a radius $R_{2}$. The second terminal's radius vector is computed from

$$
\begin{equation*}
R_{2}=R_{t}+P \tag{1}
\end{equation*}
$$

where $R_{t}$ is the position of the target at $t_{2}$ relative to the central body and where $P$ is a vector from the central body to the point of entry of the target's sphere of influence. The time, $t_{2}$, of arrival at the sphere of influence is computed from

$$
\begin{equation*}
t_{2}=t_{f}-\tau \tag{2}
\end{equation*}
$$

where $t_{f}$ is the time of arrival at closest approach (specified a priori) and where $\tau$ is the time required to trsvel on the approach hyperbola from the sphere of influence to closest approach.

The arrival phase parameters, P and $\tau$, will be described later. An iterative solution of Lambert's Problem (i.e., "Find the conic section passing from $R_{1}$ to $R_{2}$ in time $t_{2}-t_{1}^{\prime \prime}$ ' is employed. This solution provides $V_{1}{ }^{*}$, the velocity on the transfer conic at the first terminal, and $V_{2}^{*}$, the transfer conic's velocity at the second terminal. The midcourse correction impulse, $\Delta V$, is computed from

$$
\begin{equation*}
\Delta \mathrm{V}=\mathrm{V}_{1}^{*}-\mathrm{V}_{1} \tag{3}
\end{equation*}
$$

and the target-relative velocity, V , at $\mathrm{t}_{2}$ is

$$
\begin{equation*}
\mathrm{v}=\mathrm{v}_{2}^{*}-\mathrm{v}_{\mathrm{t}} \tag{4}
\end{equation*}
$$

where $V_{t}$ is the target's velocity at $t_{2}$ relative to the central body.
The first transfer conic is computed with P in the target's orbital plane at $20^{\circ}$ earthward from tangential. The target relative transfer time, $\tau$, is fixed at 66,000 seconds.

The target-relative velocity derived from this first conic is used to initiate the arrival phase calculations. A two-or-three-step iteration (transferring $V$ to arrival phase calculations and $P$ and $\tau$ to transfer phase calculations) will

converge to a steady-state value for $\Delta V$. This answer is the impulsive patched-conic, fixed-time-of-arrival midcourse correction.

## Arrival Phase

The arrival phase computations use the target-relative approach velocity and desired arrival conditions to develop (1) the time of passage from the sphere of influence to closest approach and (2) the point of entry into the sphere of influence. The desired arrival conditions are specific values of radius of closest approach and inclination. We develop characteristics which the approach hyperbola must possess in order to satisfy the desired arrival conditions. We assume that the target-relative approach velocity vector defines the direction of the arrival asymptote of the approach hyperbola. Furthermore, we assume that the point of entry into the sphere of influence can be computed for the next transfer phase using the direction of the arrival asymptote and the target-relative energy from the current arrival phase in its calculation. The energy of the approach orbit as used in arrival phase computation is defined by equation (5),

$$
\begin{equation*}
\mathrm{C}_{3}=\mathrm{V} \cdot \mathrm{~V}-\frac{2 \mu_{t}}{\mathrm{a}^{\prime}} \tag{5}
\end{equation*}
$$

where V is the approach velocity. This (Jacobian) energy is adjusted for perturbation by the target body on the transfer trajectory.

Figure (1) shows an arrival hyperbola whose no-plane character is described by a closest approach radius, $r_{p}$, and half-angle, $\alpha$. Given $r_{p}$, we can use the energy, $C_{3}$,
to solve


Figure 1


Figure 2
for the various parameters of the hyperbola
$v_{h}=\sqrt{C_{3}}$
(hyperbolic excess speed)
$v_{p}=\sqrt{C_{3}+\frac{2 \mu}{r_{p}}} \quad$ (closest approach speed)
$\alpha=\frac{-1 v_{p} v_{h}}{\left(\mu / r_{p}\right)}$
(half-angle between asymptotes)
$a=-\mu / C_{3}$
$b=\left(x_{p}-a\right) \sin \alpha$
(asymptotic miss distance)
$e=\sqrt{1+\left(\frac{b}{a}\right)^{2}}$
(eccentricity)

$$
\begin{array}{ll}
\mathrm{p}=-\mathrm{a}\left(\frac{\mathrm{~b}}{\mathrm{a}}\right)^{2} & \text { (semi-latus rectum) } \\
\frac{1}{\mathrm{n}}=|\mathrm{a}| \sqrt{\frac{|\mathrm{a}|}{\mu}} & \text { (reciprocal mean motion) } \tag{13}
\end{array}
$$

The true anomaly, f , at "patch" distance, r , is computed from

$$
\begin{equation*}
\mathbf{f}=\cos ^{-1}\left\{\left(\frac{p}{r}-1\right) / e\right\} \tag{14}
\end{equation*}
$$

and the time of passage, $\tau$, from the sphere of influence to the point of closest approach is

$$
\begin{equation*}
\tau=\frac{1}{n}\left\{e \sinh f^{\prime}-f^{\prime}\right\} \tag{15}
\end{equation*}
$$

where $\quad f^{\prime}=\ln \left\{1+\frac{\sqrt{e-1}}{e+1} \quad \tan \frac{f}{2}\right\}-\ln \left\{1-\frac{e-1}{e+1} \quad \tan \frac{f}{2}\right\} \quad$.
We next calculate the point of entry into the sphere of influence in such a way that the hyperbola has the specified inclination, $i_{d}$. We first define the arrival asympoin. S , as V normalized to unity. Then, if K is a unit vector normal to the target's equatorial plane at time of closest approach, we can define vectors $T$ and $R$ normal to S as follows. (See Figure 2)

$$
\begin{align*}
& T=\frac{S \times K}{|S \times K|}  \tag{17}\\
& R=T \times S \tag{18}
\end{align*}
$$

The miss-vector, $B$, lies in the orbital plane and in the plane of $T$ and $R$ at an angle $\theta$ measured from T towards R .

$$
\mathrm{B}=\mathrm{T} \cos \theta+\mathrm{R} \sin \theta
$$

The plane of the hyperbola is defined by a unit vector, $H$, in the direction of the angular momentum.

$$
\begin{equation*}
H=B \times S=R \cos \theta-T \sin \theta \tag{20}
\end{equation*}
$$

The condition that the orbit's inclination is $i_{d}$ is

$$
\begin{equation*}
H \cdot K=\cos i_{d}=R \cdot K \cos \theta \tag{21}
\end{equation*}
$$

which can be solved for $\cos \theta$ if $|R \cdot K| \leq\left|\cos i_{d}\right|$. If $|R \cdot K|>\left|\cos i_{d}\right|$, it means $i_{d}$ cannot be attained. In that case, the best that can be done is

$$
\begin{equation*}
\cos \theta=1 \cdot \operatorname{sign}\left(\frac{\cos i_{d}}{R \cdot K}\right) \tag{22}
\end{equation*}
$$

The $\operatorname{sign}$ on $\sin \theta= \pm \sqrt{1-\cos ^{2} \theta}$ can be chosen to make the miss-vector lie above or below the equator in the miss plane. Having now calculated $B$, we can form the vector, $P$, from the target's center toward the point of entry into the sphere of influence.

$$
\begin{equation*}
P=r\{S \cos (f+\alpha)+B \sin (f+\alpha)\} \tag{23}
\end{equation*}
$$

It has been found beneficial for convergence of the arrival-transfer iteration to introduce a "gain" of .7 on the change of P between iteration steps.

The computations above provide the point of entry of the sphere of influence and the time of passage from the sphere of influence to closest approach. The point of entry is then used to establish a new second terminal position for the transfer trajectory.

## Guidance Laws

The solution provided by the above process is the fixed-time-of-arrival (FTA) guidance solution. The FTA guidance law constrains arrival (at closest approach) time, $t_{f}$, to be a specific value while satisfying the desired end conditions of radius of closest approach and inclination. Other guidance laws of interest relative to the RAE-B mission are:

1) minimum midcourse fuel,
2) fixed target energy, and
3) variable target energy.

Each of the other guidance laws constrain radius of closest approach and inclination just as the FTA law does, but do not specifically constrain arrival time. The minimum midcourse fuel law embraces the "critical plane" solution, the fixed target energy law constrains hyperbolic excess speed at the target, and the variable target energy law constrains post-retro velocity at the target subject to a prescribed de-boost strategy. The solution for each of these laws, however, corresponds to a particular arrival time (or, flight time), so flight time is used as the independent variable in seeking solutions for each law.

The MFG and MTF laws are pre-targeted by means of a Newton-Raphson-type iteration with flight time as the independent variable. The iteration seeks to null the dot product of the difference of two successive midcourse correction impulses with the impulse itself. That is, it seeks to find the flight time for which the magnitude of the correction velocity doesn't change (i.e., is minimum).

Equation 24 defines the condition for minimizing the magnitude of $\Delta \mathrm{V}$.

$$
\begin{equation*}
\Delta V \cdot \frac{d(\Delta V)}{d t}=0 \tag{24}
\end{equation*}
$$

where $t$ signifies flight time in this case. We define $t_{n}$ to be the flight time for the n-th trial and $\Delta V_{n}$ to be the velocity impulse for that trial. Then, approximately,

$$
\begin{equation*}
\left(\frac{d(\Delta V)}{d t}\right)_{n}=\frac{\Delta V_{n}-\Delta V_{n-1}}{t_{n}-t_{n-1}} \tag{25}
\end{equation*}
$$

and more approximately,

$$
\begin{equation*}
\Delta V_{n+1}=\Delta V_{n}+\left(t_{n+1}-t_{n}\right) \quad\left(\frac{d(\Delta V)}{d t}\right)_{n} \tag{26}
\end{equation*}
$$

We look for $t_{n+1}$ such that equation 27 holds.

$$
\begin{equation*}
\Delta V_{n+1} \cdot\left(\frac{d(\Delta V)}{d t}\right)_{n+1}=0 \tag{27}
\end{equation*}
$$

By making liberal use of equations 25 and 26 , equation 27 may be solved to render $t_{n+1}-t_{n}$.

$$
\begin{equation*}
t_{n+1}-t_{n}=-\left(t_{n}-t_{n-1}\right) \quad \frac{\Delta V_{n} \cdot\left(\Delta V_{n}-\Delta V_{n-1}\right)}{\left|\Delta V_{n}-\Delta V_{n-1}\right|^{2}} \tag{28}
\end{equation*}
$$

This process converges well in all cases tested. The resultant $\Delta V$ changes very little with precise targeting, although the corresponding flight time shifts by an hour or two.

The fixed time of arrival guidance law is pre-targeted to 1.015 times the desired flight time. This empirical factor tends to compensate for the difference between patchedconic and integrated-perturbed lunar transfer trajectory flight times. No iteration is required as for the other guidance laws, since flight time is both the third constraint of the FTA law and the independent variable for pre-targeting.

The pre-targeting process for the FTE and VTE guidance laws include a regular Newton-Raphson iteration to null the third constraint by varying flight time. For the FTE law, the third constraint error function is

$$
\begin{equation*}
\left.{ }_{3}\right)_{3}=v_{\infty}(\text { desired })-\sqrt{C_{3 T}} \tag{29}
\end{equation*}
$$

where $\mathrm{C}_{3 \mathrm{~T}}$ is defined by equation 5. For the VTE law, the function to be nulled is

$$
\begin{equation*}
\psi_{3}=\sqrt{\frac{\mu_{m}}{r_{p d}}}-\sqrt{C_{3 T}+\frac{2 \mu_{m}}{r_{p d}}}+\delta v_{r} \tag{30}
\end{equation*}
$$

where $r_{p d}$ is the desired distance of closest approach and $\delta v_{r}$ is the velocity impulse imparted by the retro motor. The VTE law usually has two solutions, i.e., two flight
times for which $\psi_{3}=0$. The iteration is constrained to find the solution with positive slope, which is the solution of the longer flight time.


## SUBROUTINE LUNA

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required:
Reference:

## CALL LUNA

To determine the position of the Moon with respect to the Earth.

CONST, CNTRL, INPUT, INTVAR, MOON, PLNET, STATE

MVTRN, M50MDT, OBLTY, ORBIT, ROTATE, TRMN

1. Supplement to the American Ephemeris and Nautical Almanac, U.S. Naval Observatory, U.S. Government Printing office.


## Theory:

This subroutine calculates the position and velocity of the Moon from the mean element, mean elements plus extra terms from the lunar theory or from input osculating elements. The mean elements are obtained from the supplement to the nautical ephemeris. The mean longitude of the Moon, the mean longitude of perigee and the longitude of the ascending node are obtained from the following polynomials in time:

$$
\begin{align*}
& \lambda=4.3853720+8399.0912 \mathrm{C}-1.97746 \mathrm{D}-5 \mathrm{C}^{2} \\
& \gamma=1.730894+71.017994 \mathrm{C}-3.6267 \mathrm{D}-5 \mathrm{C}^{2} \\
& \Omega=1.4312588+33.757099 \mathrm{C}+3.6263 \mathrm{D}-5 \mathrm{C}^{2} \tag{1}
\end{align*}
$$

where
$\lambda=$ longitude of the Moon
$\gamma=$ longitude of perigee
$\Omega=$ longitude of the ascending node
$\mathrm{C}=$ number of julian centuries since 1965.
These quantities are in the mean equinox and ecliptic of date. The argument of perigee, $\omega_{,}$, and the mean anomaly, AM, of the lunar orbit is obtained from

$$
\begin{align*}
& \mathrm{AM}=\lambda-\gamma \\
& \omega=\gamma-\Omega \tag{2}
\end{align*}
$$

Next, the eccentric anomaly of the lunar orbit is determined from Kepler's equation for small eccentricities, or
$E=A M+e \sin A M+\frac{e^{2}}{2}+\frac{e^{3}}{8}(3 \sin 3 A M-\sin A M)$
where e is the eccentricity of the lunar orbit (.054900489)
The position and velocity of the Moon in its orbit plane can now be obtained from
$\mathrm{X}=\mathrm{a}(\cos \mathrm{E}-\mathrm{e})$
$Y=a \sqrt{1-e^{2}} \sin E$
$\dot{X}=-V \sin E / S$
$\dot{Y}=V \sqrt{1-} e^{2} \cos E / S$
where
$S=\sqrt{\left(1-e^{2}\right) \cos ^{2} E+\sin ^{2} E}$
$V=\sqrt{\text { GMN }\left(\sqrt[2]{x^{2}+y^{2}}-\frac{1}{a}\right)}$ (the velocity of the Moon)
$a$ is the semi-major axis of the lunar orbit (384750.8998)

The X and Y components are in the lunar orbit plane with the X axis pointing to perigee and the Y axis in the direction of motion.

The cartesian coordinates of the Moon are next transformed to the mean equinox and ecliptic of date using the following rotation matrix:

$$
\left\{\begin{array}{l}
\mathrm{x}  \tag{5}\\
\mathrm{y} \\
\mathrm{z}
\end{array}\right\}=\left[\begin{array}{ll}
\cos \omega \cos \Omega & -\sin \omega \cos \Omega \\
-\cos i \sin \Omega \sin \omega & -\cos i \sin \Omega \cos \omega \\
\cos \omega \sin \Omega & -\sin \omega \sin \Omega \\
+\cos i \cos \Omega \sin \omega & +\cos i \cos \Omega \cos \omega \\
\sin i \sin \omega & \sin i \cos \omega
\end{array}\right]\left\{\begin{array}{l}
x \\
y
\end{array}\right\}
$$

where $i$ is the inclination of the lunar orbit $w$ 。r.t. the ecliptic (. 08980414 radians)

The Moon's position and velocity vectors in the Earth's mean equinox and equator of 1950 are obtained from standard rotations.

Extra terms are used in the calculation of the inclination, longitude of the ascending node of the lunar orbit, mean anomaly and semi-major axis of the lunar orbit when the proper input flag is set. The terms include the effects of evection, variation, and other periodic terms due to the Sun's force on the Moon.

The longitude of the Sun and the longitude of the perigee of the Sun's orbit is determined from mean elements similar to the lunar terms. The polynomials used are:

$$
\begin{align*}
& \lambda \mathrm{s}=4.8860536+6.28331958 \mathrm{C}+5.2796 \mathrm{D}-6 \mathrm{C}^{2}  \tag{6}\\
& \gamma \mathrm{~s}=4.9082294+.030005364 \mathrm{C}+7.90246 \mathrm{D}-6 \mathrm{C}^{2}
\end{align*}
$$

The changes in the above mentioned quantities are:

$$
\begin{array}{ll}
i & =i \text { mean }+.002515665 \cos (2 \lambda s-2 \Omega \text { mean }) \\
\Omega & =\Omega \text { mean }+.02805487 \sin (2 \lambda s-2 \Omega \text { mean })  \tag{7}\\
a & =a \operatorname{mean}(1+.0090714046 \cos (2 \lambda-2 \lambda s)
\end{array}
$$

and the true anomaly of the Moon is determined from

$$
\begin{align*}
u= & A M+.10975961 \sin A M+.0037634149 \sin 2 A M \\
& +.00017926303 \sin 3 A M-.00054493 \sin \lambda-\lambda s) \\
& +.022238412 \sin (2 \lambda-2 \lambda s-A M)+.011493967 \sin (2 \lambda-2 \lambda s)  \tag{8}\\
& -.00324292 \sin \left(\lambda s-\gamma^{s}\right)
\end{align*}
$$

The osculating lunar position and velocity vectors are determined in a completely different manner. The mean anomaly at the desired time is determined from the mean motion and mean anomaly at epoch from

$$
\mathrm{AM}=P \mathrm{M}(\mathrm{DELT})+F O M
$$

where PM is the mean motion
DELT is the time since epoch
FOM is the mean anomaly at epoch.
The true anomaly is determined from subroutine TRMN, and the position and velocity vectors obtained from the osculating elements from subroutine ORBIT.

## Description:

The ephemeris epoch and time since the epoch that the ephemeris is to be determined are brought into the routine via common blocks. The JMN flag is used to determine the type of ephemeris as follows

$$
\begin{aligned}
\mathrm{JM} & =1 \text { mean elements } \\
& =2 \text { mean elements plus extra terms } \\
& =3 \text { osculating elements. }
\end{aligned}
$$

If JMN equals one or two, the mean elements of the lunar orbit are determined from the polynomials in equation (1). If the extra terms are to be included, they are determined from equation (6) and the eccentric anomaly from equation (7). Otherwise, the extra terms are not included and equations (2) and (3) are used. Finally, equation (4) is used to determine the coordinates and they are rotated to the mean equinox and equator of date using the rotation matrix shown in equation (5). Subroutines M50MDT and OBLTY are used to obtain the vectors in Earth mean equinox and equator of date. The osculating position and velocity vectors are obtained from TRMN and equation (9). The osculating elements are brought into the subroutine via Moon common and describe the Moon's orbit with respect to the Earth in the Earth's mean equator and equinox of 1950 .

The position and velocity vectors determined are loaded into the XP array according to the value of the current central planet number in JC.

$$
\begin{aligned}
& \mathrm{JC}=3 \quad \text { Earth central planet, thus position and velocity } \\
& \text { of Moon w. r.t. Earth loaded into XP array in } \\
& \text { Moon?s position }
\end{aligned}
$$

| JC | $=11 \quad$Moon central planet, thus position and velocity <br> of Earth w.r.t. Moon loaded into XP array in <br> Earth's position. The negative of the values <br> determined for the Moon w.r.t. Earth are loaded |
| :--- | :--- |
| JC $\quad=\quad 3$ or $11 \quad$into XP(I, 3). <br> Some other planet central, thus the position of <br> the Earth W.r.t. central planet is added to Moon's <br> position and velocity and loaded into the Moon's <br> position in the XP array. |  |




## SUBROUTINE MATMPY

| Calling Sequence: | CALL MATMPY (KRL, A, KCLRR, B, KCR, C) |
| :--- | :--- |
| Purpose: | MATMPY multiplies the matrices A (KRL $\times$ KCLRR $)$ <br> by $B(K C L R R \times K C R) ~ t o ~ g e t ~ C(K R L ~$ KCR) |
| Common Blocks: | None |
| Subroutines | None |

## Input/Output

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | A | Variable | Calling Operand | Matrix on the left |
| I | B | Variable | Calling Operand | Matrix on the right |
| O | C | Variable | Calling Operand | [A] x [B] |
| I | KCLRR | 1 | Calling Operand | Number of columns <br> in A and rows in B |
| I | KCR | 1 | Calling Operand | Number of columns <br> in C and B |
| I | KRL | 1 | Calling Operand <br> A and C of rows in |  |

## Theory:

General matrix multiplication is defined by $[\mathrm{C}]=[\mathrm{A}] \times[\mathrm{B}]$
where

$$
C_{i j}=\sum_{k=1}^{m} \quad A_{i k} B_{k j}
$$

$m$ is the number of columns in $A$ and rows in $B$.

## Description:

MATMPY first initializes the matrix $C$ to zero and then forms the sum of the products above for $\mathbf{i}=1, K R L$ and $\mathfrak{j}=1, K C R$. The arrays are singly-dimensioned to avoid variable dimensioning.

Calling Sequence:
Purpose:
CALL MCBURN
MCBURN accepts the pre-midcourse state and time plus a velocity impulse, computes the post-midcourse state and time.

CNTRL, C $\varnothing$ NST, INPUT, MCC $\varnothing \mathrm{M}, ~ S T A T E$
DVMAG, F $\varnothing$ WARD, D $\varnothing$ T, BURND

## Input/Output

| I/ $\varnothing$ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PROGRAM <br> DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| $\emptyset$ | JC | 1 | CNTRL(7) | Post-maneuver central body number |
| I | GM | 12 | C 0 NST(5) | GM gravitational array ( $\mathrm{km}^{3} / \mathrm{sec}^{2}$ ) |
| I | G | 1 | C $\emptyset \mathrm{NST}(45)$ | Earth's surface gravity ( $\mathrm{km} / \mathrm{sec}^{2}$ ) |
| $\emptyset$ | TFINAL | 1 | INPUT(4) | Integration stop time (sec) |
| I | WTØ | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| $\emptyset$ | ТСФМР | 10 | INPUT(170) | Time (sec) to change integration step size |
| $\emptyset$ | DELT | 10 | INPUT(180) | Integration step size (sec) |
| I | TWD1 | 3 | INPUT(320) | Times (sec) for weight flow rate changes |
| I | WDØT1 | 3 | INPUT(350) | Weight flow rate changes ( $\mathrm{kg} / \mathrm{sec}$ ) |
| $\emptyset$ | TIG | 1 | INPUT(380) | Ignition time (sec) |
| $\emptyset$ | TB $\varnothing$ | 1 | INPUT(383) | Burnout time (sec) |
| I | ASPMC | 1 | INPUT(441) | Specific impulse of the midcourse motor (sec) |
| Ø | KRASH | 1 | $\begin{gathered} \text { INPUT } \\ (1032) \\ \hline \end{gathered}$ | Trajectory stopping key |
| $\emptyset$ | KMETH | 1 | $\begin{gathered} \text { INPUT } \\ \quad(1036) \\ \hline \end{gathered}$ | Trajectory computation method |
| I | IBURN | 1 | $\begin{gathered} \hline \text { INPUT } \\ (1071) \\ \hline \end{gathered}$ | Burn computation method key |


| I/ $\varnothing$ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | PROGRAM DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \\ \hline \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | XMC | 6 | $\begin{array}{r} \hline \mathrm{MCCDM} \\ (6) \\ \hline \end{array}$ | Pre-maneuver state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| I | DV | 3 | $\begin{gathered} \hline \mathrm{MCCDM} \\ (12) \\ \hline \end{gathered}$ | Midcourse velocity impulse (km/sec) |
| I | TMCS | 1 | $\begin{array}{\|c\|} \hline \mathrm{MCC} \varnothing \mathrm{M} \\ (18) \\ \hline \end{array}$ | Midcourse initiation time (sec) |
| I | KBURN | 1 | MCCD M $(154)$ | Burn maneuver computation key |
| I | ICB | 1 | $\begin{array}{\|c\|} \hline \text { MCC } \emptyset \mathrm{M} \\ (164) \\ \hline \end{array}$ | Central body of XMC |
| $\emptyset$ | X | 6 | STATE(1) | Post-maneuver state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| $\varnothing$ | T | 1 | STATE(10) | Post-maneuver time (sec) |
| $\emptyset$ | ATT | 3 | STATE(11) | Thrust attitude (unit vector) |
| $\emptyset$ | WEIGHT | 1 | STATE(34) | Spacecraft weight (kg) |

## Method:

MCBURN's method for computing the post-midcourse state and time is determined by the input key, KBURN.

## Impulsive (KBURN.LE.0)

In this case, the input velocity impulse, DV, is simply added to the pre-midcourse velocity to form the post-midcourse state.

$$
\begin{align*}
& \mathrm{X}=\mathrm{XMC}+\left[\begin{array}{c}
0 \\
\mathrm{DV}
\end{array}\right]  \tag{1}\\
& \mathrm{T}=\mathrm{TMCS} \tag{2}
\end{align*}
$$

Finite Burn (KBURN.GT.0)
In this case, the input velocity impulse, DV, is used to calculate the burn duration and thrust attitude. The computation of burn duration assumes that the thrust magnitude is proportional to mass flow rate, that the mass flow rate decreases linearly in time, and that the velocity impulse is entirely attributable to thrust. Defining the characteristic velocity, $c$, of the velocity correction package as the product of gravity, G, and specific impulse, ASPMC, we write thrust as follows:

$$
\begin{equation*}
|\overrightarrow{\mathrm{T}}|=c \dot{\mathrm{~m}}=-\mathrm{c} \frac{\mathrm{dm}}{\mathrm{dt}} \tag{3}
\end{equation*}
$$

The minus sign is necessary here if we define $\dot{m}$ to be positive.
The acceleration due to thrust is thrust/mass, so the velocity impulse, $\delta \mathrm{v}$, due to thrusting for time $t_{b}$ is:

$$
\begin{equation*}
\delta v=\int_{o}^{t_{b}} \frac{|\vec{T}|}{m} d t=-c \ln \left[\frac{m\left(t_{b}\right)}{m(o)}\right] \tag{4}
\end{equation*}
$$

Equation (4) may be solved (given $\delta \mathrm{v}=|\mathrm{DV}|$ ) for the mass expenditure, $\delta \mathrm{m}$.

$$
\begin{equation*}
\delta m=m(o)-m\left(t_{b}\right)=m_{o}\left(1-e^{-\delta v / c}\right) \tag{5}
\end{equation*}
$$

Under the assumption of a linear mass flow rate, we can solve for burn duration as follows:

$$
\begin{align*}
& \dot{\mathrm{m}}(\mathrm{t})=\dot{\mathrm{m}}(0)+\ddot{\mathrm{m}} \mathrm{t} \quad(\ddot{\mathrm{~m}} \leq 0)  \tag{6}\\
& \delta \mathrm{m}=\int_{0}^{t_{b}} \dot{\mathrm{~m}} d t=\dot{\mathrm{m}}(0) \mathrm{t}_{\mathrm{b}}+\frac{\ddot{\mathrm{m}}}{2} \mathrm{t}_{\mathrm{b}}^{2}  \tag{7}\\
& \dot{m}_{\mathrm{avg}}=\frac{1}{2}\left[\dot{\mathrm{~m}}(0)+\sqrt{\dot{\mathrm{m}}(0)^{2}+2 \ddot{\mathrm{~m}} \delta \mathrm{~m}}\right]  \tag{8}\\
& \mathrm{t}_{\mathrm{b}}=\frac{\delta \mathrm{m}}{\dot{\mathrm{~m}}} \underset{\mathrm{avg}}{ } \tag{9}
\end{align*}
$$

This form (9) of the solution to (7) is preferable to the more-standard quadraticequation solution because it does not require division by $\ddot{m}$ and reduces immediately to the linear-equation solution if $\ddot{\mathrm{m}}=0$. The thrust direction, $\hat{T}$, is computed parallel to DV .

$$
\begin{equation*}
\hat{T}=\frac{D V}{\delta \mathrm{~V}} \tag{10}
\end{equation*}
$$

The method (KBURN.NE.6) for computing the post-midcourse state for a finite burn is to integrate the equations of motion over the burn duration.

$$
\begin{align*}
& \mathrm{X}=\mathrm{XMC}+\int_{\mathrm{o}}^{\mathrm{t}_{\mathrm{b}}} \dot{\mathrm{X}} \text { (gravity + thrust) } \mathrm{dt}  \tag{11}\\
& \mathrm{~T}=\mathrm{TMCS}+\mathrm{t}_{\mathrm{b}} \tag{12}
\end{align*}
$$

Closed-Form Approximation (KBURN. EQ.6)
The double integration of the thrusting acceleration can be done in closed form if a mass flow rate which varies linearly with burn time is assumed. The superposition of this closed-form solution with the thrust-free state-change solution during the burn provides an excellent approximation to the numerically-integrated solution. In MCBURN, the closed-form solution is superimposed only on the
multi-conic thrust-free solution. The closed-form development follows. The position ( $\delta \mathrm{R}$ ) and velocity ( $\delta \mathrm{V}$ ) increments are presented.

$$
\begin{align*}
& \delta V=\int_{0}^{t_{b}} \frac{T}{m} d t=-c \hat{T} \ln \left[\frac{m\left(t_{b}\right)}{m(o)}\right]  \tag{13}\\
& \delta R=\int_{0}^{t_{b}} \delta V d t=-c \hat{T} \int_{0}^{t_{b}} \ell n \frac{m}{m_{0}} d t \quad\left[m=m(t), m_{o}=m(o)\right] \tag{14}
\end{align*}
$$

To evaluate the integral in (14), we note from Eqs. (6) and (7) that

$$
\begin{equation*}
\dot{\mathrm{dm}}=\ddot{\mathrm{m}} \mathrm{dt} \tag{15}
\end{equation*}
$$

and

$$
\begin{equation*}
\mathrm{m}=\mathrm{m}_{\mathrm{o}}-\frac{1}{2 \ddot{\mathrm{~m}}}\left(\dot{\mathrm{~m}}^{2}-\dot{\mathrm{m}}_{\mathrm{o}}^{2}\right) \tag{16}
\end{equation*}
$$

The integral is

$$
\begin{align*}
\int_{o}^{t_{b}} \ln \frac{m}{m_{o}} d t & =\frac{1}{\ddot{m}} \int_{\dot{m}_{o}}^{\dot{m}\left(t_{b}\right)} \ln \left[1-\frac{1}{2 \ddot{m} m_{o}}\left(\dot{m}^{2}-\dot{m}_{o}^{2}\right)\right] d \dot{m} \\
& =\frac{1}{\ddot{m}} \int_{\dot{m}_{o}}^{\dot{m}\left(t_{b}\right)} \ln \left[\dot{m}^{2}-2 \ddot{m} m_{o}-\dot{m}_{o}^{2}\right] d \dot{m}-\ddot{m} t_{b} \ln \left(-2 \ddot{m} m_{o}\right) \tag{17}
\end{align*}
$$

The integral in (17) is of the form $\int \ln \left(x^{2}+a^{2}\right) d x$, where

$$
a^{2}=-2 \ddot{m} m_{o}-\dot{m}_{o}^{2}
$$

is positive if $\ddot{\mathrm{m}}$ is less than $\left(-\dot{\mathrm{m}}^{2} / 2 \mathrm{~m}_{\mathrm{o}}\right)$.

$$
\begin{equation*}
\int \ln \left(x^{2}+a^{2}\right) d x=x \ln \left(x^{2}+a^{2}\right)-2 x+2 a \tan ^{-1} \frac{x}{a} \tag{18}
\end{equation*}
$$

Omitting the intermediate algebra, the resultant position change is

$$
\begin{equation*}
\delta R=\left\{\frac{\dot{m}\left(t_{b}\right)}{\ddot{m}}+2 \frac{c}{\delta v}\left[t_{b}+\frac{a}{\dot{m}} \tan ^{-1}\left(\frac{a t_{b}}{2 m_{o}-\dot{m}_{o} t_{b}}\right)\right]\right\} \delta V \tag{19}
\end{equation*}
$$

It may be shown (but not simply) that the limit of (19) as $\ddot{m}$ approaches zero is the constant mass flow rate solution,

$$
\begin{equation*}
\delta R=\left\{\frac{c}{\delta v} t_{b}-\frac{m\left(t_{b}\right)}{\dot{m}}\right\} \delta V \tag{20}
\end{equation*}
$$

The only reason why the closed-form solutions, (13) and (19), do not provide exact results (under model assumptions) when added to the thrust-free state at burnout time is that the gravitational acceleration depends on $\delta R(t)$. A simple extension is included in MCBURN's approximate burn to modify the velocity for gravitational accelerations resulting from $\delta R(t)$. It is assumed for this extension that the gravitational acceleration is $\left(-\mu \mathrm{R} / \mathrm{r}^{3}\right)$. The additional velocity, $\delta \delta \mathrm{V}$, is

$$
\begin{equation*}
\delta \delta V=-\frac{\mu}{r^{3}}\left(I-3 \hat{R} \hat{R}^{T}\right) \int_{0}^{t_{b}} \delta R(t) d t \tag{21}
\end{equation*}
$$

The integral of (21) can be evaluated as follows.

$$
\begin{align*}
& \begin{aligned}
\int_{0}^{t_{b}} \delta R(t) d t & =\int_{0}^{t_{b}}\left\{-\frac{\dot{m}(t)}{\dot{m}} c \ln \frac{m(t)}{m_{o}}+2 c\left[t+\frac{a}{\ddot{m}} \tan ^{-1}\left(\frac{a t}{2 m_{0}-\dot{m}_{o} t}\right)\right]\right\} d t \hat{\delta V} \\
& =\left\{\dot{m}\left(t_{b}\right) \frac{\delta r\left(t_{b}\right)}{\delta v}+\left(m\left(t_{b}\right)-\frac{c \delta m}{\delta v}\right) \delta V\right\} / \ddot{\mathrm{m}}
\end{aligned} \\
& \delta \delta V=-\frac{\mu}{3}\left(\delta V-3 \hat{R} \hat{R}^{T} \cdot \delta V\right)\left\{\dot{m} \frac{\delta r}{\delta v}+m-\frac{c \delta m}{\delta v}\right\} / \ddot{\mathrm{m}} \tag{22}
\end{align*}
$$

This velocity correction improves the closed-form solution. MCBURN is used in midcourse targeting. When entered with DV, it provides post-burn state where the burn time is computed from Eq. (9). The integrated state differences give rise to end-constraint errors which, through targeting, change DV. The changed DV does not then represent a velocity impulse, but rather an intermediate variable set which gives rise to a changed burn time and thrust direction through MCBURN's formulas. Such a procedure has programing advantages over a procedure which would switch targeting control parameters to time and thrust direction when passing from an impulsive burn to a finite burn. Logic of this subroutine is straightforward and requires no flowchart or block diagram.

Consider the time-segment $\left\{t_{i}: t\right\}$ on which $\dot{\mathrm{m}}$ is linear


$$
\begin{aligned}
& \delta R_{i}(t)=\int_{t_{i}}^{t}\left(-c \ln \frac{m(\tau)}{m_{o}}\right) d \tau \\
& \delta \delta V_{i}(t)=\int_{t_{i}}^{t} \delta R_{i}(\tau) d \tau
\end{aligned}
$$

Case 1: $\underline{\dot{m}}=+$ const. $, \quad m=m_{0}-\dot{m} \delta t \quad, \quad d\left(\frac{m}{m_{0}}\right)=-\frac{\dot{m}}{m_{0}} d \tau$

$$
\begin{aligned}
\delta R_{i}(t) & =-c \int_{0}^{t-t_{i}} \ln \left(1-\frac{\dot{m}}{m_{o}} \tau\right) d \tau \quad\left(\tau=t-t_{i}=\delta t\right) \\
& =-c\left[\frac{\dot{m}}{m_{o}} \tau\right. \\
-\frac{\dot{m}}{m_{0}} & \left.\ln \left(1-\frac{\dot{m} \tau}{m_{o}}\right)-\tau\right]_{0}^{t-t_{i}} \\
& =+\frac{c}{\dot{m}}\left[m_{o}-\dot{m} \tau \ln \left(\frac{m_{0}-\dot{m} \tau}{m_{o}}\right)+\dot{m} \tau\right]_{0}^{t-t_{i}} \\
& =\frac{c}{\dot{m}}\left[m \ln \frac{m}{m_{o}}+\dot{m} \delta t\right]=-\frac{m}{\dot{m}} \delta v+c \delta t=\delta v\left[\frac{c \delta t}{\delta v}-\frac{m}{\dot{m}}\right]
\end{aligned}
$$



## Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required:

## CALL MCSET

MCSET performs initializing calculations for midcourse and Monte Carlo analyses.

CONST, INPUT, MCCOM
None

Input / Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSION | $\begin{gathered} \hline \text { COMMON } \\ \text { BLOCK } \\ \hline \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | RAD | 1 | CONST(1) | Radian-to-degrees conversion factor |
| I | GM | 12 | CONST (5) | Gravitational constants $\left(\mathrm{km}^{3} / \mathrm{sec}^{2}\right)$ |
| I | DJL | 1 | INPUT(37) | Julian date at launch (days) |
| I | DJO | 1 | INPUT(46) | Julian date at anchor epoch (days) |
| I | COV | 6,6 | INPUT(56) | Tracking error covariance matrix |
| I | TWD | 10 | INPUT(320) | Times for weight flow rate table (sec) |
| I | WD | 10 | INPUT(350) | Weight flow rate table (kg/sec) |
| I | PSIDIN | 10 | INPUT(420) | Desired end constraint values ( km , deg, km/sec) |
| 1/O | SIGATM | 1 | INPUT(435) | Midcourse pointing error ( $\mathrm{deg} \rightarrow \mathrm{rad}$ ) |
| I/O | STGATR | 1 | INPUT(437) | Retro pointing error $(\mathrm{deg} \rightarrow \mathrm{rad})$ |
| I | TMCZIN | 1 | INPUT(440) | Second midcourse time (sec past DJL) |
| I | ASPMC | 1 | INPUT(441) | Specific impulse - midcourse motor (sec) |
| I | ASPR | 1 | INPUT(442) | Specific impulse - retro motor (sec) |
| I | WRETRO | 1 | INPUT(443) | Weight of retro fuel (kg) |
| I | RO | 1 | INPUT(444) | Desired circular orbit radius (km) |


| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I/O | NAME | DIMENSION | BLOCK | DEFINITION |
| I | BVDIN | 2 | INPUT(447) | Desired miss vector components (km) |
| I | DINK | 1 | INPUT(479) | Velocity step for partial generation ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | TOLIN | 10 | INPUT(490) | Constraint error tolerances |
| 1 | JT | 1 | INPUT(1031) | Target body number |
| I | MODE | 1 | INPUT(1044) | Mode switch <br> (3 midcourse, 4 Monte Carlo) |
| I | IBTRIN | 1 | INPUT(1062) | Target constraint type key |
| I | KGLAWI | 1 | INPUT(1063) | Guidance law selector |
| I | NGRPTI | 1 | INPUT(1064) | Number of trials to re-compute gradient |
| I | NTIN | 1 | INPUT(1065) | Number of targeting trials allowed |
| I | MCLIM | 1 | INPUT(1067) | Factor of DINK for limiting |
| I | IPROB | 1 | INPUT(1070) | Probability for scaling purposes (\%) |
| O | ALIMIT | 1 | MCCOM 1 ( | Velocity step limit (km/sec) |
| 0 | PRX | 1 | $\operatorname{MCCOM}(3)$ | Probability scale factor |
| O | DV | 3 | MCCOM $(12)$ | Initial-guess midcourse velocity impulse ( $\mathrm{km} / \mathrm{sec}$ ) |
| O | BVD | 2 | MCCOM 19$)$ | Desired miss vector (km) |
| O | DVB4 | 1 | $\operatorname{MCCOM}(24)$ | Expended midcourse velocity (km/sed |
| 0 | PFAC | 1 | MCCOM $(38)$ | Factor used in VTE law calculations (kg) |
| 0 | DJDIF | 1 | $\operatorname{MCCOM}(39)$ | Julian date difference, DJO-DJL (sec) |
| 0 | TMC2 | 1 | MCCOM(48) | Second midcourse time (sec past DJO) |
| O | PSID | 10 | $\operatorname{MCCOM}(80)$ | Desired end constraints |
| O | TOL | 10 | $\operatorname{MCCOM}(90)$ | End constraint tolerances |
| O | DW | 10 | $\operatorname{MCCOM}(110)$ | Expended fuel table (kg) wrt TWD |
| 0 | NT | 1 | MCCOM(151) | Number of targeting trials allowed |
| 0 | NP | 1 | MCCOM(160) | Number of constraints to test for convergence |


| SYMBOLIC |  | COMMON |  |
| :--- | :---: | :---: | :---: |
| I/O NAME DMMENSION BLOCK DEFINITION <br> 0 NGROPT 1 MCCOM(163) Number of trials to re-compute <br> gradient <br> 0 KGLAW 1 MCCOM(164) Guidance law indicator <br> 0 IBTR 1 MCCOM(167) Target constraint-type key <br> 0 IPD 3 MCCOM(169) Indicators for constraints to <br> test for convergence |  |  |  |

## Description:

This subroutine is straightforward and requires only a little explanation. First, the computation of the desired 5 th constraint is normally circular velocity at desired circular radius, $r_{d}=R O$, which is also the desired radius of closest approach, $r_{p}=\operatorname{PSD}(1)$.

$$
\operatorname{PSID}(5)=\sqrt{\frac{\mu}{r_{p}}}+\text { desired overburn velocity }
$$

The user may choose to target the VTE law to $r_{p}>r_{d}$ in just such a way that the apsidal radius opposite $r_{p}$ on the post-retro orbit is $r_{d}$.
in this case,

$$
\operatorname{PSID}(5)=\sqrt{\frac{\mu}{r_{p}}} \sqrt{\frac{2 r_{d}}{r_{p}+r_{d}}} .
$$

The second calculation worthy of mention is that of the expended fuel weight table, DW. It is computed by integrating the piecewise-linear mass flow rate table, WD.

$$
\delta W_{i+1}=\delta W_{i}+\int_{t_{i}}^{t_{i}+1} \stackrel{\circ}{W}_{i} d \tau=\sum_{j=1}^{j=1}\left(\stackrel{\circ}{W}_{j} \delta t_{j}+\frac{1}{2} \stackrel{\circ}{W}_{j} \delta t_{j}^{2}\right)
$$

The third calculation is of PRX, used in BELL for scaling the propagated ensemble error statistics. The input IPROB is interpreted as the desired Gaussian probability, $P(x)$.

$$
P(x)=\int_{-\infty}^{x} p(x) d x=\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)
$$

The equation, $\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)=I P R O B$, is solved iteratively with a Newton-Raphson predictor, beginning with the first guess, $x=1$.

## SUBROUTINE MCVERF

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required:

## CALL MCVERF

This subroutine controls the logic flow during Midcourse Verification Analysis

CNTRL, CONST, ELMNT, INPUT, MCCOM, OBSIT, PLNET, STATE

CROSS, DVMAG, FOWARD, MVTRN, M50EPM, M50LEQ, PUTELS, VNORM

Input / Output

| I/O | $\begin{gathered} \hline \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \\ & \hline \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ASPMC | 1 | INPUT(441) | Specific impulse of the midcourse motor. |
| I/O | ATT | 3 | STATE(10) | Attitude unit vector |
| I | BURNT | 1 | INPUT(476) | Midcourse motor burn time |
| I | DAY | 1 | INPUT(51) | Day of state epoch |
| 0 | DELT | 10 | INPUT(180) | Compute intervals |
| 0 | DELTO | 1 | INPUT(2) | Initial compute interval when in automatic compute interval mode |
| I | DJO | 1 | INPUT(46) | Modified julian date of state epoch |
| I | DJL | 1 | INPUT(37) | Modified julian date of liftoff epoch |
| I | DV | 3 | $\operatorname{MCCOM}(12)$ | Midcourse impulsive velocity increment |
| I | DX | 3 | STATE(4) | Spacecraft's velocity |
| 0 | ERRC | 1 | INPUT(1) | Error control limit of automatic compute interval |
| I | HR | 1 | INPUT(53) | Hour of state epoch |
| 1 | HRL | 1 | INPUT(23) | Hour of launch epoch |
| 1 | IDSAT | 1 | INPUT(1084) | Satellite identification number |


| SYMVBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I | ISET | 1 | HNPUT(1092) | Flag used to write out state for the orbit determination program |
| O | KMETH | 1 | INPUT(1036) | Initial trajectory propagation indicator |
| 0 | KOUT | 1 | INPUT(1030) | Output frequency flag |
| I | KREAD | 1 | INPUT(1057) | Flag to read initial state from midcourse tape |
| I | MCUNIT | 1 | INPUT(61) | Midcourse unit number |
| I | OBSLON | 10 | INPUT(410) | Longitudes of the tracking stations |
| I | SEC | 1 | INPUT(55) | Seconds of state epoch |
| I | SECL | 1 | INPUT(25) | Seconds of launch epoch |
| I | T | 1 | STATE (10) | Seconds since state epoch |
| 0 | TBO | 1 | INPUT(383) | Burnout time of midcourse engine (engine 1) |
| 0 | TCOMP | 10 | INPUT(170) | Switching times of compute interval table |
| 0 | TF | 1 | INPUT(4) | Final time |
| 0 | TIG | 1 | INPUT(380) | Ignition time of midcourse engine (engine 1) |
| I | W | 1 | INPUT(38) | Initial spacecraft mass |
| I | WT | 1 | STATE(35) | Current spacecraft mass |
| I | X | 3 | STATE(1) | Spacecraft's position vector |
| I | XMC | 6 | MCCOM(6) | Spacecraft's position and velocity vectors at midcourse motor ignition $\qquad$ |
| I | XMIN | 1 | INPUT(54) | Minutes of state epoch |
| I | XMINL | 1 | InPUT(24) | Minutes of launch epoch |
| I | XMON | 1 | INPUT(50) | Month of state epoch |
| 1 | XOBS | 10,3 | OBSIT(21) | Position vectors of tracking stations |
| I | XP | 6,12 | PLNET(1) | Position and velocity vectors of the planets |
| I | YEAR | 1 | InPUT(52) | Year of state epoch |

## Description:

The midcourse verification analysis is used to numerically integrate the midcourse motor and present output which describe conditions at motor ignition and at closest approach to the Moon. The conditions at motor ignition, burntime and burn attitude can be input via a tape generated in a previous midcourse analysis. The KREAD flag is used to determine if the tape option is to be used. If the tape is not read, the inttial conditions are in $X$ and the burntime and ignition time must be input via TBURN and TIG, respectively. The burn attitude is brought into the subroutine via ATT.

The initial conditions are propagated to midcourse motor ignition time using subroutine FOWARD. The compute interval is adjusted for the motor burn and the midcourse motor is numerically integrated using subroutine FOWARD. A flag is set to obtain the doppler output during the motor burn. This output is obtained from subroutine DOPLER. The state is written on a file to be used by the Orbit Determination Program if ISET is one. Subroutine PUTELS is used to write this file. Next the state is propagated to closest approach to the target planet using FOWARD. Information about the motor burn and conditions at the target planet are printed before the subroutine terminates. The information printed includes midcourse fuel expended, tracking station elevation, azimuth angles at motor ignition, and the state at the target planet.


Set midcourse motor ignition, burnout and attitude from tape


## SUBROUTINE MDCORS

## Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required: CROSS, DOT, DVMAG, FOWARD, JET, MVTRN, MCBURN, RETDV, SENSO, SPER, VNORM, TARGET

Input / Output

| I/O | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | DIMENSION | $\begin{gathered} \hline \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | JC | 1 | CNTRL(7) | Central body number |
| I | RTD | 1 | CONST(1) | Radian to degree conversion factor |
| I | TFINAL | 1 | INPUT(4) | Integration stop time (sec) from anchor epoch |
| I | WTO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| 1 | JTARG | 1 | INPUT(1031) | Target body number |
| 0 | KRASH | 1 | INPUT(1032) | Trajectory stop-type key |
| I | KMETH | 1 | INPUT(1036) | Trajectory computation method key |
| I | MCOUT | 1 | INPUT(1050) | Midcourse extra output key |
| I | MCKLUG | 1 | INPUT(1066) | Pre-targeting option key |
| I | IBURN | 1 | INसUT(1071) | Midcourse burn computation option key |
| I | KMETHP | 1 | INPUT(1075) | Initial trajectory computation method key |
| I | KTF | 1 | INPUT(1077) | Number of extra points in flight time scan |
| I | ALIMIT | 1 | $\operatorname{MCCOM}(1)$ | Control step limit ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | XMC | 6 | $\operatorname{MCCOM}(6)$ | Pre-ignition midcourse state <br> (km, km $/ \mathrm{sec}$ ) |


| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{gathered} \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I/O | DV | 3 | MCCOM 12 ) | Midcourse velocity impulse $(\mathrm{km} / \mathrm{sec})$ |
| 0 | DVS | 3 | $\operatorname{MCCOM}(15)$ | Spherical DV <br> (mag-km/sec, dec-deg, ra-deg) |
| I | TMCS | 1 | $\mathrm{MCCOM}(18)$ | Midcourse ignition time (sec) from anchor epoch |
| 0 | XSUN | 3 | MCCOM(21) | Spacecraft-to-sun vector at ignition (unit) |
| I | DVB4 | 1 | $\mathrm{MCCOM}(24)$ | Previous midcourse velocity used (km/sec) |
| 0 | DVRET | 1 | $\operatorname{MCCOM}(25)$ | Retro-velocity impulse magnitude ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | PFAC | 1 | MCCOM (38) | Propulsion factor ( $\mathrm{kg}, \mathrm{km} / \mathrm{sec}$ ) |
| 0 | WTF | 1 | $\mathrm{MCCOM}(47)$ | Spacecratt weight (kg) after midcourse burn |
| I | DPT | 3,10 | $\operatorname{MCCOM}(50)$ | Constraint/control sensitivity matrix transposed |
| I | PSID | 10 | MCCOM $(80)$ | Desired end condition vector |
| I | TOL | 10 | $\mathrm{MCCOM}(90)$ | Tolerances on end constraint errors |
| I | PSI | 10 | $\mathrm{MCCOM}(100)$ | End constraint error vector |
| I | NT | 1 | $\operatorname{MCCOM}(151)$ | Number of trials allowed in targeting |
| 0 | KBURN | 1 | MCCOM(154) | Burn computation method indicator |
| 0 | IT | 1 | MCCOM(157) | Running number of trials |
| 0 | IR | 1 | $\operatorname{MCCOM}(158)$ | Return key for targeting status |
| I | NP | 1 | $\operatorname{MCCOM}(160)$ | Number of constraints in targeting process |
| I | NGROPT | 1 | $\mathrm{MCCOM}(163)$ | Number of trials to re-compute gradients |
| 1 | KGLAW | 1 | MCCOM(164) | Guidance law indicator |
| O | ICB | 1 | MCCOM(165) | Midcourse central body number |
| 0 | ISP | 1 | MCCOM(166) | Gradient-computed indicator |
| I | IPD | 3 | MCCOM(169) | Constraint indicator vector |
| I | X | 6 | STATE(1) | Anchor vector state |
| I | T | 1 | STATE(10) | Anchor time (sec) anchor epoch, DJO |

## Description:

MDCORS is best described with reference to its accompanying flow charts. The first of these depicts the gross targeting logic.

The first step in MDCORS is to propagate the state, $\mathrm{X}_{2}$, forward from T to the midcourse time, TMCS, by calling FOWARD. The resulting pre-ignition midcourse state, XMC, is then saved. If the pre-targeting option key, MCKLUG, is positive, JET is then called to furnish a first-guess value for DV. If MCKLUG is zero, the starting value for DV is obtained from common where it was placed either by input or by previous targeting. Having thusly initialized, targeting is begun.

Subroutine SENSO is called to transform XMC and DV into an end constraint error vector, PSI. SENSO effects this transformation in the following three steps:

1. XMC and DV are converted into a post-maneuver state, X , and time, T , by subroutine MCBURN.
2. The post-maneuver state is propagated to the point and time of target closest approach by subroutine FOWARD.
3. The end state is used to compute end constraint function values which are subtracted from desired values to render the constraint error vector, PST.

PSI is dimensioned 10 , although only 8 of its components are used. These components represent the following errors:

| PSI(1) | B•T, miss-vector component |
| :--- | :--- |
| PSI(2) | B•R, miss-vector component |
| PSI(3) | Time of flight |
| PSI(4) | Hyperbolic excess velocity of arrival hyperbola. |
| PSI(5) | Circuiar excess velocity after retro at periapsis |
| PSI(6) | Total correction fuel expended (value, not error) |
| PSI(7) | Radius at periapsis of the arrival hyperbola |
| PSI(8) | Inclination of hyperbola to target's equator |

The tolerance array, TOL, is similarly defined, so that the criterion for convergence is

$$
\text { PSI (I) } \leq \operatorname{TOL}(\mathrm{I})
$$

for each of the end constraints associated with the particular guidance law in force. Each of the guidance laws available in MDCORS constrains the same first two functions: $B \cdot T$ and $B \cdot R$ if IBTR=1 or radius at periapsis and inclination if IBTR $=2$. (IBTR is an input quantity used by AUTO to set the array, IPD). The remaining function constrained by each law is:

| 1. Minimum (midcourse) fuel | None |
| :--- | :--- |
| 2. Fixed time of arrival | $\operatorname{PSI}(3)$ |
| 3. Fixed target energy | $\operatorname{PSI}(4)$ |
| 4. Variable target energy | $\operatorname{PSI}(5)$ |
| 5. Minimum total fuel | $\operatorname{PSI}(3) *$ See MTF procedure <br> description |

If one or more of the errors for the guidance law in force exceeds tolerance, a new estimate of DV is computed. Details of this computation will be described later. If a new gradient is to be generated (iteration trial number less than input, NGROPT), SENSO is called with IR $=2$ to generate it. SENSO uses the secant method (or finite increments of DV) in repeating the three steps described above to compute sensitivities of PSI to variations in DV. The result is interpreted as $\frac{\partial P S I}{\partial D V}$ evaluated at DV. Its transpose, a $3 \times 10$ matrix is stored in DPT. The last two columns of DPT are undefined. When the iteration trial number equals or exceeds NGROPT, no new gradient is computed. In this case, the last-computed DPT is used for succeeding iteration trials in computation of DV. By not computing new gradients at each trial, three trajectories per trial may be saved. The resulting deterioration in convergence is small if DV is "near" its tinal value when gradient computation is terminated. Results show that when pre-targeting is performed, a single evaluation of the gradient at the first guess DV is optimal in terms of total trajectories required and run time.

Before discussing the computation of DV, let us consider the post-targeting logic. This consists of tests for jumping back into the targeting logic. First, if a minimum midcourse
fuel iteration has converged on the 0th step, another step is forced. This is done because only the first two constraint errors are tested for the MFG law. Tne solution DV for any other law would satisfy these two constraints as well, without necessarily being a minimum DV or "critical plane" maneuver. The second reason for jumping back into the targeting logic is due to the minimum total fuel law implementation. The reason will become apparent when this implementation is described. The third jump-back is because of the approximate-intermediatetrajectory capability. The initial estimate of DV is obtained using trajectory method KMETHP, which should be set to 6 for multi-conic for translunar trajectories. If it is desired that DV be estimated for a more precise trajectory computation method, KMETH $\neq$ KMETHP, the initial estimate of DV is used to re-start the KMETH targeting procedure. It should never be necessary to re-generate gradients for this additional targeting.

## Estimation of $\Delta V$

The method of calculation of $\Delta V$ is dependent on the guidance law in force, although each guidance law is designed to constrain two common end conditions, i.e., B•T and $B \cdot R$ or radius at periapsis and inclination. It has been found that even when the latter conditions are to be constrained, $B \cdot T$ and $B \cdot R$ errors should be used to calculate $\Delta V$. That is, desired values of radius at periapsis and inclination should be used, together with hyperbolic excess velocity (vector), to compute the "desired" B•T and B•R and their corresponding errors, $\Psi_{1}$ and $\mathbb{\Psi}_{2}$. When these errors are nulled, the radius of periapsis and inclination errors will be nulled also. Formulae for computing "desired" $\mathrm{B} \cdot \mathrm{T}$ and $\mathrm{B} \cdot \mathrm{R}$ are to be found in Reference 1.

Minimum (midcourse) Fuel Guidance (MFG)
The MFG law minimizes the magnitude of $\Delta V$ while constraining the $B \cdot T$ and $B \cdot R$ errors, $\Psi_{1}$ and $\Psi_{2}$. It is desirable from a programing point of view to avoid numerical minimization procedures if possible - which it is in this case. We could null $\Psi_{1}$ and $\Psi_{2}$ (linearly speaking) with any $\Delta V$ such that

$$
-\binom{\Psi_{1}}{\Psi_{2}}=\binom{\mathrm{R}_{1}}{\mathrm{R}_{2}} \Delta V
$$

where $R_{1}$ and $R_{2}$ are rows of the $2 \times 3$ gradient matrix, $\frac{2 \Psi}{2 \Delta V}$. The family of such $\Delta V^{\prime}$ s is

$$
\Delta v=-\frac{\left(\begin{array}{llllll}
R_{2}^{\mathrm{t}} \times \mathrm{R}_{3}^{\mathrm{t}} & \mathrm{R}_{3}^{\mathrm{t}} \times \mathrm{R}_{1}^{\mathrm{t}} & \mathrm{R}_{1}^{\mathrm{t}} \times \mathrm{R}_{2}^{\mathrm{t}}
\end{array}\right)}{\left(\mathrm{R}_{1} \times \mathrm{R}_{2}\right)^{\mathrm{t}} \mathrm{R}_{3}}\left(\begin{array}{c}
\Psi_{1} \\
\Psi_{2} \\
\Psi_{3}
\end{array}\right)
$$

where $J_{3}$ is a free parameter and $R_{3}$ is any vector which is linearly independent of $R_{1}$ and $R_{2}$. That is, if we adjoin a third row, $R_{3}$, to the gradient, we can invert the results and solve for $\Delta V$. Making the substitution $C_{i}$ for columns of the inverse, we can write

$$
\Delta V=-\left(\begin{array}{lll}
C_{1} & C_{2} & C_{3}
\end{array}\right)\left(\begin{array}{l}
\Psi_{1} \\
\Psi_{2} \\
\Psi_{3}
\end{array}\right)=-C \Psi
$$

and $\quad \delta \mathrm{v}^{2}=\psi^{\mathrm{t}} \mathrm{C}^{\mathrm{t}} \mathrm{C} \psi$.
We now minimize $\delta \mathrm{v}^{2}$ with respect to the free parameter, $\Psi_{3}$, by setting the derivative to zero and solving for $\Psi_{3}$.

$$
\begin{aligned}
\frac{\mathrm{d} \delta \mathrm{v}^{2}}{\mathrm{~d} \Psi_{3}} & =2 \mathrm{Us}^{\mathrm{t}} \mathrm{C}^{\mathrm{t}} \mathrm{C}\left(\begin{array}{l}
0 \\
0 \\
1
\end{array}\right)=2\left(\Psi_{1} \mathrm{C}_{1}+\Psi_{2} \mathrm{C}_{2}+\Psi_{3} \mathrm{C}_{3}\right)^{\mathrm{t}} \mathrm{C}_{3}=0 \\
\Psi_{3} & =-\frac{\left(\Psi_{1} \mathrm{C}_{1}+\Psi_{2} \mathrm{C}_{2}{ }^{\mathrm{t}} \mathrm{C}_{3}\right.}{\mathrm{C}_{3}^{\mathrm{t}} \mathrm{C}_{3}}
\end{aligned}
$$

This $\mathbb{U}_{3}$ leads to the following $\wedge v$,

$$
\Delta V=-\left(I-\frac{\mathrm{C}_{3} \mathrm{C}_{3}^{\mathrm{t}}}{\mathrm{C}_{3}^{\mathrm{t}} \mathrm{C}_{3}}\right)\left(\mathrm{C}_{1} \mathrm{C}_{2} \mathrm{C}_{3}\right)\left(\begin{array}{l}
\Psi_{1} \\
\Psi_{2} \\
\Psi_{3}
\end{array}\right)
$$

where $\Psi_{3}$ can have any value, since $\left(\mathrm{I}-\frac{\mathrm{C}_{3} \mathrm{C}_{3}^{\mathrm{t}}}{\mathrm{C}_{3}^{\mathrm{t}} \mathrm{C}_{3}}\right) \mathrm{C}_{3}=0$. The vector, $\mathrm{C}_{3}$, is normal to the "critical plane" defined by $R_{1}$ and $R_{2}$, and the minimum-magnitude $\Delta V$ lies in this plane. If the adjoined row, $\mathrm{R}_{3}$, is defined as the cross-product of $\mathrm{R}_{1}$ and $\mathrm{R}_{2}$, the implemented solution is identical to the steepest ascent solution. The minimum fuel
guidance solution is computed by first computing

$$
\Delta \mathrm{V}^{*}=-\left(\begin{array}{lll}
\mathrm{C}_{1} & \mathrm{C}_{2} & \mathrm{C}_{3}
\end{array}\right)\left(\begin{array}{c}
\psi_{1} \\
\psi_{2} \\
0
\end{array}\right)
$$

and then projecting the result onto the critical plane.

$$
\Delta V=\Delta V^{*}-\left(\frac{C_{3}^{t} \Delta V^{*}}{C_{3}^{t} C_{3}}\right) C_{3}
$$

Since the constraint functions are not strictly linear functions of $\Delta V$, an iteration is required to find the solution, and the gradient varies during the iteration. And since the "critical plane" may change somewhat during an iteration, it is important to project the whole $\Delta V$ solution onto the current critical plane rather than simply to project the incremental $\Delta V$ due to residual constraint errors.

## Fixed Time of Arrival (FTA) Guidance

The end constraints for this law are $\mathrm{B} \cdot \mathrm{T} ; \mathrm{B} \cdot \mathrm{R}$ and time of flight. Representing errors in these constraints by $\Psi_{1}, \Psi_{2}$ and $\Psi_{3}$, respectively, we need a $\Delta V$ such that

$$
-\left(\begin{array}{l}
\psi_{1} \\
\psi_{2} \\
\psi_{3}
\end{array}\right)=\left(\begin{array}{l}
\mathrm{R}_{1} \\
\mathrm{R}_{2}^{1} \\
\mathrm{R}_{3}
\end{array}\right) \Delta \dot{\mathrm{V}}
$$

where $R_{i}=\frac{\partial \Psi_{i}}{2 \Delta V}$ is a row of the gradient, $\frac{\partial \Psi}{2 \Delta V}$. The solution is

$$
\Delta \mathrm{V}=-\left(\begin{array}{llll}
\mathrm{C}_{1} & \mathrm{C}_{2} & \mathrm{C}_{3}
\end{array}\right)\left(\begin{array}{l}
\psi_{1} \\
\psi_{2} \\
\psi_{3}
\end{array}\right)
$$

where

$$
\left(C_{1} C_{2} C_{3}\right)=\frac{1}{\left(R_{1} \times R_{2}\right)^{t} R_{3}}\left(\left(R_{2} \times R_{3}\right)^{t}\left(R_{3} \times R_{1}\right)^{t}\left(R_{1} \times R_{2}\right)^{t}\right)
$$

The only difference between the FTE and FTA laws is that the third constraint for the FTE law is hyperbolic excess speed with respect to the target at arrival.

## Variable Target Energy (VTE) Guidance

Application of the VTE law is peculiar to missions where the retro-burn at the target is an attitude-controllable fixed impulse. The VTE law is set up to calculate the midcourse maneuver $\Delta \mathrm{V}$ for which the post-retro velocity is a desired value (especially circular) while constraining $B \cdot T$ and $B \cdot R$ or closest approach distance and inclination.

The post-retro velocity is dependent upon the weight of fuel expended at midcourse and the arrival energy. Implementation of this law assumes that the retro-burn is executed at periapsis of the arrival hyperbola with the thrust deflected parallel to the periapsis velocity. The constraint error, $\Psi_{3}$, for this law is formulated as "circular excess" velocity.

$$
\Psi_{3}=\sqrt{\frac{\mu}{r_{d}}}+\varepsilon-\quad v_{\infty}^{2}+\frac{2 \mu}{r_{d}}-\delta v_{r}
$$

In this formulation, $r_{d}$ is the desired final orbit radius, $\epsilon$ is an input desired circular excess velocity, and $\delta \mathrm{v}_{\mathrm{r}}$ is the velocity impulse magnitude imparted by the retro-burn. Calculation of an appropriate midcourse $\triangle V$ for the VTE law can be formally identical to that for the FTA and FTE laws, given a good first-guess $\Delta V$ to start the iteration. An "assistance" procedure is beneticial, however, to reduce convergence difticulties which may arise from nonlinearities of $\Psi_{3}(\Delta \mathrm{~V})$ 。 This procedure consists of an iteration loop imbedded in the Newton-Raphson iteration loop to predict changes in $\delta \mathrm{v}_{\mathbf{r}}$ due to nextstep changes in $\triangle V$. Reference 1 contains a detailed description of this procedure. Minimum Total Fuel (MTF) Guidance

The implemented MTF guidance solution is the restricted solution which constrains approach conditions, $\mathrm{B}^{\circ} \mathrm{T}$ and $\mathrm{B} \cdot \mathrm{R}$ or radius at periapsis and inclination. In this respect, the implemented solution is neither general nor optimal. Results of tests indicate,
however , that for missions of the RAE-B type, the implemented solution will differ from the optimal by only a few hundredths of a kilogram in total fuel. The total correction fuel (when arrival conditions are constrained) is conveniently parameterized by flight time to target periapsis. The MTF solution is therefore obtained by solving for the FTA guidance solution within a loop which minimizes total fuel as a function of flight time. The minimization procedure will be found in the description of subroutine MINTF. The procedure is initiated at the MFG flight time, since the MFG solution is never far from the MTF solution. It is unnecessary to recompute gradients for obtaining rapid FTA convergence within the MTF procedure.

Reference 1 Bjorkman, W.S., Midcourse Guidance for Lunar and Planetary Orbiting Missions, AMA 71-16, March, 1971.



SUBROUTINE MINTF

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Called:

CALL MINTF
MINTF controls the gross logic for targeting the minimum total fuel guidance law.

MCCOM

None

Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I/O | PSSD | 10 | MCCOM ${ }^{(80)}$ | Desired constraint vector - esp. PSID $(3)=$ time of flight (sec) |
| I | TOL | 10 | $\mathrm{MCCOM}(90)$ | TOL(3) =flight time tolerance(sec) TOL (6)=total fuel tolerance ( kg ) |
| I/O | PSI | 10 | $\mathrm{MCCOM}(100)$ | $\begin{aligned} & \text { PSI }(3)=\text { fight time error }(\mathrm{sec}) \\ & \text { PSS }(6)=\text { total fuel }(\mathrm{kg}) \end{aligned}$ |
| 0 | KM | 1 | MCCOM $(152)$ | Cutput key for MDCORS logic |
| 1/O | JUMPTF | 1 | $\mathrm{MCCOM}(153)$ | Logic key for MDCORS \& MINTF |
| 0 | 13 | 1 | $\mathrm{MCCOM}(155)$ | Third constraint indicator |
| 0 | IT | 1 | MCCOM $(157)$ | Trial counter for MDCORS |
| 0 | NP | 1 | $\mathrm{MCCOM}(160)$ | Number of constraints |
| 0 | KGLAW | 1 | MCCOM(164) | Guidance law indicator |
| O | IPD | 3 | $\operatorname{MCCOM}(169)$ | Constraint indicator array |

## Description:

The minimum total fuel guidance law solution minimizes total correction fuel subject to equality constraints on closest approach distance and inclination. To effect this solution, the fixed time of arrival guidance (FTA) law is invoked with systematic variation of flight time. MINTF contains the logic for varying flight time to find the minimum total fuel solution.

The first time MDCORS calls MINTF, MINTF sets up indicators for the MFG (minimum midcourse fuel) law. When MINTF is called again, the MFG solution has been found, with its corresponding values of total fuel and time of flight. MINTF initializes the FTA law at this time and requests a flight time increase of 1800 seconds on return to MDCORS. On subsequent entries, MINTF seeks the minimum total fuel flight time by the following procedure:

1. If, on return from targeting the FTA law to $t_{M F G}+1800^{\mathrm{S}}$, it is found that the total fuel is,
a. less than it was for the MFG solution, flight time is stepped forward until the total fuel increases.
b. greater than it was for the MFG solution, flight time is stepped backward in $1800^{5}$. - steps until the total fuel increases.
2. The result of (1) is three values of total fuel, $f(t)$, corresponding to three different flight times. These obey the following inequality.

$$
f_{n-1} \geq f_{n} \geq f_{n+1}, \quad f_{n}=f\left(t_{M F G}+n * 1800^{s}\right)
$$

The three values are used to determine the coefficients of a quadratic equation describing total fuel as a function of flight time. This equation is then solved for the flight time corresponding to the minimum of the quadratic equation. (See MINV for details of fitting a quadratic equation。) This flight time is then used for the final FTA targeting step and the resultant total fuel is assumed to be the minimum value.

## SUBROUTINE MINTF



Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Called:

CALL MINV (ELM, RA, DEC, DVR, DVT, FFIRE)
MINV finds the retro-firing true anomaly of minimum trim velocity.

CONST, INPUT
CROSS, DOT, DVMAG, ORBIT, ORIENT, TRIM, VNORM

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| I | ELM | 12 | ARGUMENT <br> LIST | Orbital elements of the <br> hyperbola (LEQ) |
| I | RA | 1 | ARGUMENT <br> LIST | Right ascension of the <br> spin-axis (LEQ, rad) |
| I | DEC | 1 | ARGUMENT <br> LIST | Declination of the spin-axis <br> (LEQ, rad) |
| I | DVR | 1 | ARGUMENT <br> LIST | Retro velocity impulse (km/sec) |
| 0 | DVT | 1 | ARGUMENT <br> LIST | Trim velocity required (km/sec) |
| O | FFTRE | 1 | ARGUMENT <br> LIST | Firing true anomaly (rad) |
| I | RAD | 1 | CONST(1) | Radians-to-degrees conversion <br> factor (deg/rad) |
| I | GM | 12 | CONST(5) | Gravitational constant array <br> (km3/sec ${ }^{2}$ ) |
| I | TRUE | 1 | INPUT(475) | Limiting true anomaly allowed <br> (deg) |
| I | JT | 1 | INPUT(1031) |  | | Target body number |
| :--- |
| I |

The method used by MINV in trying to establish the minimum trim velocity and corresponding firing true anomaly is:

1. Step along the hyperbola in KFIRE regular steps from -TRUE to +TRUE, computing and storing required trim velocity at each step.
2. Fit a quadratic (trim velocity as a function of true anomaly) about the point of least trim velocity as found in procedure 1 . Solve for the minimum trim velocity and corresponding true anomaly as coordinates of the quadratic's minimum.

Subroutine TRIM is used by MINV to determine the trim velocity. TRIM, when supplied with the post-retro orbit (which must be elliptical), computes the trim velocity as a two-impulse in-plane Hohmann transfer plus a nodal inclinationtrim impulse. A large part of MINV's coding is concerned with establishing true anomaly bounds within which the post-retro orbit will be elliptical. These bounds are determined from the condition,

$$
V(f) \cdot \Delta V=\frac{\mu}{h}-\hat{P} \cdot \Delta V \sin \hat{t}+\hat{Q} \cdot \Delta V(e+\cos f)<-\frac{C_{3 h}+\delta V^{2}}{2}
$$

which must hold in the elliptical range。 $\hat{P}$ and $\hat{Q}$ are unit vectors toward periapsis and along periapsis velocity, respectively, $\mathrm{C}_{3 \mathrm{~h}}$ is the hyperbola's $\mathrm{C}_{3}$, and f is true anomaly. The other symbols are standard enough. The equation may be solved for $f$ (two solutions usually) by replacing the inequality with an equality. If the input limits are outside the computed bounds, they are replaced by the computed bounds as limits for the stepping procedure ( see 1 above).

If the least trim velocity from the stepping procedure occurs at the first or last step, this limit value is returned from MINV. Otherwise, the least trim velocity occurs at the i-th point and the quadratic fit coefficients, $a, b$ and $c$ are determined as follows,

$$
\begin{aligned}
& v(f)=a\left(f-f_{i}\right)^{2}+b\left(f-f_{i}\right)+c \\
& v\left(f_{i+1}\right)=a\left(f_{i+1}-f_{i}\right)^{2}+b\left(f_{i+1}-f_{i}\right)+c=v_{i+1}
\end{aligned}
$$

$$
\begin{aligned}
& v\left(f_{i}\right)=c=v_{i} \\
& v\left(f_{i-1}\right)=a\left(f_{i-1}-f_{i}\right)^{2}+b\left(f_{i-1}-f_{1}\right)+c=v_{i-1}
\end{aligned}
$$

Since the true anomaly step-size is constant,

$$
\delta f=\mathbf{f}_{i+1}-\mathbf{f}_{\mathbf{i}}=\mathbf{f}_{\mathbf{i}}-\mathbf{f}_{\mathbf{i}-1}
$$

the solution for the coefficients is

$$
\begin{aligned}
& a=\frac{1}{2 \delta f^{2}}\left(v_{i+1}+v_{i-1}\right)-2 v_{i} \\
& b=\frac{1}{2 \delta f}\left(v_{i+1}-v_{i}\right) \\
& c=v_{i}
\end{aligned}
$$

The minimum (extremum) occurs where

$$
\frac{d v}{d f}=2 a\left(f-f_{i}\right)+b=0
$$

or $\quad f_{\min }=f_{i}-\frac{b}{2 a}$.

The minimum trim velocity (returned by MINV) is computed by evaluating the quadratic for $v\left(f_{\text {min }}\right)$.


Calling Sequence:

## Purpose:

Common Blocks Required: ANKOR, CONST, INPUT, MCCOM, STATE
Subroutines:Called:

## CALL MONTE

MONTE performs a Monte Carlo analysis of success probability for the RAE-B mission.

BIGMAT, COVERT, FOWARD, MCBURN, MDCORS, MVTRN, M50LEQ, ORBIT, RANDM1, RETRO, SENSO, TRIM, VNORM

## Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | ANKVEC | 6 | ANKOR(1) | Anchor vector state $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| I | RAD | 1 | CONST(1) | Degrees per radian |
| I | TDS | 1 | CONST(43) | Seconds per day |
| I | TSH | 1 | CONST(42) | Hours per second |
| I | GM | 12 | CONST(5) | Gravitational constants (km $/ \mathrm{sec}^{2}$ ) |
| I | G | 1 | CONST(45) | Earth's surface gravity (km/sec) ${ }^{2}$ |
| I | TF | 1 | INPUT(4) | Trajectory stop time (sec) |
| I | DJL | 1 | INPUT(37) | Julian date at launch (days) |
| I | WO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | DJO | 1 | INPUT(46) | Julian date at anchor <br> epoch (days) |
| I | HRO | 1 | INPUT(53) | Hours of anchor epoch |
| I | XMINO | 1 | INPUT(54) | Minutes of anchor epoch |
| I | SECO | 1 | INPUT(55) | Seconds of anchor epoch |
| I | COV | 6,6 | INPUT(56) | Tracking covariance matrix |
| I | DTFIN | 1 | INPUT(422) | Desired time of flight (seconds) |


| SYMBOLIC |  |  | COMMON DEFINTTION |  |
| :---: | :---: | :---: | :---: | :---: |
| 1 | STGATM | 1 | INPUT(435) | Midcourse pointing error (rad) |
| I | SIGDVM | 1 | INPUT(436) | Midcourse velocity error (fraction) |
| I | SIGATR | 1 | INPUT(437) | Retro pointing error (rad) |
| I | SIGDVR | 1 | INPUT(438) | Retro velocity error (fraction) |
| I | TMC1 | 1 | INPUT(439) | First midcourse time (sec) |
| I | TMC2 | 1 | INPUT(440) | Second midcourse time (sec) |
| I | ASPMC | 1 | INPUT(441) | Specific impulse of midcourse and trim motor ( sec ) |
| I | ASPR | 1 | INPUT(442) | Specific impulse of retro motor ( sec ) |
| I | WRETRO | 1 | INPUT(443) | Weight of retro fuel ( kg ) |
| I | ATFULA | 1 | INPUT(470) | Available attitude maneuver fuel ( kg ) |
| I | FTOT | 1 | INPUT(472) | Available correction fuel (kg) |
| I | WDROP | 1 | INPUT(473) | Post-retro drop-weight (kg) |
| I | JT | 1 | INPUT(1031) | Target body number (11) |
| O | KCRASH | 1 | INPUT(1032) | Trajectory stop-type indicator |
| I | KMONTE | 1 | INPUT(1052) | Monte Carlo logic key |
| 1 | KMAX | 1 | INPUT(1053) | Monte Carlo sample size |
| 1/O | KSTART | 1 | INPUT(1054) | Random number kernel |
| 0 | MCKLUG | 1 | INPUT(1066) | Midcourse pre-targeting key |
| I | KMTOUT | 1 | INPUT(1074) | Extra output $\ddagger 7 \mathrm{lag}$ |
| I | KCOV | 1 | INPUT(1085) | Covariance matrix conversion key |
| 0 | XMC | 6 | MCCOM(6) | $\begin{aligned} & \text { Pre-ignition midcourse state } \\ & (\mathrm{km}, \mathrm{~km} / \mathrm{sec}) \end{aligned}$ |
| 1/O | DELV | 3 | MCCOM ${ }^{(12)}$ | Midcourse velocity impulse (km/sec |
| 1/O | DVMG | 1 | MCCOM $(15)$ | Midcourse velocity magnitude ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | TMC | 1 | MCCOM(18) | Midcourse ignition time (sec) |


| I/O | MBOLIC <br> NAME | DIMENSION $\begin{gathered}\text { COMMON } \\ \text { BLOCK }\end{gathered}$ |  | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | DV | 1 | MCCOM(24) | Accumulated correction velocity |
| 1/O | DPT | 30 | $\mathrm{MCCOM}(50)$ | Constraint/control sensitivity matrix |
| 0 | DTF | 1 | MCCOM(82) | Desired time of flight from anchor epoch (sec) |
| 0 | KENTRY | 1 | MCCOM(156) | Entry key for MDCORS |
| 0 | IR | 1 | MCCOM(158) | Sensitivity generation key for SENSO |
| 0 | NGROPT | 1 | MCCOM(163) | Gradient re-computation key for MDCORS |
| I | ISP | 1 | MCCOM(166) | Gradient-calculated indicator from SENSO |
| 0 | X | 6 | STATE(1) | Trajectory state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| 0 | T | 1 | STATE (10) | Time from epoch (sec) |
| I | ATT | 3 | STATE(11) | Spin-axis attitude (unit vector) |

## Description:

MONTE simulates the events of the RAE-B mission with sampled random errors to establish the probability of mission success. The events to be simulated for a given case are specified by KMONTE;*

KMONTE $=1 \quad$ Retro plus trim only - no midcourses
KMONTE $= \pm 2$ One midcourse correction plus retro and trim
KMONTE $= \pm 3$ Two midcourse corrections plus retro and trim
If KMONTE is negative, the first (or only) midcourse correction maneuver is calculated for the estimated (anchor vector) trajectory and not re-calculated for sampled errant anchor vectors. If KMONTE is +2 or +3 , the first (or only) midcourse correction is re-calculated for each sampled errant trajectory. The positive KMONTE case is applicable to pre-flight studies for which the "tracking" error covariance matrix in reality describes expected trajectory dispersions due to expected launch and injection errors.

The event logic as implemented is fairly straightforward as seen in the principal flow
chart. Some of the logic associated with midcourse maneuver computation requires some explanation, however. In order to avoid redundant computations in MDCORS, the first-maneuver correction impulse is computed and saved for the anchor vector trajectory. The constraint/control sensitivity matrix or gradient is also saved. If two midcourse maneuvers are simulated, the gradient at the second correction is also computed about the once-corrected anchor trajectory and stored. The stored gradients are good enough to bring about convergence for the midcourse maneuver calculations related to perturbed trajectories, so the gradient re-computation key, NGROPT, is set zero for the Monte Carlo process. The first-maneuver correction impulse computed for the anchor trajectory is a good first-guess for successive midcourse calculations, so the pre-targeting key, MCKLUG, is set zero as well. The first-guess value for the second maneuver can be taken as zero if the first maneuver execution errors are small. The logic shown in flow charts A and B avoids redundant and time-consuming calculations in MDCORS in the computation of midcourse correction maneuvers.

## Error Models

The tracking error covariance matrix, $P$, is a $6 \times 6$ positive definite matrix of anchor vector estimation errors.

$$
P=E \quad(x-\tilde{x})(x-\tilde{x}) \quad(\tilde{x}=\text { anchor vector, } x=\text { true state })
$$

There is a preferred coordinate system in which components of the error vector, $y$, are uncorrelated. The (similarity) transformation between $y$ and $x-\widetilde{x}$ is $S$, an orthogonal matrix which diagonalizes $P_{0}$.

$$
x-\tilde{x}=S_{y}
$$

The diagonal matrix, $D$, defined by

$$
D=S^{t} P S
$$

has as its diagonal elements the variances of the uncorrelated components of $y$. Scaling a white noise (uncorrelated random numbers) sampled by the standard deviations of $y$, we obtain an error vector in uncorrelated coordinates with components, yi.

$$
\mathrm{yi}=\mathrm{d}_{\mathrm{i}} \mathrm{n}(0,1)
$$

( $d_{i}$ is the $i$-th diagonal element of $D, n(0,1)$ is a sample random number from a normal (Gaussian) distribution of mean 0 and variance 1.) A sampled errant trajectory state is then computed from

$$
x=\widetilde{x}+S y
$$

Execution errors for the midcourse and retro maneuvers are computed from assigned standard deviations of pointing and velocity errors. The velocity error is treated as a proportional error normally distributed plus a resolution error uniformly distributed.

$$
\epsilon_{v}=v \sigma_{v} n(0,1)+(.0001 \mathrm{~km} / \mathrm{sec}) u
$$

( $v$ is the velocity impulse magnitude and $u$ is a random number from a distribution uniform on the interval $\{-.5<u<.5\}$ ). The retro velocity error is formulated without a resolution exror. The pointing error is formulated as two independent errors normally distributed along mutually orthogonal axes which are both orthogonal to the maneuver impulse direction. Let the maneuver impulse be denoted by V and its direction by $\tilde{\mathrm{V}}_{0}$ If $\hat{\mathrm{K}}$ is the unit polar axis of $V^{\mathfrak{t}} \mathrm{S}$ coordinate reference frame, we can construct unit vectors normal to $V$.

$$
\begin{aligned}
& \hat{E}=\frac{\hat{K} \times \hat{V}}{\hat{K} \times \hat{V}}=\frac{\begin{array}{c}
-v_{2} \\
v_{1}
\end{array}}{v_{1}^{2}+v_{2}}=\frac{-v_{2}}{v_{1}} \\
& \hat{v} \cos (\mathrm{dec}) \\
& \hat{N}=\hat{V} \times \ddot{E}=\frac{\hat{\mathrm{K}}-\hat{\mathrm{V}}(\hat{\mathrm{~K}} \cdot \hat{\mathrm{~V}})}{\cos (\operatorname{dec})}
\end{aligned}
$$

The "eastward" and "northward" pointing errors, $\theta_{e}$ and $\theta_{n}$, are computed by scaling random numbers with the input pointing erros, $\sigma_{a}$. They are then converted to velocity deviations using the small angle approximation.

$$
\begin{aligned}
& \epsilon_{e}=v \sigma_{a} n_{1}(0,1) \\
& \epsilon_{n}=v_{a} n_{2}(0,1)
\end{aligned}
$$

The misdirected velocity vector has the direction

$$
\hat{\mathrm{V}}^{\prime}=\left(\mathrm{v} \hat{\mathrm{~V}}+\epsilon_{\mathrm{e}} \hat{E}+\epsilon_{\mathrm{n}} \hat{N}\right) / \sqrt{v^{2}+\varepsilon_{\mathrm{e}}^{2}+\epsilon_{\mathrm{n}}^{2}}
$$

and the magnitude $v+\epsilon_{v}$, so that it may be programmed as follows,

$$
v^{\prime}=\frac{v+\epsilon_{v}}{v \sqrt{1+\sigma_{a}^{2}\left(n_{1}^{2}+n_{2}^{2}\right)}}\left\{\left(1-\frac{J_{a} N_{2} v_{3}}{v \cos (\operatorname{dec})}\right)\left(\begin{array}{l}
v_{1} \\
v_{2} \\
v_{3}
\end{array}\right)+\frac{J_{a}}{\cos (\operatorname{dec})}\left(\begin{array}{cc}
-n_{1} & v_{2} \\
n_{1} & v_{1} \\
n_{2} & v
\end{array}\right)\right\}
$$





## SUBROUTINE MOTORS

Calling Sequence:
Purpose:

CALL MOTORS (KMOD)
This subroutine determines thrusting status and increments velocity, or acceleration and weight accordingly.

Common Blocks Required: PERT, CNTRL, INTVAR, STATE, INPUT Subroutines Called:

TABINT

Input / Output

| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :--- | :--- | :--- | :--- |
| I | KMOD | 1 | Calling <br> Operand | Impulsive velocity flag |
| I/ $\varnothing$ | RCART | 3 | PERT(1) | Perturbing acceleration <br> vector |
| I | KDIS | 1 | CNTRL(05) | Discontinuity <br> Flag |
| I/ $\varnothing$ | KTHRST | 1 | CNTRL(02) | Thrusting Flag |
| I | KFIRST | 1 | CNTRL(12) | First step Flag |
| I | T | 1 | INTVAR(1) | Current time |
| I/ $\varnothing$ | THRUST | 1 | STATE(33) | Magnitude of thrust |
| I/ $\varnothing$ | WT | 1 | STATE(34) | Weight at engine ignition |
| I | ATT | 3 | STATE(11) | Attitude of spacecraft |
| $\emptyset$ | W | 1 | STATE(35) | Current weight |
| I/ $\varnothing$ | DX | 3 | STATE(04) | Current velocity |
| I | DV | 3 | INPUT(430) | Impulsive velocity |
| I | TIG | 3 | INPUT(380) | Ignition times |
| I | TBO | 3 | INPUT(383) | Burnout times |

Description: The engine characteristics can be simulated by either impulsive velocity or input thrust and weight flow histories. If KMOD is equal to 1 , impulsive velocity is assumed. The impulsive velocity of engine I is applied along the current attitude when the current time, $T$, is equal to the ignition time, TIG.

When KMOD is not equal to one, the instantaneous thrust and weight are determined from linear interpolation between time points on the input tables. Next the magnitude of the acceleration is determined from: $a=.001$ THRUST/WEIGHT .

The acceleration is added to the total perturbing acceleration vector of the spacecraft (PERT) according to the current attitude (ATT).


| Calling Sequence: | CALL MULCON |
| :--- | :--- |
| Purpose: | To propagate the state in time using the <br> Multiconic technique. |
| Common Blocks Required: | CNTRL, CONST, DUM, INPUT, INTVAR, <br> PLNET, STATE |
| Subroutines Required: | AVERGE, DVMAG, OBLATE, ORBIT, PLANET, <br> OUTPUT, PRINT, TOBODY |
| Reference: | D. Byrnes and H. Hooper, Multi-conic: <br> A Fast and Accurate Method of Computing |
| Space Flight Trajectories, AIAA |  |
| Paper No. 70-1062, 1970. |  |


| SYMBOLICICNAME |  |  COMIION <br> DLMENSION BLOCK |  | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | DELT | 10 | INPUT(180) | Compute Intervals |
| I | DJO | 1 | INPUT(46) | Epoch of the initial state |
| I/O | DX | 3 | STATE(4) | Spacecraft velocity |
| 0 | ELM | 6 | STATE(14) | Orbital elements |
| I/O | JC | 1 | CNTRL(7) | Central Planet |
| I | JL | 1 | INPUT(1015) | Launch planet |
| I | KCRASH | 1 | INPUT(1032) | Closest approach flag |
| I | KOBLE | 1 | INPUT(1018) | Earth oblateness flag |
| I | KOUT | 1 | INPUT(1030) | Output frequency flag |
| 1 | KP | 12 | INPUT(1001) | Planets in the system |
| I/O | T | 1. | STATE(10) | Current time since DJO |
| I | TCOMP | 10 | INPUT(170) | Switching times of compute interval table |
| I | TF | 1 | INPUT(4) | Final time |
| O | UJT | 1 | STATE(32) | Current julian date |
| I/O | X | 3 | STATE(1) | Spacecraft position |

Theory:
The Multiconic technique is an approximate technique to determine space trajectories. This technique differs from most approximate techniques and more resembles a numerical integration method as the independent variable, time, is stepped along and the state determined at the end of each step.

The geocentric equation governing the motion of a spacecraft perturbed by the Sun and Moon is as follows:
where $\quad \overline{\mathrm{R}}$ is the geocentric position vector of the spacecraft, $\bar{R}_{m}, \bar{R}_{s}$ are the geocentric position vectors of the Moon and Sun, respectively, $r_{m}, r_{s}$ are the vectors from the Moon and the Sun to the spacecraft, and $\mu_{\mathrm{e}}, \mu_{\mathrm{m}}{ }^{\prime} \mu_{\mathrm{s}}$ are gravitational constants of the Earth, Moon and Sun, respectively,

Each term of equations (1) describes an inverse square force and, taken separately, would yield simple conic motion. The Multiconic technique assumes that the trajectory can be simulated by sequentially summing these conics over several time intervals. However, the terms due to the Sun and the second term due to the Moon are slowly-varying functions. Therefore, without any significant loss in accuracy and with a substantial decrease in computation time, these terms are simplified. It is assumed that the acceleration due to these three terms can be approximated by a constant acceleration equal to the average accelerations at the beginning and at the end of the step applied over the entire step.

There is also the problem of separating the Keplerian orbit with respect to the Moon (second term of equation 1) from the other accelerations. Byrnes and Hooper solved this problem by flying backwards in a gravity free environment, an increment of time equal to the step, dt , and then flying forward along the selenocentric conic, a time increment dt.

The algorithm used to propagate the state from $\mathrm{T}_{1}$ to $\mathrm{T}_{2}$ is as follows:

1. The positions of the Moon and Sun are stored at time $\mathrm{T}_{1}$.
2. The geocentric state is propagated along the Earth conic from $T_{1}$ to $T_{2}$ using the Keplerian orbit defined by the first term of equation (1).
3. The positions of the Moon and Sun are stored at time $T_{2}$.
4. The mean acceleration due to the Moon's indirect term (third term of equation 1) is calculated as:

$$
\bar{a}_{1}=\frac{\mu_{m}}{2}\left[\left(\frac{\bar{R}_{m}}{\left|\mathrm{R}_{\mathrm{m}}\right|^{3}}\right)_{\mathrm{T}_{2}}+\left(\frac{\overline{\mathrm{R}}_{\mathrm{m}}}{\left|\mathrm{R}_{\mathrm{m}}\right|^{3}}\right)_{\mathrm{T}_{1}}\right]
$$

5. The mean acceleration due to the Sun is calculated as:

$$
\overline{\mathrm{a}}_{2}=\frac{\mu_{\mathrm{s}}}{2}\left[\left(\frac{\overline{\mathrm{r}}_{\mathrm{s}}}{\left|\overline{\mathrm{r}}_{\mathrm{s}}\right|^{3}}+\frac{\overline{\mathrm{R}}_{\mathrm{S}}}{\left|\overline{\mathrm{R}}_{\mathrm{S}}\right|^{3}}\right)_{\mathrm{T}_{2}}+\left(\frac{\overline{\mathrm{r}}_{s}}{\left.\overline{\mathrm{r}}_{\mathrm{s}}\right|^{3}}+\frac{\overline{\mathrm{R}}_{\mathrm{s}}}{\left|\overline{\mathrm{R}}_{s}\right|^{3}}\right)_{\mathrm{T}_{1}}\right]
$$

6. The state at the completion of the Earth conic (Step 2) is adjusted to account for the accelerations determined in Steps 4 and 5. The corrections are obtained from,

$$
\begin{aligned}
& \Delta \overline{\mathrm{V}}=\left(\bar{a}_{1}+\bar{a}_{2}\right) d t \\
& \Delta \overline{\mathrm{r}}=\frac{1}{2}\left(\bar{a}_{1}+\overline{\mathrm{a}}_{2}\right) d t^{2}
\end{aligned}
$$

where $\Delta \overline{\mathrm{V}}$ and $\Delta \overline{\mathrm{r}}$ are the changes in the position and velocity vectors and, dt is the step $\left(\mathrm{T}_{2}-\mathrm{T}_{1}\right)$
7. The corrected geocentric state is converted to selenocentric coordinates and propagated back in time along a straight line defined by the selenocentric velocity an amount dt.
8. The state is then propagated forward along the Keplerian selenocentric conic described by the second term in equation (1) from $T_{1}$ to $T_{2}$. This completes the algorithm and describes the state at the end of the step.

During the trans-Earth leg of a circumlunar trajectory, the algorithm is slightly modified. The new order is

1. Forward along the selenocentric conic.
2. Backward along a straight line.
3. Apply averaged accelerations.
4. Forward along an Earth-centered conic.

## Description:

The Mulcon subroutine is divided into three parts. The first part initializes the necessary flags and constants and determines the compute interval. The second part controls the logic flow of the Multiconic algorithm as the translunar leg of a trajectory. The third leg controls the logic flow of the Multiconic algorithm on the transearth leg of the trajectory.

The logic flow of this subroutine can more easily be understood by examining the accompanying flow chart. The transmoon leg starts at statement 50. If Earth oblateness is to be used, the orbital elements are adjusted due to $J_{2}$ and the spacecraft flown to approximately 5 Earth radii。 After that, the Multiconic algorithm is applied as described earlier. The test to determine if the spacecraft passed through pericynthion or if time is greater than the final time is made before statement 31. If these tests are not satisfied, the logic flow transfers to statement 10 where the next Multiconic interval begins. If pericynthion is passed, the time is adjusted to pericynthion and AVERGE called to reset the averaged accelerations. If the KCRASH flag is set to 1 , the state is translated back to a selenocentric system and flags set to begin the transearth leg. The transearth leg is similar to the translunar leg. Logic for this leg begins at statement 51.


CALL $\emptyset B L A T E:$ Orbital elements updated due to $J_{2}$ and spacecraft flown to 5 Earth radii on updated orbit.

Set times to hit next
compute point






## SUBROUTINE MVTRN

Calling Sequence:
CALL MVTRN(A, B, C, M, N)
CALL RめTATE(M, A, B, C)
Purpose:
To form the matrix product $C=A B$ or $C=A^{T} B$ where $A$ is a $3 \times 3$ matrix and $B$ and $C$ are $3 \times N$ matrices ( $3 \times 1$ in R $\varnothing$ TATE).

Common Blocks Required:
None
Subroutines Called:
None

Input/Output

| I/ $\varnothing$ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | PROGRAM DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | A | 9 | Call List | Matrix multiplier |
| I | B | 3, N | Call List | Matrix multiplier |
| $\emptyset$ | C | 3, N | Call List | Product matrix |
| I | M | 1 | Call List | Indicator: $C=A B$ if $M=1$ $C=A^{T} B$ otherwise |
| I | N | 1 | Call List | Number of columns of $B$ and $C$ |

Calling Sequence:
Purpose:

CALL M50EPM (UJT, C)
M50EPM calculates the transformation matrix from the mean equinox and equator of 1950 to the Earth's true equator and prime meridian.

CONST
NUTAIT, M50MDT

Input / Output

|  |  |  |  | COMMON |  |  |
| :--- | :---: | :---: | :--- | :--- | :---: | :---: |
| I/O | NYMME | DIMENSION |  |  |  |  |
| BLOCK |  |  |  |  |  |  |$\quad$ DE FINITION

Theory:
The transformation from mean equator and equinox of 1950 to true equator and prime meridian can be accomplished by first transforming to the mean equator and equinox of date (Call the matrix $[\mathrm{M}]$ ) and then from there to the true equator and equinox of date (Call the matrix [T]) and finally from equinox to the prime meridian ([P]).

Thus, if [C] is to be the transformation from mean of 1950 to true equator and prime meridian, then

$$
\lceil\mathrm{C}]=[\mathrm{P}]\lceil\mathrm{T}][\mathrm{M}]
$$

Both [T] and [M] are calculated in other subroutines (NUTAIT and M50MDT respectively). The [P] matrix represents the rotation through the Greenwich hour angle about the Z -axis. In general, a rotation through an angle $\theta$ about the positive Z -axis is accomplished through the following matrix

$$
\left[\begin{array}{ccc}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{array}\right]
$$

In this case, $\theta$ is the Greenwich hour angle, (GHA), which is given by .

$$
\begin{aligned}
\text { GHA }= & 100^{\circ} .075542+0^{\circ} .98564735(\mathrm{~d})+(2.9015) 10^{-13}(\mathrm{~d})^{2} \\
& +\omega(\mathrm{t})+(\text { Nutation in right ascension })
\end{aligned}
$$

$\omega$ is the angular velocity ( $\mathrm{deg} / \mathrm{sec}$ ) of the Earth and is given by

$$
\omega=.0041780742 /\left(1 .+(5.21) 10^{-13}(\mathrm{~d})\right)
$$

d is the number of whole days since 1950.
$t$ is the fraction of days (i.e. modified julian date $-33282.5=d+t)$

The nutation in right ascension is element $(2,1)$ of the matrix [T] from NUTAIT.

The three matrices are multiplied to give the total transformation matrix [C].

## Description:

M50EPM calculates the transformation matrix from the mean equator and equinox of 1950 to the Earth's true equator and prime meridian. This transformation is composed of three separate transformations.

The first transformation is from mean of 1950 to mean of date. This matrix is calculated in M50MDT. The second transformation is from mean of date to true equator and equinox of date. This matrix is calculated in NUTAIT. The third transformation is from true equator and equinox of date to true equator and prime meridian. This matrix is set up in this subroutine and consists of a rotation about the z-axis (North Pole) through the Greenwich hour angle. Once these transformation matrices are found, they are multiplied together to form the total transformation.


Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL M50JPM (UJT, B, J)
M50JPM calculates the transformation matrix from Earth mean equator and equinox of 1950 to true equator and prime meridian of date with respect to planet $J$.

CONST
M50EPM, M50LEQ, M50MDT

Input/Output

| SYMBOLIC |  |  | COMMONBLOCK |  |
| :---: | :---: | :---: | :---: | :---: |
| 0 | B | 3,3 | Calling Operand | Transformation matrix |
| I | J | 1 | Calling Operand | Planet number |
| 1 | RAD | 1 | CONST(1) | Degrees per radian |
| I | TDS | 1 | CONST(43) | Seconds per day |
| I | UJT | 1 | Calling Operand | Modified Julian Date |
| I | WP | 12 | CONST(29) | Spin rate of planets ( $\mathrm{rad} / \mathrm{sec}$ ) |

Theory:
The transformation from Earth mean of 1950 to true equator and prime meridian of date with respect to the desired planet is composed of two separate transformations. The first of these, Earth mean of 1950 to Earth mean of date, is calculated and explained in M50MDT. The second, Earth mean of date to true equator and prime meridian of date, with respect to the desired planet, is explained here.

The tranformation, $T$, involves three rotations about three axes.

$$
(\mathrm{T})=\left(\begin{array}{ccc}
\mathrm{CP} & \mathrm{SP} & 0 \\
-\mathrm{SP} & \mathrm{CP} & 0 \\
0 & 0 & 1
\end{array}\right) \quad\left(\begin{array}{ccc}
1 & 0 & 0 \\
0 & \mathrm{CD} & \mathrm{SD} \\
0 & -\mathrm{SD} & \mathrm{CD}
\end{array}\right) \quad\left(\begin{array}{ccc}
\mathrm{CR} & \mathrm{SR} & 0 \\
-\mathrm{SR} & \mathrm{CR} & 0 \\
0 & 0 & 1
\end{array}\right)
$$

where
$C R$ and $S R$ represent the cosine and sine of $R A+90^{\circ}$ respectively $C D$ and $S D$ represent the cosine and sine of $90^{\circ}$ - DEC respectively $C P$ and $S P$ represent the cosine and sine of PHE respectively
and
RA and DEC are the right ascension and declination of the planet's spin axis with respect to Earth mean equator and equinox of date.

PHE is the angle to the planet's prime meridian from the ascending node of the pianet's equator on the Earth's equator.

The values of RA, DEC and PHE are calculated through equations of the type

$$
\alpha=\alpha_{0}+\dot{\alpha}\left(t-t_{0}\right)
$$

where $\alpha_{0}$ is the value of $\alpha$ at $t_{0}$ and $\dot{\alpha}$ is the rate of change of $\alpha$ with respect to time.

The following constants are used for $\alpha_{0}, \dot{\alpha}$ and $t_{0}$,
MARS

| $\alpha_{0}$ | $\mathrm{t}_{0}$ (modified Julian date) | $\dot{\alpha}$ |
| :---: | :---: | :---: |
| RA | $317 .^{0} 3$ | 16846.024 |
| DEC. $522^{0} .7$ | 16846.024 | $6^{\circ} .^{0} 751 \times 10^{-3} / \mathrm{year}$ |
| PHE | $344 . .^{0} 41$ | 18322. |


| JUPITER |  |  |
| :---: | :---: | :---: |
| $\alpha_{0}$ | to $_{0}$ (modified Julian date) | $\dot{\alpha}$ |
| RA | 4.6775435 rad | 18673. |

*Note: The surface features of Jupiter are relatively unknown, so that the location of its prime meridian is a matter of speculation. For this reason, its initial position is set equal to that of Mars.

## Description:

M50rPM calculates the transformation matrix B from Earth mean of 1950 to true equator and prime meridian of date with respect to planet $J$. The date is given in terms of the modified Julian date UJT. At present, only the following planets may be used: Earth, Moon, Mars, and Jupiter (i.e., J $=3,11,4$ and 5 respectively).

Subroutines M50EPM and M50LEQ calculate the matrix B for the planet Earth and the Moon respectively, thus the calculation of $B$ for Mars and Jupiter represents the bulk of M50JPM. These calculations are done in three main steps. First, the right ascension and declination of the planet's spin axis with respect to the Earth mean equator and equinox of date are found in addition to the angle of prime meridian. These angles represent rotations which make up the transformation from Earth mean of date to true equator and prime meridian of date with respect to the planet. Second, the transformation from Earth mean of 1950 to Earth mean of date is calculated in M50MDT. Third, these matrices are multiplied to give the total desired transformation.

| Calling Sequence: |  |  | CALL M50LEQ (UJT, SELNEQ) |  |
| :---: | :---: | :---: | :---: | :---: |
| Purpose: |  |  | M50LEQ calculates the transformation matrix from the Earth mean equator and equinox of 1950 to the true lunar equator and prime meridian of date. |  |
| Common Blocks Required: |  |  | None |  |
| Subroutines Required: |  |  | NUTATE |  |
| Input / Output |  |  |  |  |
| 1/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSTION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| 0 | SELNEQ | 3,3 | Calling Operand | Transformation matrix |
| I | UJT | 1 | Calling Operand | Modified Julian Date |

Theory:
The transformation from Earth mean of 1950.0 to true lunar equator and prime meridian can be broken up into three transformations: (1) Earth mean equator and equinox of date to true lunar equator and ascending node, (2) Earth mean equator and equinox of 1950 to Earth mean equator and equinox of date, (3) True lunar equator and ascending node to true lunar equator and prime meridian.

The matrix for the first transformation is calculated in subroutine NUTATE, and the theory is described there.

The second matrix describes the precession of the Earth's mean equator since 1950.0. This precession is defined by the three small Euler angles

$$
\begin{aligned}
& \alpha=2304 . .^{\prime} 997(\mathrm{~T})+0!^{\prime} 302(\mathrm{~T})^{2}+0!^{\prime \prime} 019(\mathrm{~T})^{3} \\
& \beta=2304!^{\prime \prime} 997(\mathrm{~T})+1!^{\prime \prime} 093(\mathrm{~T})^{2}+0!^{\prime \prime} 0192(\mathrm{~T})^{3} \\
& \gamma=2004!^{\prime} 298(\mathrm{~T})-0.426(\mathrm{~T})^{2}+0 . .^{\prime \prime} 0416(\mathrm{~T})^{3}
\end{aligned}
$$

The transformation $A$ is defined by the three rotations

$$
A=\left[\begin{array}{ccc}
\cos \beta & -\sin \beta & 0 \\
\sin \beta & \cos \beta & 0 \\
0 & 0 & 1
\end{array}\right] \quad\left[\begin{array}{ccc}
\cos \gamma & 0 & -\sin \gamma \\
0 & 1 & 0 \\
\sin \gamma & 0 & \cos \gamma
\end{array}\right] \quad\left[\begin{array}{ccc}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{array}\right]
$$

The elements of [A] may be expanded in powers of $T$ to yield:
$a_{11}=1-.00029697(\mathrm{~T})^{2}-(0.13) 10^{-6}(\mathrm{~T})^{3}$
$\mathrm{a}_{12}=-\mathrm{a}_{21}=-.02234988(\mathrm{~T})-(0.676) 10^{-5}(\mathrm{~T})^{2}+(0.221) 10^{-5}(\mathrm{~T})^{3}$
$a_{13}=-a_{31}=-.00971711(\mathrm{~T})+(0.207) 10^{-5}(\mathrm{~T})^{2}+(0.96) 10^{-6}(\mathrm{~T})^{3}$
$\mathrm{a}_{22}=1-.00024976(\mathrm{~T})^{2}-(0.15) 10^{-6}(\mathrm{~T})^{3}$
$\mathrm{a}_{23}=\mathrm{a}_{32}=-.00010859(\mathrm{~T})^{2}-(0.3) 10^{-7}(\mathrm{~T})^{3}$
$a_{33}=1-.00004721(\mathrm{~T})^{2}+(0.2) 10^{-7}(\mathrm{~T})^{3}$
The third transformation involves the simple rotation about the $Z$-axis (perpendicular to the true equator of date and positive up) from the ascending node to the prime meridian. The angle between the ascending node and prime meridian is calculated in NUTATE, and its derivation is described there. If the angle is $\delta$, the rotation matrix is

$$
\mathrm{B}=\left[\begin{array}{ccc}
\cos \delta & \sin \delta & 0 \\
-\sin \delta & \cos \delta & 0 \\
0 & 0 & 1
\end{array}\right]
$$

Once these three matrioes have been calculated, the total transformation matrix is

$$
[\mathrm{T}]=[\mathrm{B}][\mathrm{m}][\mathrm{A}]
$$

where [N] is the matrix from NUTATE (for the transformation from Earth mean of date to true lunar equator and ascending node).

## Description:

M50LEQ first calls NUTATE to get BUMROT, the transformation matrix from Earth mean equator and equinox of date to Moon true equator with the ascending node on the ecliptic. Also output from NUTATE is RAPM which is the angle from the lunar equator ascending node on the ecliptic to the prime meridian.

The next step in the transformation is obtaining the transformation matrix A from Earth mean of 1950.0 to Earth mean of date. Then A is multiplied by BUMROT to get DUMROT, which is the transformation matrix from Earth mean of 1950.0 to Nioon true equator and ascending node.

The final transformation needed is from ascending node to prime meridian in the Moon's true equator of date. This involves a simple rotation about the Z-axis (perpendicular to the true equator and positive up) through the angle RAPM. This matrix is stored in BUMROT.

The final step is the multiplication of DUMROT by BUMROT which is stored in SELNEQ and is output as the total transformation matrix.


## SUBROUTINE M50MDT

Calling Sequence: $\quad$ CALL M50MD'T (T, A)
Purpose:
M50MDT calculates the transformation matrix from Earth mean equator and equinox of 1950.0 to Earth mean equator and equinox of date.

Common Blocks Required: None
Subroutines Required: None

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :---: |
| O | A | 3,3 | CALLING <br> OPERAND | Transformation Matrix |
| I | T | 1 | CALLING <br> OPERAND | Modified Julian Date |

## Description:

M50MDT calculates the time in Julian centuries and then calculates the elements of the matrix as polynomials in time.

## Theory:

The transformation from Earth mean equator and equinox of 1950.0 to Earth mean equator and equinox of date is accomplished by three rotations through three Euler angles that represent the precession of the Earth since 1950.0 .

The three angles are as follows:

$$
\begin{aligned}
& \alpha=2304 . " 997(\mathrm{~T} 1)+0 . " 302(\mathrm{~T} 1)^{2}+0 . " 019(\mathrm{~T} 1)^{3} \\
& R=2304 . " 997(\mathrm{~T} 1)+1 . " 093(\mathrm{~T} 1)^{2}+0 . " 0192(\mathrm{~T} 1)^{3} \\
& \nu=2004 . " 298(\mathrm{~T} 1)-0 . " 426(\mathrm{~T} 1)^{2}+0 . " 0416(\mathrm{~T} 1)^{3}
\end{aligned}
$$

where T 1 is the time in Julian centuries since the beginning of the Besselian year 1950.

The transformation matrix [A] is defined by

$$
A=\left[\begin{array}{ccc}
\cos R & -\sin R & 0 \\
\sin R & \cos R & 0 \\
0 & 0 & 1
\end{array}\right] \quad\left[\begin{array}{ccc}
\cos \gamma & 0 & -\sin \gamma \\
0 & 1 & 0 \\
\sin \gamma & 0 & \cos \gamma
\end{array}\right] \quad\left[\begin{array}{ccc}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{array}\right]
$$

The elements of A may be expanded into the following polynomials of T :

$$
\begin{aligned}
& \mathrm{a}_{11}=1-.00029697(\mathrm{~T} 1)^{2}-(0.13) 10^{-6}(\mathrm{~T} 1)^{3} \\
& \mathrm{a}_{12}=-\mathrm{a}_{21}=-.02234988(\mathrm{~T} 1)-(0.676) 10^{-5}(\mathrm{~T} 1)^{2}+(0.221) 10^{-5}(\mathrm{~T} 1)^{3} \\
& \mathrm{a}_{13}=-\mathrm{a}_{31}=-.00971711(\mathrm{~T} 1)+(.207) 10^{-5}(\mathrm{~T} 1)^{2}+(.96) 10^{-6}(\mathrm{~T} 1)^{3} \\
& \mathrm{a}_{22}=1-.00024976(\mathrm{~T} 1)^{2}-(0.15) 10^{-6}(\mathrm{~T} 1)^{3} \\
& \mathrm{a}_{23}=\mathrm{a}_{32}=-.00010859(\mathrm{~T} 1)^{2}-(0.3) 10^{-7}(\mathrm{~T} 1)^{3} \\
& \mathrm{a}_{33}=1-.00004721(\mathrm{~T} 1)^{2}+(0.2) 10^{-7}(\mathrm{~T} 1)^{3}
\end{aligned}
$$

## SUBROUTINE NUTAIT



## Description:

NUTAIT calculates the mean longitude of the Sun and Moon (VL \& CR) and the mean longitude of perigee of the Sun and Moon (G \& GP) as functions of time. Then the nutations of the obliquity of the ecliptic and longitude of the Sun (DE \& DT) are calculated as functions of $V L, C R, G$ and $G P$. The mean obliquity of the ecliptic is calculated as a function of time, and then the elements of the matrix are calculated and stored in EN for output.

Theory:
The transformation from mean equator to true equator can be approximated by

$$
\mathrm{EN}=\left[\begin{array}{ccc}
1 & -\delta \mathrm{L} \cos (\bar{\varepsilon}) & -\delta \mathrm{L} \sin (\bar{\epsilon}) \\
\delta \mathrm{L} \cos \bar{\epsilon} & 1 & -\delta \epsilon \\
\delta \mathrm{L} \sin \bar{\epsilon} & \delta \epsilon & 1
\end{array}\right]
$$

where the nutations $\delta \mathrm{L}$ and $\delta \epsilon$ are given by

$$
\begin{aligned}
& \delta \in \times 10^{4}=25^{\circ} .5844 \cos \bar{\Omega}+0^{\circ} .2456 \cos 2 \overrightarrow{\mathrm{~L}}^{\prime} \\
& -0.0_{2511} \cos 2 \bar{\Omega}+0.0508 \cos \left(2 \bar{L}^{\prime}-\bar{\Omega}\right) \\
& +1.5336 \cos 2 \overline{\mathrm{~L}}+0.0369 \cos \left(3 \overline{\mathrm{~L}}^{\prime}-\overline{\mathrm{T}}^{\prime}\right) \\
& +0^{\circ} 0666 \cos (3 \overline{\mathrm{~L}}-\overline{\mathrm{\Gamma}})-0.0^{0} 0139 \cos \left(\overline{\mathrm{~L}}^{\prime}+\bar{\Gamma}^{\prime}\right) \\
& -0.0258 \cos (\overline{\mathrm{~L}}+\bar{\Gamma})-0.0086 \cos \left(\bar{L}^{\prime}-\bar{\Gamma}^{\prime}+\bar{\Omega}\right) \\
& -0.0183 \cos (2 \overline{\mathrm{~L}}-\bar{\Omega})+0.0^{0} 0083 \cos \left(\bar{L}^{\prime} \bar{\Gamma}^{\prime}-\bar{\Omega}\right) \\
& -0.0067 \cos \left(2 \bar{\Gamma}^{\prime}-\bar{\Omega}\right)+00^{0} 0061 \cos \left(3 \bar{L}^{\prime}+\bar{\Gamma}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& +0^{\circ} .0064 \cos \left(3 \bar{L}^{\prime}-\vec{\Gamma}^{\prime}-\bar{\Omega}\right)
\end{aligned}
$$

$$
\begin{aligned}
& \delta \mathrm{L} \times 10^{-4}=-\left(47 .{ }^{\circ} 8927+0.0482 \mathrm{~T}\right) \sin \vec{\Omega}-0 .{ }^{\circ} 5658 \sin \left(2 \overline{\mathrm{~L}}^{\prime}\right) \\
& +0.5800 \sin (2 \bar{\Omega}) \\
& -0.0950 \sin \left(2 \bar{L}^{\prime}-\vec{\Omega}\right) \\
& -3^{\circ} .5361 \sin (2 \bar{L}) \quad-0.0725 \sin \left(3 \bar{L}^{\prime}-\bar{T}^{\prime}\right) \\
& -0.01378 \sin (3 \overline{\mathrm{~L}}-\overline{\mathrm{I}}) \quad+0.0317 \sin \left(\overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{I}}^{\prime}\right) \\
& +0.0594 \sin (\overline{\mathrm{~L}}+\overrightarrow{\mathrm{T}}) \quad+0.0161 \sin \left(\overline{\mathrm{~L}}^{\prime}-\overline{\mathrm{T}}^{\prime}+\bar{\Omega}\right) \\
& +0.0344 \sin (2 \overline{\mathrm{~L}}-\bar{\Omega}) \quad+0.0158 \sin \left(\overrightarrow{\mathrm{~L}}^{\prime}-\overline{\mathrm{T}}^{\prime}-\bar{\Omega}\right) \\
& +0.0125 \sin \left(2 \overline{\mathrm{~L}}^{\prime}-\Omega\right) \quad-0.0144 \sin \left(3 \overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{T}}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& +0^{\circ} .3500 \sin (\overline{\mathrm{~L}}-\overline{\mathrm{\Gamma}}) \quad-0^{\circ} .0122 \sin \left(3 \overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}-\bar{\Omega}\right) \\
& +0.0125 \sin (2 \overline{\mathrm{~L}}-2 \overline{\mathrm{~T}}) \quad+0.1875 \sin \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}\right) \\
& +0.0078 \sin \left(2 \mathrm{~L}^{\prime}-2 \vec{\Gamma}^{\prime}\right) \\
& +0.0414 \sin \left(\bar{L}^{\prime}+\bar{\Gamma}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& +0.0167 \sin \left(2 \bar{L}^{\prime}-2 \bar{L}\right) \\
& -0.0089 \sin \left(4 \bar{L}^{\prime}-2 \overline{\mathrm{~L}}\right)
\end{aligned}
$$

and

$$
\begin{aligned}
& \bar{L}^{\prime}=64^{\circ} .37545167+13.1763965268 \mathrm{~d} \\
& -{ }^{0} 001131575 \mathrm{~T}-.00113015 \mathrm{~T}^{2}+{ }^{\circ} .019 \times 10^{-4} \mathrm{~T}^{3} \\
& \bar{L}=280^{\circ} .08121009+0^{0} .9856473354 \mathrm{~d} \\
& +{ }^{\circ} 000303 \mathrm{~T}+.{ }_{0}^{0} 000303 \mathrm{~T}^{2} \\
& \bar{\Gamma}^{\prime}=208^{\circ} .8439877+0.1114040803 \mathrm{~d} \\
& -{ }^{0} 01334 \mathrm{~T}-{ }^{0} .010343 \mathrm{~T}^{2}-. .12 \times 10^{-4} \mathrm{~T}^{3} \\
& \bar{\Omega}=12.1127902-0.0529539222 \mathrm{~d} \\
& +{ }^{\circ} 0020795 \mathrm{~T}+{ }^{\circ} .002081 \mathrm{~T}^{2}+{ }^{\circ} .02 \times 10^{-4} \mathrm{~T}^{3}
\end{aligned}
$$

$$
\begin{aligned}
\overline{\mathrm{I}}= & 282{ }^{\circ} .08053028+0^{0 .} 470684 \times 10^{-4} \mathrm{~d} \\
& +.00045525 \mathrm{~T}+.0004575 \mathrm{~T}^{2}+{ }^{0} .03 \times 10^{-4} \mathrm{~T}^{3} \\
& \\
\overline{\mathrm{E}}= & 23.4457587-0.01309404 \mathrm{~T}-{ }^{0} .0088 \times 10^{-4} \mathrm{~T}^{2}+{ }^{0} .0050 \times 10^{-4} \mathrm{~T}^{3}
\end{aligned}
$$

where the above symbols represent the following quantities,

| Symbol | Quantity <br> Longitude of Sun, from the equinox of date |
| :--- | :--- |
| $L^{\prime}$ | Longitude of perigee of the Sun <br> Longitude of the Moon, measured in the <br> ecliptic from the mean equinox of date to <br> the ascending node of the lunar orbit, and <br> then along the orbit. |
| $\Gamma^{\prime}$ | Longitude of the perigee of the Moon, <br> measured as $L^{\prime}$ |
| $\Omega$ | Obliquity of the ecliptic |
| Longitude of the ascending node of the Moon's |  |
| T | orbit on the ecliptic |
|  | Time in days since 1950.0 |

Note: Barred quantities are mean values and $\delta \mathrm{L}$ represents the nutation in longitude of the Sun and $\delta \leqslant$ represents the nutation in obliquity.

## SUBROUTINE NUTATE

Calling Sequence: CALL NUTATE (K, TW, TF, TN, TM, RA, MG)
Purpose:
To compute the transformations from Earth's mean equator, equinox to Earth's and Moon's true equator, equinox and equator, node.

Common Blocks Required: None
Subroutines Required: None

Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | K | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \\ & \hline \end{aligned}$ | Output option key. |
| I | TW, TF | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \end{aligned}$ | Whole and fractional days since 1950.0, days |
| 0 | TN | 3,3 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \\ & \hline \end{aligned}$ | Earth's transformation matrix |
| O | TM | 3,3 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \\ & \hline \end{aligned}$ | Moon's transformation matrix |
| 0 | RA | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \end{aligned}$ | Right ascension of Moon's prime meridian, rad. |
| 0 | OMG | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \end{aligned}$ | Moon's rotation rate, rad/sec |

## Output Options

If $K=1$, only TN is computed, and the lunar orientation variables TM, RA, OMG are ignored.

If $\mathrm{K}=2$, TM, RA, OMG are computed, and the Earth orientation matrix, TN, is ignored. If $K=3$, all list items are computed.

## Equinox Coordinate Systems

Reference: Holdridge, D. B., "Space Trajectories Program for the IBM 7090 Computer." JPL Technical Report No. 32-223, Pasadena, California, March 2, 1962.

The cartesian coordinate systems defined using the vernal equinox and ecliptic or equatorial planes, and the transformations relating those systems are described below. Numerical expressions for the angles defining relative orientations are taken from the cited reference, though different computational forms are used. For completeness, the orientations of body-fixed systems are also treated here.

1. Notation

The symbols used for the angles follow the general rules:
a. Quantities not otherwise noted are true values of date.
b. Quantities superscribed by a bar are mean values.
c. Quantities subscripted " 50 " are 1950.0 values.
d. The prefix " $\delta$ " denotes a nutation or libration in the quantity prefixed.

In general, quantities defining Earth-Moon relationships use the symbol primed of the equivalent quantity from the Earth-Sun relationships. Unless otherwise stated, longitude is measured in the Earth's ecliptic, from the vernal equinox, and right ascension is measured in the equator, from the vernal equinox.

For each of the coordinate systems, the z-axis is taken normal to the reference plane, positive in the northern hemisphere, and the $x$-axis is taken along the reference direction in the reference plane.

The reference direction is the vernal equinox for the Earth-equinox systems, and the descending node of the Moon's equator on the ecliptic for the Moon-equinox systems. For the body-fixed systems, the reference direction is the prime meridian of the body.

A list of symbols is given in Table 1. The symbols used by the reference are also shown for ease of reference.

## 2. Equations

The computational equations are based on the equations given by the cited reference. Numerical values are taken from that report, but the form of many of the equations has been changed.

### 2.1 Earth's Mean Equator, Equinox of Date

The precession of the Earth's equator is defined by the small angles

$$
\begin{align*}
& \zeta_{0}=2304!997 \mathrm{~T}+0!302 \mathrm{~T}^{2}+0!0179 \mathrm{~T}^{3} \\
& \mathrm{z}=2304!997 \mathrm{~T}+1!093 \mathrm{~T}^{2}+0!0192 \mathrm{~T}^{3}  \tag{1}\\
& \theta=2004!298 \mathrm{~T}-0!426 \mathrm{~T}^{2}-0!0416 \mathrm{~T}^{3}
\end{align*}
$$



Ecliptic of date Ecliptic of 1950

Mean equator of date Mean equator of 1950

Figure 1
The transformation

$$
\begin{aligned}
A & =\left\{a_{i j}\right\} \\
& =\left[\begin{array}{ccc}
\cos z & -\sin z & 0 \\
\sin z & \cos z & 0 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{ccc}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{array}\right]\left[\begin{array}{ccc}
\cos \zeta_{0} & -\sin \zeta_{0} & 0 \\
\sin \zeta_{0} & \cos \zeta_{0} & 0 \\
0 & 0 & 1
\end{array}\right]
\end{aligned}
$$

may be expanded in powers of $T$. with the result

$$
\begin{align*}
& \mathrm{a}_{11}=--0.00029697 \mathrm{~T}^{2}-0.00000013 \mathrm{~T}^{3} \\
& \mathrm{a}_{12}=-\mathrm{a}_{21}=-0.02234988 \mathrm{~T}-0.00000676 \mathrm{~T}^{2}+0.00000221 \mathrm{~T}^{3} \\
& \mathrm{a}_{13}=-\mathrm{a}_{31}=-0.00971711 \mathrm{~T}+0.00000207 \mathrm{~T}^{2}+0.00000096 \mathrm{~T}^{3} \tag{3}
\end{align*}
$$

$$
\begin{aligned}
& a_{22}=1-0.00024976 \mathrm{~T}^{2}-0.00000015 \mathrm{~T}^{3} \\
& \mathrm{a}_{23}=\mathrm{a}_{32}=-0.00010859 \mathrm{~T}^{2}-0.00000003 \mathrm{~T}^{3} \\
& \mathrm{a}_{33}=1-0.00004721 \mathrm{~T}^{2}+0.00000002 \mathrm{~T}^{3}
\end{aligned}
$$

given by the reference.

### 2.2 Earth's True Equator, Equinox of Date

The transformation from mean equator to true equator is approximated by

$$
N=\left[\begin{array}{ccc}
1 & -\delta L \cos \bar{\epsilon} & -\delta L \sin \bar{\epsilon} \\
\delta L \cos \bar{\epsilon} & 1 & -\delta \epsilon \\
\delta L \sin \bar{\epsilon} & \delta \epsilon & 1
\end{array}\right]
$$

where the nutations $\delta \mathrm{L}, \delta \epsilon$ are given by
$\delta \mathrm{L} \times 10^{4}=-(47.8927+0.0482 \mathrm{~T}) \sin \bar{\Omega}-0_{0}^{0} 5658 \sin 2 \overline{\mathrm{~L}}^{\prime}$

$$
\begin{aligned}
& +0_{0}^{0} 5800 \sin 2 \bar{\Omega} \\
& -3_{0}^{0} 5361 \sin 2 \overline{\mathrm{~L}} \\
& -0_{0}^{0} 1378 \sin (3 \overline{\mathrm{~L}}-\overline{\mathrm{F}}) \\
& +0^{0} .0594 \sin (\overline{\mathrm{~L}}+\overline{\mathrm{T}})
\end{aligned}
$$

$$
+0.0344 \sin (2 \overline{\mathrm{~L}}-\bar{\Omega}) \quad 206 \quad+0^{\circ} .0158 \sin \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}-\bar{\Omega}\right)
$$

$$
\begin{aligned}
& \delta \in \times 10^{4}=25.5844 \cos \vec{\Omega} \quad+0.02456 \cos 2 \vec{L}^{\prime} \\
& -0.02511 \cos 2 \vec{\Omega} \quad+0.0508 \cos (2 \overrightarrow{\mathrm{~L}}-\vec{\Omega}) \\
& +1{ }^{\circ}{ }_{5336} \cos 2 \overline{\mathrm{~L}} \quad+0{ }_{0}^{\mathrm{o}} 0369 \cos \left(3 \overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{-}\right) \\
& +0^{\circ} .0666 \cos (3 \overline{\mathrm{~L}}-\overline{\mathrm{L}})-0.0139 \cos \left(\overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{T}}^{\prime}\right) \\
& -0.0258 \cos (\overline{\mathrm{~L}}+\overline{\mathrm{\Gamma}})-0.0086 \cos \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}+\bar{\Omega}\right) \\
& -0.0183 \cos (2 \bar{L}-\bar{\sim})+0.0083 \cos \left(\bar{L}^{\prime}-\bar{\Gamma}^{\prime}-\vec{\Omega}\right) \\
& -0_{0}^{\mathrm{o}} 0067 \cos \left(2 \bar{\Gamma}^{\prime}-\bar{\Omega}\right)+0.0061 \cos \left(3 \overline{\mathrm{~L}}^{\prime}+\bar{\Gamma}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& +0_{0}^{0.0064} \cos \left(3 L^{\prime}-\bar{\Gamma}^{\prime}-\bar{\Omega}\right)
\end{aligned}
$$

$$
\begin{array}{ll}
+0^{\circ} .0125 \sin \left(2 \bar{\Gamma}^{\prime}-\bar{\Omega}\right) & -0^{\circ} 0.0144 \sin \left(3 \overline{\mathrm{~L}}^{\prime}+\bar{\Gamma}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
+0^{\circ} .3500 \sin (\overline{\mathrm{~L}}-\overline{\mathrm{\Gamma}}) & -0.0122 \sin \left(3 \overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}-\bar{\Omega}\right) \\
+0^{\circ} 0.0125 \sin (2 \overline{\mathrm{~L}}-2 \overline{\mathrm{\Gamma}}) & +0.1875 \sin \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}\right) \\
& +0^{0.0078 \sin \left(2 \overline{\mathrm{~L}}^{\prime}-2 \bar{\Gamma}^{\prime}\right)} \\
& +0.0414 \sin \left(\overline{\mathrm{~L}}^{\prime}+\bar{\Gamma}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& +0^{0} .0167 \sin \left(2 \overline{\mathrm{~L}}^{\prime}-2 \overline{\mathrm{~L}}\right) \\
& -0.0089 \sin \left(4 \overline{\mathrm{~L}}^{\prime}-2 \overline{\mathrm{~L}}\right)
\end{array}
$$



$$
\bar{L}^{\prime}=64^{\circ} .37545167+13^{\circ} .1763965268 \mathrm{~d}
$$

$$
-0.001131575 \mathrm{~T}-0.00113015 \mathrm{~T}^{2}+0.019 \times 10^{-4} \mathrm{~T}^{3}
$$

$$
\overline{\mathrm{L}}=280^{\circ} .08121009+0^{\circ} .9856473354 \mathrm{~d}
$$

$$
+{ }^{o} .000303 \mathrm{~T}+{ }_{0}^{0} 000303 \mathrm{~T}^{2}
$$

$$
\begin{equation*}
\bar{\Gamma}^{\prime}=208.8439877+0^{\circ} .1114040803 \mathrm{~d} \tag{6}
\end{equation*}
$$

$$
-{ }^{0} .010334 \mathrm{~T}-{ }^{0} .010343 \mathrm{~T}^{2}-{ }^{0} .12 \times 10^{-4} \mathrm{~T}^{3}
$$

$$
\bar{\Omega}=12^{\circ} .1127902-0^{\circ} 0529539222 \mathrm{~d}
$$

$$
\begin{aligned}
& \quad{ }_{0}^{0} 0020795 \mathrm{~T}+{ }^{0} .002081 \mathrm{~T}^{2}+{ }^{0} .02 \times 10^{-4} \mathrm{~T}^{3} \\
& \bar{I}=282_{.08053028}^{0}+0^{0 .} .470684 \times 10^{-4} \mathrm{~d}
\end{aligned}
$$

$$
+{ }^{\mathrm{o}} .00045525 \mathrm{~T}+{ }^{o} .0004575 \mathrm{~T}^{2}+{ }^{o} .03 \times 10^{-4} \mathrm{~T}^{3}
$$

Equations (5) are computationally inefficient, requiring the evaluation of 39 sines and cosines. The subroutine NUTATE computes the transformation by computing the sines and cosines of the angles $\bar{¢}, \bar{\Omega}, \bar{\Gamma}-\bar{\Omega}, \bar{L}-\Omega, \bar{\Gamma}^{\prime}-\Omega, \bar{L}^{\prime}-\Omega$, with the remaining sines and cosines computed by trigonometric identities.

### 2.3 Earth-Fixed

The transformation from true equator, equinox of date to Earth-fixed coordinates is

$$
T=\left[\begin{array}{ccc}
\cos \xi & \sin \xi & 0  \tag{7}\\
-\sin \xi & \cos \xi & 0 \\
0 & 0 & 1
\end{array}\right]
$$

where

$$
\left.\begin{array}{rl}
\bar{\xi}= & 100^{\circ} .07554260+0^{0.9856473460[\mathrm{~d}]} \\
& \quad+(2.9015) 10^{-13}[\mathrm{~d}]^{2}+\omega \mathrm{t}
\end{array}\right] \begin{aligned}
\omega= & 0.00417807417 /\left(1+(5.21) 10^{-13}[\mathrm{~d}]\right) \mathrm{deg} / \mathrm{sec} \\
\delta \xi= & \delta \mathrm{L} \cos \bar{\epsilon} \\
\xi= & \bar{\xi}+\delta \xi
\end{aligned}
$$

The nutation in right ascension, $\delta \xi$ is the element $\mathrm{n}_{21}$ of the nutation matrix, equation (4)。

### 2.4 Moon's Equator, Node of Date

The transformation from Earth's equator, equinox to Moon's equator, node may be written as shown on the next page:
$M=\left[\begin{array}{ccc}\cos \Delta & \sin \Delta & 0 \\ -\sin \Delta & \cos \Delta & 0 \\ 0 & 0 & 1\end{array}\right] \quad\left[\begin{array}{ccc}1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i\end{array}\right] \quad\left[\begin{array}{ccc}\cos \Omega & \sin \Omega & 0 \\ -\sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1\end{array}\right]$
as in the reference, or
$M=\left[\begin{array}{ccc}1 & 0 & 0 \\ 0 & \cos \epsilon^{\prime} & \sin \epsilon^{\prime} \\ 0 & -\sin \epsilon^{\prime} & \cos \epsilon^{\prime}\end{array}\right] \quad\left[\begin{array}{cc}-\cos (\Omega+\delta L)-\sin (\Omega+\delta L) & 0 \\ \sin (\Omega+\delta L)-\cos (\Omega+\delta L) & 0 \\ 0 & 0\end{array}\right]$
where

$$
\begin{aligned}
& \epsilon^{\prime}=\bar{\epsilon}+\delta \epsilon^{\prime} \\
& \epsilon=\bar{\epsilon}+\delta \epsilon \\
& \Omega=\bar{\Omega}+\delta \Omega
\end{aligned}
$$

Retaining linear terms in $\delta \kappa, \delta L$,

$$
\begin{aligned}
& \mathrm{M}=\overline{\mathrm{M}} \mathrm{~N}^{\mathrm{T}} \\
& \overline{\mathrm{M}}=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos \epsilon^{\prime} & \sin \epsilon^{\prime} \\
0 & -\sin \epsilon^{\prime} & \cos \epsilon^{\prime}
\end{array}\right]\left[\begin{array}{ccc}
-\cos \Omega & -\sin \Omega & 0 \\
\sin \Omega & -\cos \Omega & 0 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos \bar{\epsilon} & \sin \bar{\epsilon} \\
0 & -\sin \bar{\epsilon} & \cos \bar{\epsilon}
\end{array}\right]
\end{aligned}
$$

where $N$ is the nutation matrix (4). Clearly, $\bar{M}$ is the transformation from Earth's mean equator, equinox to Moon's true equator, node.

The Earth's mean obliquity, $\bar{\epsilon}$, is given by (6). The mean inclination of the Moon's equator with the ecliptic, $\vec{\epsilon}^{\prime}$, is

$$
\bar{\epsilon}^{\prime}=1_{0}^{0} .535
$$

and the librations in inclination and longitude are

$$
\begin{aligned}
& \delta \epsilon^{\prime}=-0.0297222 \cos \left(\overline{\mathrm{~L}}^{\prime}-\overrightarrow{\mathrm{T}}^{\prime}\right)+0.0102777 \cos \left(\overrightarrow{\mathrm{~L}}^{\prime}+\vec{\Gamma}^{\prime}-2 \vec{\Omega}\right) \\
&-0.00305555 \cos \left(2 \overline{\mathrm{~L}}^{\prime}-2 \bar{\Omega}\right) \\
& \delta \Omega=\csc \bar{\epsilon}^{\prime}\left\{-0.0302777 \sin \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}\right)+0.0102777 \sin \left(\overrightarrow{\mathrm{~L}}^{\prime}+\bar{\Gamma}^{\prime}-2 \bar{\Omega}\right)\right. \\
&\left.-0.0030555 \sin \left(2 \overrightarrow{\mathrm{~L}}^{\prime}-2 \vec{\Omega}\right)\right\}
\end{aligned}
$$

The transformation $\overline{\mathrm{M}}$ is computed by subroutine NUTATE.

### 2.5 Moon-Fixed

The transformation from Moon's equator, node of date to equator, prime-meridian is

$$
L=\left[\begin{array}{cc}
\cos T^{\prime} & \sin Y^{\prime} 0 \\
-\sin Y^{\prime} & \cos T^{\prime} 0 \\
0 & 0
\end{array}\right]
$$

where

$$
\begin{aligned}
& r^{\prime}=L^{\prime}-\Omega \\
& \delta L^{\prime}=-0^{0} .003333 \sin \left(\bar{L}^{\prime}-\bar{\Gamma}^{\prime}\right)+0^{0} 00163888 \sin (\overline{\mathrm{~L}}-\bar{\Gamma}) \\
& \\
& \\
& \quad+0.005 \sin (2 \bar{\Gamma}-2 \bar{\Omega})
\end{aligned}
$$

The angular velocity of the Moon is the vector sum of the angular velocities

1. $\dot{L}^{\prime}-\dot{\Omega}$ about the Moon's polar axis, and
2. $\dot{\Omega}$ about the normal to the ecliptic.

Since the inclination of the two axes is $\epsilon$, the angular velocity is

$$
\omega^{\prime}=\left[\begin{array}{c}
0  \tag{14}\\
-\dot{\Omega} \sin \epsilon^{\prime} \\
\left(\dot{L}^{\prime}-\dot{\Omega}\right)+\dot{\Omega} \cos \epsilon^{\prime}
\end{array}\right]
$$

in equator, node components.

Subroutine NUTATE computes the polar component

$$
\begin{equation*}
\omega^{\prime}=\left(\dot{L}^{\prime}-\dot{\Omega}+\dot{\Omega} \cos \epsilon^{\prime}\right) \tag{15}
\end{equation*}
$$

from

$$
\begin{aligned}
\omega^{\prime}= & \dot{L}^{\prime}-\dot{\Omega}\left(1-\cos \bar{\epsilon}^{\prime}+\delta \epsilon^{\prime} \sin \bar{\epsilon}^{-\prime}\right) \\
1-\cos \bar{\epsilon}^{\prime}= & .000358852 \\
\sin \bar{\epsilon}^{\prime}= & .026787599 \\
\dot{\bar{L}}^{\prime}= & 0.266170762 \times 10^{-5}-0.12499171 \times 10^{-13} \mathrm{~T}(\mathrm{rad} / \mathrm{sec}) \\
\dot{\bar{\Omega}}= & -0.106969844 \times 10^{-7}+0.23015329 \times 10^{-13} \mathrm{~T} \\
\dot{\delta} \mathrm{~L}^{\prime}= & -0.1535272946 \times 10^{-9} \cos \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}^{\prime}\right)+0.569494067 \times 10^{-10} \cos (\overline{\mathrm{~L}}-\overline{\mathrm{T}}) \\
& +0.579473484 \times 10^{-11} \cos \left(2 \bar{\Gamma}^{\prime}-2 \bar{\Omega}\right)
\end{aligned}
$$

$$
\begin{gathered}
\Omega=-0.520642191 \times 10^{-7} \cos \left(\overline{\mathrm{~L}}^{\prime}-\bar{\Gamma}\right)+0.181177445 \times 10^{-7} \cos \left(\overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{\Gamma}}-2 \bar{\Omega}\right) \\
-0.106405786 \times 10^{-7} \cos \left(2 \overline{\mathrm{~L}}^{\prime}-2 \bar{\Omega}\right)
\end{gathered}
$$

Retaining nine significant figures,

$$
\begin{aligned}
& \omega^{\prime}=0.266171146 \times 10^{-5}-.13499 \times 10^{-9} \cos \left(\overline{\mathrm{~L}}^{\prime}-\overline{\mathrm{I}}\right) \\
&+.5695 \times 10^{-10} \cos (\overline{\mathrm{~L}}-\overrightarrow{\mathrm{I}})-.645 \times 1 \overline{0}^{11} \cos \left(\overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{\Gamma}}^{\prime}-2 \bar{\Omega}\right) \\
&+.380 \times 10^{-11} \cos \left(2 \overline{\mathrm{~L}}^{\prime}-2 \bar{\Omega}\right)+.579 \times 10^{-11} \cos \left(2 \bar{\Gamma}^{\prime}-2 \bar{\Omega}\right) \\
&+\delta \epsilon^{\prime} {\left[.\left[139.10^{-8} \cos \left(\overline{\mathrm{~L}}^{\prime}-\overline{\mathrm{T}}^{\prime}\right)-.485 \times 10^{-9} \cos \left(\overline{\mathrm{~L}}^{\prime}+\overline{\mathrm{T}}^{\prime}-2 \bar{\Omega}\right)\right.\right.} \\
&\left.+.285 \times 10^{-9} \cos \left(2 \overline{\mathrm{~L}}^{\prime}-2 \bar{\Omega}\right)\right]
\end{aligned}
$$

| Symbol | Reference | Quantity |
| :---: | :---: | :---: |
| L | L | Longitude of the Sun, from the equinox of date. |
| $\Gamma$ | $\Gamma$ | Longitude of perigee of the Sun. |
| g | g ${ }^{\prime}$ | True anomaly of the Sun |
| $L^{\prime}$ | $($ | Longitude of the Moon, measured in the ecliptic from the mean equinox of date to the ascending node of the lunar orbit, and then along the orbit. |
| $\Gamma^{\prime}$ | $\Gamma^{\prime}$ | Longitude of perigee of the Moon, measured as $\mathrm{L}^{\prime}$. |
| $\mathrm{g}^{\prime}$ | g | True anomaly of the Moon. |
| $\epsilon$ | $\epsilon$ | Obliquity of the ecliptic. |
| $\epsilon^{\prime}$ | I | Inclination of the Moon's equator with the ecliptic. |
| $\omega$ | $\omega$ | Angular velocity of the central body about its north polar axis. |
| $\xi$ | $\xi$ | Right ascension of the prime meridian (hour angle of the vernal equinox). |
| $\Omega$ | $\Omega$ | Longitude of the ascending node of the Moon's orbit on the ecliptic. |
| $\omega_{p}$ | $\omega$ | Argument of perigee of the Moon. |
| i | i | Inclination of the Moon's equator with the Earth's equator. |
| $\Omega^{\prime}$ | $\Omega^{\prime}$ | Right ascension of the ascending node of the Moon's equator on the Earth's equator. |
| $\Delta$ | $\Delta$ | Arc in the Moon's equator from the Earth's equator to the ecliptic. |


| Symbol | Reference | Quantity |
| :---: | :---: | :---: |
| $\xi^{\prime}$ | $\Delta-\Delta$ | Right ascension of the Moon's prime meridian, from the ascending node of the Moon's equator on the ecliptic. |
| $\delta^{\text {L }}$ | $\delta \Psi$ | Nutation in longitude of the Sun |
| $\delta \xi$ | $\delta \xi$ | Nutation in right ascension of the prime meridian. |
| $5 \Omega$ | $\sigma$ | Libration in longitude of the ascending node. |
| $\delta \mathrm{L}^{\prime}$ | t | Libration in longitude of the Moon. |
| $\delta \epsilon^{\prime}$ | 8 I | Libration in inclination. |
| $\delta \epsilon$ | $\delta \epsilon$ | Nutation in obliquity |
| $90^{\circ}+\mathrm{z}$ | $90^{\circ}+\mathrm{z}$ | Right ascension of the mean equator of 1950 , from the mean equinox of date. |
| $90^{\circ}-\zeta_{0}$ | $90^{\circ}-\zeta_{0}$ | Right ascension of the mean equator of date, from the mean equinox of 1950 . |
| $\theta$ | $\theta$ | Inclination of the mean equator of date with the mean equator of 1950 . |

## SUBROUTINE OBLATE

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:
References:

## CALL OBLATE

This subroutine calculates the approximate effect of the Earth's $J_{2}$ gravitational term on a trajectory. $\quad$ This subroutine is used with the Multiconic propagator.

CONST, DUM, INPUT, STATE.

ORBIT, TRMN.
Paul Penzo, Computing Earth Oblateness Effects on Lunar and Interplanetary Trajectories, AIAA paper No. 70-97, January, 1970.

Inputs / Outputs

| I/O | $\begin{array}{c}\text { SYMBOLIC } \\ \text { NAME }\end{array}$ | $\begin{array}{c}\text { PROGRAM } \\ \text { DIMENSION }\end{array}$ | $\begin{array}{c}\text { COMMON } \\ \text { BLOCK }\end{array}$ | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| $O$ | AM | 1 | DUM (9) | Mean anomaly at end of step |\(\left.| \begin{array}{l}Orbital elements and sines and <br>

cosines of elements of state\end{array}\right\}\)

Theory:
The approximate effect of $J_{2}$ on the orbital elements is determined in this subroutine. The time derivatives of the angular elements can be expressed by:

$$
\begin{aligned}
& \frac{d i}{d t}=\frac{r \cos \theta}{h} A_{n} \\
& \frac{d \Omega}{d t}=\frac{r \sin \theta}{h \sin i} \quad A_{n}
\end{aligned}
$$

$$
\frac{d \boldsymbol{\omega}}{d t}=-\cos i \frac{d \Omega}{d t}-\frac{1}{e h}\left[P \cos f A_{r}-(P+r) \sin f A_{t}\right]
$$

where,

$$
\begin{aligned}
\mathrm{r} & =\text { radial distance } \\
\mathrm{f} & =\text { true anomaly } \\
\omega & =\text { argument of perigee } \\
\mathrm{i} & =\text { inclination } \\
\Omega & =\text { longitude of the ascending node } \\
\Theta & =\omega+\mathrm{f} \\
\mathrm{~h} & =\text { angular momentum } \\
\mathrm{e} & =\text { eccentricity } \\
\mathrm{P} & =\text { semi-latus rectum }
\end{aligned}
$$

and where $A_{r}, A_{t}, A_{n}$ are the components of the disturbing acceleration caused by Earth oblateness, see Figure 1. Z


It is more convenient to exchange the true anomaly for time as the independent variable. To do this, the expression for angular momentum may be used, as

$$
h=r^{2} \frac{d f}{d t}
$$

then,

$$
\begin{equation*}
\frac{\mathrm{di}}{\mathrm{dt}}=\frac{\mathrm{di}}{\mathrm{df}} \quad \frac{\mathrm{df}}{\mathrm{dt}}=\frac{\mathrm{h}}{\mathrm{r}^{Z}} \frac{\mathrm{di}}{\mathrm{df}} \tag{2}
\end{equation*}
$$

Similar equations are used for the other elements. Thus equation (1) becomes,

$$
\begin{aligned}
& \frac{d i}{d f}=\frac{r^{3} \cos Q}{h^{2}} A_{n} \\
& \frac{d \Omega}{d \hat{f}}=\frac{r^{3} \sin \theta}{h^{2} \sin i} A_{n} \\
& \frac{d \omega}{d f}=-\cos i \frac{d \Omega}{d f} \cdot \frac{r^{2}}{e h^{2}}\left[P \cos f A_{r}-(P+r) \sin f A_{t}\right]
\end{aligned}
$$

The acceleration due to the second harmonic term is given by,

$$
\left\{\begin{array}{l}
A_{r}  \tag{4}\\
A_{t} \\
A_{n}
\end{array}\right\}=\frac{-3 \mu J_{2} r_{e q}^{2}}{2 r^{4}} \quad\left(\begin{array}{r}
-3 \sin ^{2} \theta \\
\sin ^{2} i \\
\sin ^{2} \theta \\
\sin ^{2} i \\
\sin \theta \\
\sin ^{2}{ }_{i}
\end{array}\right)
$$

where $\mu$ is the Earth's gravitational constant

$$
r_{\text {eq }} \text { is the Earth's equatorial radius }
$$

$$
J_{2}=.0010823
$$

The substitution of equation (4) into equation (3) provides the final form of the derivatives of the orbital elements. The only variable remaining besides $f$ and $\theta$ is $r$. However, $r$ can be eliminated by the use of

$$
\begin{equation*}
r=\frac{P}{1+e \cos f} \tag{5}
\end{equation*}
$$

Now, equations (3), (4) and (5) can be combined and analytically integrated to provide the effects of $J_{2}$ on the orientation elements. Since the changes in the elements are small through the integration interval, the orbital elements can be held constant during the integration for a first approximation to their variations. Thus, the variation in the elements becomes,

$$
\Delta i=K \sin i \cos i\left[\frac{1}{2} \sin ^{2} \theta-\frac{e}{3} \cos \omega \cos ^{3} \theta+\frac{e}{3} \sin \omega \sin ^{3} \theta\right]_{\theta_{0}}^{\theta_{1}}
$$

$$
\begin{align*}
\Delta \Omega=K \operatorname{cosi} & {\left[\frac{1}{2}(\theta-\sin \theta \cos \theta)+\frac{e}{3}\left(\cos \omega \sin ^{3} \theta+\sin \omega \cos ^{3} \theta\right)\right.}  \tag{6}\\
& \left.+e \sin \omega \sin ^{3} \theta\right]_{\vartheta_{0}}^{\theta_{1}}
\end{align*}
$$

$$
\Delta \omega=-\operatorname{cosi} \Delta \Omega+\Delta \omega_{\mathrm{r} \theta}
$$

where

$$
\begin{aligned}
& \begin{aligned}
K & =\frac{-3 \mu^{2} J_{2}{ }^{r} \mathrm{eq}}{h^{4}} \\
r_{\theta} & =\frac{-K}{2 e}\left[e\left(1-\frac{3}{2} \sin ^{2} i\right) f\right.
\end{aligned} \\
& \begin{array}{l}
\left.\quad+\left(1+\frac{3}{2} e^{2}\right) \sin f+e \sin f \cos f+\frac{e^{2}}{3} \cos ^{2} f \sin f\right]_{f_{1}}^{f_{1}} \\
+\frac{K}{2 e} \sin ^{2} i \sin \omega \cos \omega\left[4 \cos f+3 e \cos ^{2} f\right.
\end{array} \\
& \left.-\frac{2}{3}\left(7-e^{2}\right) \cos ^{3} f-6 e \cos ^{4} f-2 e^{2} \cos ^{5} f\right] \begin{array}{l}
f_{1} \\
f_{0}
\end{array} \\
& +\frac{K}{2 e} \sin ^{2} i\left(A \sin f+B \sin f \cos f+C \cos ^{2} f \sin f\right. \\
& \left.+3 D \cos ^{3} f \sin f+e D \cos ^{4} f \sin f\right) \begin{array}{l}
f_{1} \\
f_{0}
\end{array}
\end{aligned}
$$

where

$$
\begin{aligned}
& A=\frac{1}{3}\left(7+2 \mathrm{e}^{2}\right) \cos ^{2} \omega+\frac{3}{2}\left(1+2 \mathrm{e}^{2}\right) \sin ^{2} \omega \\
& B=\frac{3}{2} \mathrm{e} \\
& C=\frac{1}{3}\left(\mathrm{e}^{2}-7\right) \cos ^{2} \omega+\frac{1}{3}\left(7+2 \mathrm{e}^{2}\right) \sin ^{2} \omega \\
& D=-e\left(\cos ^{2} \omega-\sin ^{2} \omega\right)
\end{aligned}
$$

The in-plane elements, e and $p$, are determined if the energy, $c$, and the angular momentum, $h$, are known. The variation in the energy is determined since the force field of an oblate Earth is derivable from a potential function. Considering only the second harmonic term, the energy per unit mass is,

$$
\begin{equation*}
\mathrm{C}=\frac{\mathrm{v}^{2}}{2}-\frac{\mu}{\mathrm{r}}-\frac{\mu \mathrm{J}_{2} \mathrm{r}_{\mathrm{eq}}{ }^{2}}{2 \mathrm{r}^{2}}\left(1-3 \cos ^{2} \phi\right) \tag{7}
\end{equation*}
$$

where $\phi$ is the colatitude. The Keplerian portion of the energy remains constant. Thus, the variation in energy is only the third term of equation (7), or

$$
\Delta C=\left[\frac{-\mu J_{2}{ }^{r_{e q}^{2}}}{2 \mathrm{r}^{3}}\left(1-\cos ^{2} \phi\right)\right] \begin{align*}
& \mathrm{r}_{1}  \tag{8}\\
& \mathrm{r}_{0}
\end{align*}
$$

The angular momentum does vary and its derivative is given by,

$$
\frac{d \bar{h}}{d t}=\bar{r} \times \stackrel{\bar{r}}{ }
$$

However, since the potential has no longitude dependence, the $z$ - component of the derivative in equation (9) is zero. As seen from Figure 1, the z-component of the angular momentum is given by,

$$
h_{z}=h \cos i
$$

then

$$
\begin{equation*}
\frac{d \mathrm{z}}{\mathrm{dt}}=\frac{\mathrm{dh}}{\mathrm{dt}} \cos i-\mathrm{h} \sin i \frac{\mathrm{di}}{\mathrm{dt}}=0 \tag{10}
\end{equation*}
$$

Thus, the variation in the magnitude of the angular momentum over the same interval as $\Delta i$ is,

$$
\mathrm{h}=\mathrm{h} \tan \mathrm{i} \Delta \mathrm{i}
$$

The elements at the end of a step can now be obtained from,

$$
\begin{aligned}
& \mathrm{i}_{1}=\mathrm{i}+\Delta \mathrm{i} \\
& \Omega_{1}=\Omega+\Delta \Omega \\
& \omega_{1}=\omega+\Delta \omega \\
& \mathrm{P}_{1}=\frac{\mathrm{h}_{1}}{\mu} \\
& \mathrm{e}_{1}=\sqrt{1+\frac{2 \mathrm{P}_{1} \mathrm{C}_{1}}{\mu}}
\end{aligned}
$$

where
the subscript 1 denotes the values at the end of a step, and $C_{1}=C_{k}+\Delta C$
$C_{k}$ is the Keplerian energy
$\mathrm{h}_{1}=\mathrm{h}+\Delta \mathrm{h}$

## Description:

The variations in the orbital elements are determined in a straightforward manner from the above equations. The initial state is brought into the routine as position and velocity vectors in $x$ of STATE common.

The orbital elements are determined using subroutine ORBIT. Next, the true anomaly $f$, and $\omega+f$ are determined at the beginning and at the end of the step and stored in the arrays $F$ and THET, respectively. The variation in $\omega, i$ and $r$ is determined in the DO 2 loop, where the index refers to the limits of integration. The variation in the energy and angular momentum is calculated from equations (8) and (11). Finally, the orbital elements at the end of a step are determined from equation (12).

If the time at the end of the step is greater than the final time, logic flow transfers to statement 5 where an iteration scheme is employed to determine the radius corresponding to the final time.

The position and velocity vectors, mean, anomaly, and reciprocal of the mean motion are determined at the end of the step before the subroutine terminates.


## SUBROUTINE OBLE

## Calling Sequence:

Purpose:

Common Block Required:

Subroutines Required: M5OMDT, NUTATE, ROTATE

Input / Output

| 1/0 | SYMBOLIC NAME | DIMENSION | $\begin{gathered} \hline \text { COMMON } \\ \text { BLOCK } \\ \hline \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | GM | 12 | CONST | Gravitational potential |
| I | JC | 1. | CNTRI | Central Planet |
| I | POS | 3 | GRAVTY | Position vector |
| I | UJT | 1. | STATE(32) | Current julian date |
| 1/0 | XDUM | 3 | PERT | Acceleration Vector |

Usage:
JC must $=3$
Position Vector must be in GRAVTY (Mean Equator of 1950).

Acceleration Vector must be in PERT (Mean Equator of 1950).

## Description:

This subroutine first transforms the position vector of the $S / C$ to mean equator of date. It then calculates the acceleration acting on the $\mathrm{S} / \mathrm{C}$ due to the aspherical Earth with harmonics $A_{2}, A_{3}$ and $A_{4}$. The equations are as follows:

$$
\begin{aligned}
& \stackrel{\bullet}{\mathrm{x}}=\frac{3 \mathrm{x}}{2 \mathrm{r}^{5}} \mathrm{~A}_{2} \operatorname{GM}(3)\left(1-5 \mathrm{U}_{\mathrm{Z}}{ }^{2}\right)+\frac{5 \mathrm{xZ}}{2 \mathrm{r}^{7}} \mathrm{~A}_{3} \operatorname{GM}(3)\left(7 \mathrm{U}_{\mathrm{z}}{ }^{2}-3\right) \\
& +\frac{5 \mathrm{x}}{8 \mathrm{r}^{7}} \mathrm{~A}_{4} \operatorname{GM}(3)\left(3-42 \mathrm{U}_{\mathrm{z}}{ }^{2}+63 \mathrm{U}_{\mathrm{z}}{ }^{4}\right) \\
& \dot{y}=\frac{y^{*}}{x} \ddot{x} \\
& \underset{\mathrm{z}}{\mathrm{P}}=\frac{-3 \mathrm{z}}{2 \mathrm{r}^{5}} \mathrm{~A}_{2} \operatorname{GM}(3)\left(3-5 \mathrm{U}_{\mathrm{z}}{ }^{2}\right)+\frac{3}{2 \mathrm{r}^{5}} \mathrm{~A}_{3} \operatorname{GM}(3)\left(1-10 \mathrm{U}_{\mathrm{z}}^{2}+\frac{35}{3} U_{\mathrm{z}}^{4}\right) \\
& +\frac{5 \mathrm{z}}{8 \mathrm{r}^{7}} \mathrm{~A}_{4} \operatorname{GM}(3)\left(15-7 \mathrm{OU}_{\mathrm{z}}{ }^{2}+63 \mathrm{U}_{\mathrm{z}}{ }^{4}\right)
\end{aligned}
$$

where $\mathrm{U}_{\mathrm{z}}=\frac{\mathrm{z}}{\mathrm{r}}$

These accelerations are then transformed back into Earth mean equinox and equator of 1950, and then added to the total acceleration vector XDUM.

| Calling Sequence: | CALL OBLTY (EP, B) |
| :--- | :--- |
| Purpose: | OBLTY calculates the transformation matrix <br> that rotates from the Earth's ecliptic to the <br> Earth's equator. |
| Common Blocks Required: | CONST, INPUT |
| Subroutines Required: | None |

Input/Output

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DTMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| O | B | 3,3 | Calling Operand | Transformation <br> Matrix |
| I | EP | 1 | Calling Operand | Julian centuries <br> since 1900 |
| I | RAD | 1 | CONST (1) | Degrees per radian |

Theory:
The transformation from the Earth's ecliptic to the Earth's equator involves a simple rotation about the vernal equinox through the angle $\theta$,
where

$$
\theta=1.03 \times 10^{-8} \mathrm{t}^{3}-1.23 \times 10^{-7} \mathrm{t}^{2}-3.562 \times 10^{-3} \mathrm{t}+23.452294 \text { degrees }
$$

The time variable,t,represents the number of Julian centuries since 1900 (days since 1900.0 times $10^{-4}$ ).

The matrix is given by:

$$
\mathrm{B}=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos \theta & -\sin \theta \\
0 & \sin \theta & \cos \theta
\end{array}\right]
$$

## Description:

OBLTY calculates the transformation matrix for the rotation from the Earth's ecliptic to the Earth's equator. This matrix is loaded into B. The input EP represents the number of days since 1900.0 times $10^{-4}$.

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL OBSET (MODEL)
OBSET initializes the spherical harmonics coefficients for a given field model.

FIE LDM
None

Input/Output

| I/O | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :--- |
| 0 | MMOD | 1 | FIELDM(309) | Highest tesseral order |
| I | MODEL | 1 | Calling Operand | Model number |
| 0 | NMOD | 1 | FIELDM(308) | Highest zonal order |
| 0 | TSRL | 16,17 | FIELDM (17) | Tesseral array |
| 0 | ZONL | 16 | FIELDM (1) | Zonal vector |

Theory:
The relationships between zonals and tesserals and the sine and cosine coefficients of a spherical harmonic potential field are as follows:

$$
\begin{array}{ll}
\operatorname{ZONL}(i)=-C_{i o} & i=1,2, \ldots \ldots \ldots \\
\operatorname{TSRL}(i, j)=C_{i j} & 0 . j \quad i, i=1,2, \ldots \ldots \ldots \\
\operatorname{TSRL}(j, i+1)=S_{i j} & o \quad j \quad i=1,2, \ldots \ldots
\end{array}
$$

## Description:

OBSET initializes the spherical harmonic coefficients according to the field model selected. The values NMOD and MMOD are also set to indicate the size of the selected model. NMOD indicates the highest-ordered zonal and MMOD indicates the highest tesseral order. The following chart indicates the possible values of MODEL and the corresponding fields:

|  |  |  |  |
| :---: | :---: | :---: | :---: |
| MODEL | FIELD DESCRIPTION | NMOD | MMOD |
| 1 | Houston L1 model of <br> lunar field | 3 | 3 |
| 3 | JPL 15-8 model of the <br> lunar field | 15 | 8 |
| 4 | Mars field--post <br> Mariner 1971 | 4 | 3 |
| 5 | Clears coefficient arrays <br> for loading of input field | 0 | 0 |

See Figure 1 for the val ues of the $C$ and $S$ terms for the first three models above.

HOUSTRE LI MODEL OF LTMAR FTRID, WOR $=3$, MPOD $=$ ?

| $C(2, J)=-0.2071 D-03$ | 0.0 | $0.2972 D-04$ |  |  |
| :--- | :--- | :--- | :--- | :--- |
| $C(3, J)=0.2100 D-04$ | $0.3400 D-04$ | 0.0 | $0.2583 D-05$ |  |
| $S(2, J)=0.0$ | 0.0 | 0.0 |  |  |
| $S(3, J)=$ | 0.0 | 0.0 | 0.0 | 0.0 |

JPL 15-8 MODEL OF LUNAR FIELD, NMOD $=15$, MMOD $=8$

| $C(2, J)=$ | -0.1996D-03 | $0.81710-05$ | $0.2359 \mathrm{D}-04$ |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $C(3, J)=$ | -0.58730-05 | $0.3001 \mathrm{D}-31$ | 0. $0.598 \mathrm{D}-05$ | 0.18470-05 |  |
| $C(4, J)=$ | $0.11950-04$ | -0.2226D-05 | -0.2418D-05 | $0.23050-05$ | -0.4547)-06 |
| $C(5, J)=$ | $\begin{aligned} & -0.45440-05 \\ & -0.30410-09 \end{aligned}$ | -0.2455D-05 | $0.8880 \mathrm{D}-06$ | -0.7774D-06 | $0.7097 \mathrm{D}-06$ |
| $c(6, J)=$ | $\begin{array}{r} 0.70830-05 \\ -0.75645-0 ? \end{array}$ | $\begin{aligned} & -0.63730-05 \\ & -0.7277 \mathrm{D}-08 \end{aligned}$ | -0.69170-06 | -0.20410-06 | 0.45200-07 |
| $C(7, J)=$ | 0. $37790-04$ | $0.13210-05$ | -0. 12930-06 | -0.13050-06 | -0.35?30-08 |
|  | $0.60610-08$ | -0.29378-09 | $0.24540-10$ |  |  |
| $C(8, J)=$ | -0.59675-05 | -0.8040-05 | 0.25500-06 | - $0.67710-07$ | $0.29350-07$ |
|  | 0.4844D-09 | $0.1252 \mathrm{n}-08$ | -0.1523n-10 | -0.3179D-11 |  |
| $C(3, J)=$ | -0.3206D-05 | 0.0 | 0.0 | 0.9 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(10, J)=$ | 0. $2.367 \mathrm{D}-05$ | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(11, J)=$ | -0.7311-05 | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(12, J)=$ | $0.12510-04$ | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(1.3, \mathrm{~J})=$ | -0.3315D-04 | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(14, J)=$ | $0.20440-04$ | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $C(15, J)=$ | -0.2977D-04 | 0.0 | 0.0 | 0.0 | 0.0 |
|  | 0.0 | 0.0 | 0.0 | 0.0 |  |
| $S(2, J)=$ | 0.0 | -0.72135-05 | $0.4538 \mathrm{D}-05$ |  |  |
| $S(3, J)=$ | 0.0 | 0.14210-05 | $0.57480-06$ | -0.2919D-05 |  |
| $S(A, J)=$ | 0.0 | $0.32990-05$ | -0.2389D-0.5 | $-0.6222 n-06$ | $0.42480-06$ |
| $S(5, J)=$ | $\begin{aligned} & 0.0 \\ & -0.3012 \mathrm{D}-07 \end{aligned}$ | -0.692.5D-05 | $-0.3178 \mathrm{D}-06$ | 0.11590-05 | $0.6296 \mathrm{D}-07$ |
| $S(6, J)=$ | $\begin{aligned} & 0.0 \\ & -0.1654 \mathrm{D}-07 \end{aligned}$ | $\begin{aligned} & 0.5342 n-05 \\ & 0.23735-08 \end{aligned}$ | -0.17570-95 | -0.3484D-06 | -0.98700-07 |
| $S(7, J)=$ | 0.0 | -0.1579D-05 | 0.17520-06 | 0.2.440D-06 | $0.12950-07$ |
|  | $0.5582 \mathrm{D}-08$ | $0.12270-08$ | -0. $01310-10$ |  |  |
| $S(8, J)=$ | 0.0 | 0.620 80-05 | -0.17670-05 | -0.1437n-06 | -0.7757D-07 |
|  | 0.2391D-08 | -0.2.390D-09 | $-0.5813 D-10$ | $0.77985-1.1$ |  |

NARS FIELD-POST MARINER 1971 , NMON $=4, ~ M T O D=3$

| $C(2, J)=-0.1950 D-02$ | 0.0 | $-0.5400 D-04$ |  |
| :--- | :--- | :--- | :--- | :--- |
| $C(3, J)=-0.3050 D-04$ | $0.2770 D-05$ | 0.0 | 0.0 |
| $C(4, J)=-0.3140 D-04$ | 0.0 | 0.0 | 0.0 |
| $S(2, J)=0.0$ | 0.0 | $0.2900 D-04$ |  |
| $S(3, J)=0.0$ | $0.2500 D-04$ | 0.0 | 0.0 |



Calling Sequence:

Purpose:

Common Blocks:Required:
Subroutines Required:

CALL ORBIT (K, J, X, XD, GM, ELM)
ORBIT calculates orbital elements from position and velocity,or position and velocity from orbital elements.

CONST

None

Input / Output

| SYMBOLIC |  | COMMON |  |  |
| :---: | :---: | :---: | :--- | :--- |
| I/O | NAME | DIMENSION | BLOCK |  |$\quad$ DEFINITION

## Theory:

Consider first the calculation of the orbital elements $P$, e, f, $\omega, i, \Omega$ from the position and velocity vectors $\overline{\mathrm{X}}$ and $\overline{\mathrm{V}}$. The angular momentum vector is given by

$$
\overline{\mathrm{H}}=\overline{\mathrm{X}} \times \overline{\mathrm{V}}=(\mathrm{h} \sin (\mathrm{i}) \sin (\Omega),-h \sin (\mathrm{i}) \cos (\Omega), h \cos (\mathrm{i}))
$$

so that $|\vec{H}|=|\vec{X} \times \vec{V}|=h$

$$
\text { and } P=h^{2} / G M
$$

The equation of motion of a body rotating about a central body is given by

$$
R=P /(1+e \cos f) \text { where } R \text { is the distance to the body. }
$$

Thus $R=|\overrightarrow{\mathrm{x}}|$ and $\mathrm{e} \cos (\mathrm{f})=(\mathrm{P}-\mathrm{R}) / R$,
which implies $-e \sin (f) \dot{f}=-\left(P / R^{2}\right) \dot{R}$

$$
\text { or e } \sin (f)=\frac{P}{h} \quad\left(\frac{X_{1} V_{1}+x_{2} V_{2}+x_{3} V_{3}}{R}\right) \text { since } h=R^{2} f^{f}
$$

From $e \cos$ (f) and e $\sin$ (f), both e and $f$ can be calculated, as long as e is not equal to zero, which is explained later on.
The calculation of $i$ and $\Omega$ is straightforward given the components of $\vec{H}$, as long as the inclination is not $0^{\circ}$ or $180^{\circ}$, in which case $\Omega$ is set to zero.

In order to find $u^{\prime}$, let $u=f+\omega$. Then the geometry of the orbit can be used to yield the following equations:

$$
\begin{aligned}
& R \cos (u)=X_{1} \cos (\Omega)+X_{2} \sin (\Omega) \\
& R \sin (u)=\left(-X_{1} \sin (\Omega)+X_{2} \cos (\Omega)\right) \cos (i)+X_{3} \sin (i)
\end{aligned}
$$

Now the calculation of $u$ is straightforward, and

$$
\mu=u-f, \text { unless e equals zero, }
$$

In whioh case $\omega$ is set to zero and $f=u$.

The second part of the subroutine is the reverse case, i.e., given orbit elements find $\bar{X}$ and $\bar{V}$. The first step in this process is to calculate the sines and cosines of $\omega$, $i$, and $\Omega$ (if $K=5$ this is input, also if $K=6 \mathrm{e}, \omega$, and f must be calculated as well as the above sines and cosines). After this, a transformation is set up which represents rotations through $\omega$, $i$, and $\Omega$ to transform orbit coordinates to whatever coordinate system you may be in. The rotations and the resulting matrix are as shown on the following page.

$$
\begin{aligned}
\left(\begin{array}{l}
x \\
y \\
z
\end{array}\right) & =\left(\begin{array}{ccc}
\cos \Omega & -\sin \Omega & 0 \\
\sin \Omega & \cos \Omega & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos i & -\sin 1 \\
0 & \sin i & \cos i
\end{array}\right)\left(\begin{array}{ccc}
\cos \omega & -\sin \omega & 0 \\
\sin \omega & \cos \omega & 0 \\
0 & 0 & 1
\end{array}\right)\left(\begin{array}{l}
X^{\prime} \\
Y^{\prime} \\
0
\end{array}\right) \\
& =\left(\begin{array}{cc}
\cos (\omega) \cos (\Omega)-\cos (i) \sin (\Omega) \sin (\omega) & -\sin (\omega) \cos (\Omega)-\cos (i) \sin (\Omega) \cos (\omega) \\
\cos (\omega) \sin (\Omega)+\cos (i) \cos (\Omega) \sin (\omega) & -\sin (\omega) \sin (\Omega)+\cos (i) \cos (\Omega) \cos (\omega) \\
\sin (i) \sin (\omega) & \sin (i) \cos (\omega)
\end{array}\right) / X
\end{aligned}
$$

Here $x^{\prime}$ and $y^{\prime}$ refer to a coordinate system in the orbit plane with the $x$-axis pointing toward pericenter. Thus finding position and velocity vectors is reduced to finding the vectors in the orbit plane and then transforming them.

For position, R can be found by

$$
R=\frac{P}{1+e \cos (f)}
$$

and then

$$
\begin{aligned}
& x^{\prime}=R \cos (f) \\
& y^{\prime}=R \sin (f)
\end{aligned}
$$

For the velocity, the magnitude can be found by the energy relationship

$$
\begin{aligned}
& E=-\frac{\mu}{2 a}=\frac{V^{2}}{2}-\frac{\mu}{R} \\
& \text { or } V=\sqrt{\frac{2 \mu}{R}-\frac{\left(1-e^{2}\right)}{P} \mu}
\end{aligned}
$$

thus $\mathrm{V}_{1}^{\prime}=-\mathrm{V} \sin (\mathrm{f}-\gamma)$ and $\mathrm{V}_{2}^{\prime}=\mathrm{V} \cos (\mathrm{f}-\gamma)$
where $\gamma$ is the flight path angle and is given by

$$
\gamma=\arctan (e \sin (f) /(1+e \cos (f)))
$$

Once $V_{1}^{\prime}$ and $V_{2}^{\prime}$ are calculated, the velocity vector may be found by transforming the primed system as was done before.

## Description:

ORBIT provides the interchange between position and velocity coordinates and orbital elements. If the direction flag, $J$, is greater than or equal to zero, orbital elements will be calculated from position and velocity. Note that both position and velocity vectors are required for this transformation. Three different sets of orbital elements may be calculated, depending on the condition flag, $K$. If $K$ is equal to three, only the standard orbital elements will be output, i.e., the semi-latus rectum ( P ), eccentricity ( e ), true anomaly $(\mathrm{f})$, angle of pericenter ( $\omega$ ), inclination (i), and angle of ascending node ( $\Omega$ ) will be output in $\operatorname{ELM}(1)$ - $\operatorname{ELM}(6)$. If $K$ equals five, the following values will be output in $\operatorname{ELM}(7)-\operatorname{ELM}(12)$ in addition to the previous values: $\cos (\omega), \sin (\omega), \cos (i)$, $\sin (i), \cos (\Omega)$, and $\sin (\Omega)$. If $K$ equals six, the following values will be output in $\operatorname{ELM}(1) \longrightarrow \operatorname{ELM}(6): P, e \cos (\omega)$, e $\sin (\omega), \omega+f, i, \Omega$. No other values will be output.

If $J$ is negative on input, then position and/or velocity will be output as follows: $J$ equal to minus one implies that only the position vector is desired. Other negative values indicate that both the position vector and the velocity vector are desired.

The value of K in this case indicates the type of orbital elements that were input. K equal to one implies that only the standard orbital elements were input. K equal to five implies that the sines and cosines of $\omega$, i, and $\Omega$ were input in addition to the standard elements. K equal to six implies that e $\cos (\omega)$, e $\sin (\omega)$, and $\omega+\mathrm{f}$ were input in place of $e, f$, and $\omega$.


## SUBROUTINE ORIENT

Calling Sequence:
Purpose:

CALL ORIENT (A, U, S, CD, SC, K)
To find the angle through which one vector must be rotated about a second vector in order that the first vector should form a given angle with a third vector.

Common Blocks Required: None

Subroutines Required: CROSS

Input / Output

|  SYMBOLIC  COMMON <br> I/O NAME DIMENSION BLOCK |  |  |  | DE FINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | A | 3 | ARGUMENT LIST | Rotated vecto |
| I | U | 3 | ARGUMENT LIST | Rotation vector. |
| I | S | 3 | ARGUMENT LIST | Fixed reference vector. |
| I | CD | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \end{aligned}$ | Cosine of desired angle. |
| 0 | SC | 4 | ARGUMENT LIST | Sines and cosines of the solutions |
| 0 | K | 1 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \end{aligned}$ | Indicator for solution type |

## Description:

A principal use of this subroutine is to locate the times of day (right ascension of the Earth-fixed orbital plane's ascending node) when the touch-down point for Earth returns lies in the return orbit plane. This is accomplished by asking the subroutine for sines and cosines of the incremental right ascension required for the dot product of the normal to the return orbit plane and the touch-down radius vector to be zero. This subroutine can also solve the slightly more general problem of achieving a desired non-right-angle between two vectors by rotating one of them about a given axis.

Let $T_{u}(A)$ be the $3 \times 3$ orthogonal matrix which, when applied to any non-zero vector not parallel to $u$, rotates that vector about $u$ through an angle, $\theta$, measured in the plane normal to $u$. Thus, $\overline{\mathrm{a}}=\mathrm{T}_{\mathrm{u}}(\theta)$ a is a vector which sweeps out a cone about $u$ as $\theta$ increases from 0 degrees to 360 degrees. The problem solved by this subroutine is then: Find $A$ (or $\sin A$ and $\cos \theta$ ) such that

Now

$$
\begin{aligned}
\bar{a} \cdot s & =|\bar{a}| \mid\{\cos \varphi \\
\bar{a} \cdot s & =\bar{a}^{T} s=\bar{a}^{T} T_{u}(-9) s \\
& =a^{T}\left(\cos \theta I+(1-\cos \theta) \frac{u u^{T}}{u^{T} u}-\sin \theta \frac{u x}{|u|}\right)_{s} \\
& =\cos \theta a^{T} s+(1-\cos \theta) \frac{a^{T} u u^{T} s}{u^{T} u}-\frac{a^{T} u x s}{|u|} \sin \theta
\end{aligned}
$$

The equation to be solved then becomes

$$
|a||s| \cos \varphi=a \cos \theta+\beta \sin \theta+\gamma
$$

where $\quad y=\frac{\left(\mathrm{a}^{\mathrm{T}} u^{\mathrm{T}} \mathrm{u}^{\mathrm{T}} \mathrm{s}\right)}{u^{T} u}$

$$
B=\frac{(\mathrm{sxu})^{\mathrm{T}} \mathrm{a}}{|\mathrm{u}|}
$$

and

$$
\alpha=\mathrm{a}^{\mathrm{T}} \mathrm{~s}-\gamma
$$

or

$$
\bar{\alpha} \cos \theta+\bar{a} \sin \theta+\bar{y}=0
$$

where

$$
\begin{aligned}
& \bar{\gamma}=\frac{\gamma}{|a||s|}-\cos \varphi \\
& \vec{\beta}=\frac{B}{|\mathrm{a}||\mathrm{s}|} \\
& \bar{\alpha}=\frac{\alpha}{|\mathrm{a}||\mathrm{s}|}
\end{aligned}
$$

The solutions of this equation for $\sin \theta$ amd $\cos \theta$ are given by

$$
\left[\begin{array}{c}
\sin \theta \\
\cos \theta
\end{array}\right]=\frac{1}{\overline{\mathrm{a}}^{2}+\bar{\beta}^{2}}\left[\begin{array}{cc}
\vec{\beta} & \bar{\alpha} \\
\bar{\alpha} & -\beta
\end{array}\right]\left[\begin{array}{c}
-\bar{\gamma} \\
\sqrt{\bar{\alpha}^{2}+\bar{\beta}^{2}-\bar{\gamma}^{2}}
\end{array}\right]
$$

It can be seen that two solutions exist - one for each sign on $\sqrt{\bar{\alpha}^{2}+\bar{\beta}^{2}-\bar{\gamma}^{2}}$. When $\bar{\alpha}^{2}+\bar{\beta}^{2}=\bar{\gamma}^{2}$, however, only one solution exists. This case will occur when, for the use mentioned, the latitude of touch-down is exactly equal to the inclination of the return orbit. If the latitude is greater than the inclination $\bar{\alpha}^{2}+\bar{\beta}^{2}<\bar{\gamma}^{2}$ and a real solution is impossible. The subroutine then sets the indicator, $K$, to a positive value and provides the solutions for $\theta$ when the angle between a and $s$ is a maximum or minimum.

Calling Sequence:
Purpose:

Common blocks required:

Subroutines required:

CALL OUTPUT ( $\mathrm{T}, \mathrm{X}$ )
This routine outputs the spacecraft's state with respect to the launch and/or target planets in the desired coordinate system.

CNTRL, CONST, INPUT, INTVAR, PLNET, STATE

M50LEQ, M5OMDT, M5OMDT, OBLTY, ORBIT, ROTATE, SHORB2, TRMN.

Input / Output

| I/ $\varnothing$ | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | PROGRAM DIMENSION | $\begin{gathered} \hline \text { COMMON } \\ \text { BLOCK } \end{gathered}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| I | DJO | 1 | INPUT (46) | Modified julian date of state epoch |
| I | GM | 12 | CONST (5) | Gravitational constants |
| I | JC | 1 | CNTRL (7) | Central planet number |
| I | JL | 1 | INPUT | Launch planet number |
| I | JT | 1 | (1031) | Target planet number |
| I | KOUTPT | 2 | $\begin{aligned} & \text { INPUT } \\ & (1039) \\ & \hline \end{aligned}$ | Launch and target planet output coordinate systems |
| I | KOUT9 | 1 | $\begin{aligned} & \text { INPUT } \\ & (1058) \\ & \hline \end{aligned}$ | Auxiliary output flag |
| I | KP | 12 | TNPUT | Planets in system |
| I | KTERM | 1 | $\begin{aligned} & \text { INPUT } \\ & (1059) \end{aligned}$ | Type of auxiliary output |
| I | METH | 1 | $\begin{aligned} & \text { INPUT } \\ & (1013) \\ & \hline \end{aligned}$ | Trajectory propagation method |
| I | PCON | 1 | INPUT (05) | Position units conversion factor |
| I | RAD | 1 | CONST (1) | Radians to degrees conversion factos |
| I | T | 1. | $\begin{aligned} & \text { calling } \\ & \text { argung } \end{aligned}$ | Time since DJO |
| I | VCON | 1 | INPUT (6) | Velocity units conversion factor |
| I | X | 6 | Calling argument | Position and velocity vectors of state to be output |

## Description:

This subroutine outputs the state position and velocity vectors and orbital elements with respect to the launch and target planets in a desired coordinate system. The state and time are input via the argument list. The state is translated to the launch and target planets and rotated to the desired coordinate system before output. The output coordinate system is determined by the KOUTPT flag as follows:

| KOUTPT | $=1$ |  | Mean equinox and ecliptic of date |
| ---: | :--- | ---: | :--- |
|  | $=2$ |  | True equator and prime meridian |
|  | $=3$ |  | Mean equator and equinox of 1950 |
|  | $=4$ |  | True equator and equinox of date |
|  | $=5$ |  | no output |

The subroutine also has provisions for an auxiliary abbreviated output written on the unit number designated by KOUT9. If this flag is non-zero, output is presented on unit KOUT9 in a form specified by KTERM as follows:

| KTERM | $=1$ |  | orbital elements |
| ---: | :--- | ---: | :--- |
|  | $=2$ |  | position and velocity vectors |
|  | $=3$ |  | 1 and 2 |

The coordinate system of the output is also specified by KOUTPT. This subroutine also has a provision to determine the time in the shadow cone during one orbit assuming constant elements. The output is presented in the auxiliary abbreviated output. Thus, the KOUT9 flag must be non-zero. Also, the KSHORB flag must be 1 before subroutine SHORB2 is called to perform the shadow calculations.

SUBROUTINE OUTPUT


Rotate state to desired output
coordinate system
coordinate system
Calculate position, velocity, elements


Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL OUT1
OUT1 determines the print interval and controls the interpolation logic for printing.

INPUT, INTVAR, SAVE, STATE
INTERP, OUTPUT

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| I | DJO | 1 | INPUT(46) | Modified Julian date of <br> launch epoch |
| I | DTOUT | 10 | INPUT(460) | Printing intervals |
| I | T | 1 | STATE(10) | Seconds since state epoch |
| I | TOUT | 10 | INPUT(450) | Switching times of <br> printing intervals |
| I | TOUTL | 1 | SAVE(40) | Time of last print |
| I | X | 6 | STATE(1) | Current state |
| I | UJT | 1 | STATE(32) | Current modified Julian date |

## Description:

This subroutine finds the print times and uses subroutine INTERP and OUTPUT to output the state at the print times. OUT1 is called from TIMEC at the beginning of each numerical integration step. The only print times determined on each call to OUT1 are print times which lie in the last numerical integration step.

The print interval is determined from the print table consisting of TOUT and DTOUT. The print time, TDUM, is determined from $\operatorname{TDUM}=$ TOUTL + DTOUT (IDUM)
where TOUTL is the last print time and DTOUT(IDUM) is the current print interval determined from the print table.

If this time is greater than the current time, no printing is done. If not, the state is determined at the print time using the interpolation logic from subroutine INTERP and the state is then output in subroutine OUTPUT. If the print time is within 10 seconds of the current time, the state is not interpolated for and OUTPUT is called directly.

The next print time is then determined from equation (1) and the whole process is repeated until the print time is greater than the current time. The subroutine terminates when this criterion is satisfied.

## SUBROUTINE PLANET

Calling Sequence:
Purpose:

CALL PLANET
Subroutine PLANET supervises the calculation of the planet's position and velocity vectors.

CETBL2, CETBL3, CNTRL, INPUT, INTVAR, STATE

LUNA, SOL, READE

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I | EJO | 1 | STATE(26) | Ephemeris date of launch epoch |
| I | EJT | 1 | STATE(28) | Current ephemeris date |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | JMN | 1 | INPUT(1017) | Ephemeris type flag |
| I | KP | 12 | INPUT(1001) | Planets in system <br> I <br> I KREAD |
| I | TAB3 | 1 | CNTRL(8) | Flag to determine if tape (or disk <br> read necessary |

Description:
This subroutine controls the calculation of the planet's position and velocity vectors. The JMN flag determines the type of ephemeris used as follows:

$$
\begin{aligned}
& J M N= \text { 1. Mean elements } \\
& \text { 2. Mean elements for Sun and mean elements plus first- } \\
& \text { order corrections for Moon }
\end{aligned}
$$

3. Ephemeris tape read
4. Mean elements for Sun, osculating elem ents for Moon
5. Ephemeris disk using Goddard's direct read feature

If JMN is not equal to 3 or 5 , subroutines SOL and LUNA are called to determine the planets' positions if the respective planets are in the system. If JMN is equal to 3 or 5 , subroutine READE is used to obtain the position and velocity of the planets. Before this routine is called, the IREQ array is set to the KP array and CENT set to the central planet number. These are arrays used in READE. The KREAD flag is also set. Each read of the ephemeris disk on tape brings eight days of data into arrays in READE. The KREAD flag is set to zero (no read) if the time of the ephemeris call is within the eight day range of data already started in READE, otherwise, the KREAD flag is set to one for a tape or disk read.

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required:

## CALL POST

POST computes certain parameters for output during midcourse guidance analysis.

CONST, INPUT, INTVAR, MCCOM, PLNET, STATE

BELL, BURND, DOT, MVTRN, ORBIT, PLANET, RETRO, SUNMIN, TRMN, VISIB, VNORM

Input / Output

| 1/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | WO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | DJO | 1 | INPUT(46) | Julian date of anchor vector epoch (days) |
| I | ASPMC | 1 | INPUT(441) | Specific impulse of the midcourse motor (sec) |
| I | WRETRO | 1 | INPUT(443) | Weight of retro fuel ( kg ) |
| I | AFUEL | 1 | INPUT(471) | Attitude fuel/angle factor ( $\mathrm{kg} / \mathrm{rad}$ ) |
| I | WDROP | 1 | INPUT(473) | Retro drop-weight (kg) |
| I | JT | 1 | INPUT(1031) | Target body number |
| I | MCOUT | 1 | INPUT(1050) | Extra output key |
| I | KOUT9 | 1 | INPUT(1058) | Logical unit for scope output. |
| I | MCUNIT | 1 | INPUT(1061) | Logical unit for writing out information. |
| I | IPROB | 1 | INPUT(1070) | Output probability scale (pct) |
| I | NORMIN | 1 | INPUT(1080) | Retro optimization key |
| I | XMC | 6 | $\mathrm{MCCOM}(6)$ | Pre-ignition midcourse state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$ ) |
| I | DV | 3 | MCCOM(12) | Midcourse velocity impulse (km/sec |
| I | DVMG | 1 | MCCOM(15) | Midcourse impulse magnitude( $\mathrm{km} / \mathrm{sec}$ ) |


| $\begin{array}{cc}  & \text { SYMBOLIC } \\ \text { I/O } & \text { NAME } \end{array}$ |  | COMMON |  | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | DEC | 1 | $\mathrm{MCCOM}(16)$ | Declination of DV (deg) |
| I | RA | 1 | MCCOM 17$)$ | Right ascension of DV (deg) |
| I | TMCS | 1 | MCCOM(18) | Midcourse ignition time (sec past anchor epoch) |
| I | XSUN | 3 | MCCOM $(21)$ | Spacecrafí-to-Sun unit vector <br> Mean of 1950 |
| I | VRET | 1 | MCCOM(25) | Retro delta-v (km/sec) |
| I | EXFUEL | 1 | MCCOM(26) | Expected second-midcourse fuel (kg) |
| I | BT | 3 | $\mathrm{MCCOM}(27)$ | Spin-axis unit vector at retro |
| I | DVT | 1 | $\operatorname{MCCOM}(30)$ | Trim velocity ( $\mathrm{km} / \mathrm{sec}$ ) |
| I | FUELT | 1 | $\operatorname{MCCOM}(31)$ | Trim fuel ( kg ) |
| I | XR | 6 | $\mathrm{MCCOM}(32)$ | Pericynthion state <br> ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$, Lunar Equator) |
| I | DJDIF | 1 | MCCOM(39) | Anchor epoch-launch epoch difference (days) |
| I | SIGOUT | 6 | MCCOM 40$)$ | Expected end constraint errors from BELL |
| I | FFIRE | 1 | $\mathrm{MCCOM}(46)$ | Firing true anomaly for retro (rad) |
| I | WTF | 1 | $\mathrm{MCCOM}(47)$ | Spacecraft weight after midcourse (kg) $\qquad$ |
| I | EXVZ | 1 | MCCOM $(49)$ | Expected second midcourse velocity ( $\mathrm{m} / \mathrm{sec}$ ) |
| I | IT | 1 | MCCOM(157) | Midcourse iteration counter from MDCORS |
| I | KMC | 1 | MCCOM(159) | Midcourse execution counter trom PROTO |
| I | KT | 1 | MCCOM $(161)$ | Flight time counter from PROTO |
| I | ICB | 1 | $\operatorname{MCCOM}(165)$ | Midcourse central body number |
| 0 | KEL | 10 | MCCOM(172) | Elevation indicator array for midcourse. |
| 0 | KELR | 10 | MCCOM(182) | Elevation indicator array for retro |
| I | X | 6 | STATE (1) | $\begin{aligned} & \text { State at target closest approach } \\ & (\mathrm{km}, \mathrm{km} / \mathrm{sec})\end{aligned}$ |
| I | T | 1 | STATE(10) | Time at target closest approach (sec past anchor epoch) |
| I | ATT | 3 | STATE(11) | Spin-axis attitude before midcourse (unit vector) |

## Description:

POST is a subroutine in which auxiliary output calculations are performed for midcourse guidance subroutines PROTO and FIXATG. These calculations (in order of appearance) are:

1. Midcourse fuel ( kg ) *
2. Spin-axis-Sun angle at midcourse (deg)
3. Retro velocity impulse ( $\mathrm{km} / \mathrm{sec}$ )
4. Attitude fuel to midcourse attitude ( kg )
5. Minimum spin-axis-Sun angle to midcourse (deg)
6. Array of midcourse visibility for trackers (KEL $=\mathrm{EL} / 10+1$ )
7. Burn time ( sec )
8. Trim fuel (kg) *
9. Time of retro ignition (hours past launch) *
10. Right ascension of the spin axis at retro (deg) *
11. Declination of the spin axis at retro (deg) *
12. Total correction fuel ( kg$)^{*}$
13. Spin-axis-Sun angle at retro (deg)*
14. Trim velocity ( $\mathrm{m} / \mathrm{sec}$ )*
15. Minimum spin-axis-Sun angle to retro attitude (deg)*
16. Array of retro visibility for trackers $\begin{array}{rlrl}\mathrm{KELR} & =\mathrm{EL} / 10+1 & \mathrm{EL} \geq 0 \\ & =0 & & \mathrm{EL}<0\end{array}$
( * computation depends on NORMIN- if $\leq 0$, retro attitude, trim fuel and firing time are as determined in TARGET, otherwise retro attitude and firing time are optimized in RETRO to minimize trim fuel)

The computed quantities are written on logical unit MCUNIT for later retrieval by PROTO or MCVERF. The logic of this subroutine is straightforward and merits no flowchart.

## SUBROUTINE PRINT

Calling Sequence:
Purpose:

## CALL PRINT

The purpose of this routine is to print the contents of various common blocks.

Common Blocks Required: CNTRL, CONST, INPUT, PERT, PLNET, STATE, SAVE

Subroutines Required:

ORBIT

Input / Output

| I/O | SYM BOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| O | ELM | 6 | STATE(14) | Spacecraft's osculating <br> orbital elements |
| I | GM | 12 | CONST(5) | Gravitational constants |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | METH | 1 | INPUT(1013) | Trajectory propagation <br> indicator. |
| I | X | 6 | STATE(1) | Spacecraft's position and <br> velocity vectors |

## Description:

This subroutine prints the contents of various common blocks on output unit 6 . A series of flags are used to determine which common blocks are to be printed. If the input location in the table below is set to one, the designated common block will be printed.


1020
1021
1022
1023
1024
1025
1026

COMMON
BLOCK
STATE
CNTRL
PLNET
INPUT
NOT USED
PERT
SAVE

The orbital elements of the spacecraft are determined from subroutine ORBIT if STATE common is to be output and the trajectory propagation method used is Cowell's.

## SUBROUTINE PROTO

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:

CALL PROTO
PROTO controls the gross midcourse analysis logic and printout

ANKOR, CNTRL, CONST, INPUT, MCCOM, STATE

FIXATG, MDCORS, POST

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK |
| :---: | :---: | :---: | :--- |
| I | DJL | 1 | INPUT(37) | Julian date of launch (days)


| I/O | SYMBOLIC <br> name | DIMENSION COMMON |  | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | MCOUT | 1 | INPUT(1050) | Extra midcourse output key |
| I | JMC | 1 | INPUT(1051) | Number of midcourse executions desired |
| I | MCUNIT | 1 | INPUT(1061) | Logical unit for midcourse output |
| I | IPROB | 1 | INPUT(1070) | Output-scaling probability (\%) |
| I | IBURN | 1 | INPUT(1071) | Midcourse burn model key |
| I | KTF | 1 | INPUT(1077) | Scan selector <br> (-fixed attitude, + flight time) |
| 0 | XMC | 6 | MCCOM $(6)$ | Pre-ignition midcourse state (km, km/sec) |
| 0 | DV | 3 | MCCOM(12) | Mıdcourse velocity impulse(km/sed |
| 0 | DVMG | 1 | MCCOM $(15)$ | Midcourse impulse magnitude ( $\mathrm{m} / \mathrm{sec}$ ) |
| 0 | DEC | 1 | $\operatorname{MCCOM}(16)$ | Declination of DV (deg) |
| 0 | RA | 1 | MCCOM (17) | Right ascension of DV (deg) |
| 0 | TMCS | 1 | MCCOM $(18)$ | Midcourse execution time (sec past $\mathrm{a}_{0} \mathrm{e}_{\mathrm{o}}$ ) |
| O | DVB4 | 1 | MCCOM(24) | Expended midcourse velocity ( $\mathrm{km} / \mathrm{sec}$ ) |
| O | EXFUEL | 1 | MCCOM(26) | Expected second-midcourse fuel (kg) |
| 0 | DVT | 1 | $\operatorname{MCCOM}(30)$ | Trim velocity ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | FUELT | 1 | $\operatorname{MCCOM}(31)$ | Trim fuel (kg) |
| 0 | DJDIF | 1 | MCCOM $(39)$ | Anchor-launch epoch difference (sec) |
| 0 | SIGOUT | 6 | $\mathrm{MCCOM}(40)$ | Expected end constraint errors |
| O | EXV2 | 1 | MCCOM(49) | Expected second-midcourse velocity ( $\mathrm{m} / \mathrm{sec}$ ) |
| 0 | PSID | 10 | $\operatorname{MCCOM}(80)$ | Desired constraint vector |
| I/O | PSI | 10 | $\operatorname{MCCOM}(100)$ | Constraint error vector |
| O | KBURN | 1 | $\operatorname{MCCOM}(154)$ | Midcourse burn-type key |
| 0 | KENTRY | 1 | MCCOM(156) | MDCORS re-entry key to avoid initialization |
| 0 | IT | 1 | $\mathrm{MCCOM}(157)$ | MDCORS trial counter |


| SYMBOLIC <br> I/O |  |  | NAME | DIMENSION |
| :--- | :--- | :--- | :--- | :--- |
| 0 | KMC | 1 | BLOCK | DE FINITION |
| 0 | KT | 1 | MCCOM(159) | Midcourse execution time counter |
| 0 | KGLAWI | 1 | MCCOM(161) | Flight time counter |
| 0 | MCCOM(164) | Guidance law indicator |  |  |
| 0 | KEL | 10 | MCCOM(172) | Tracking visibility indicators <br> for midcourse |
| 0 | KELR | 10 | MCCOM(182) | Tracking visibility indicators <br> for retro |

## Description:

PROTO operates in one of 3 modes, depending upon the setting of KTF。

KTF $=0 \quad$ (one-dimensional scan of midcourse execution times)
In this mode, sample midcourse correction maneuvers are calculated at JMC execution times which begin at TMC and proceed at steps of DELTMC. Each maneuver, found by MDCORS and elaborated by POST, corresponds to a particular guidance law and the same set of desired end conditions. Cnaracteristics of the maneuvers and arrival conditions are written on MCUNIT in POST, then read again for printout in PROTO at the conclusion of the scanning process.
$\underline{K T F}>0$ (Two-dimensional scan of midcourse times and flight times)
Tnis mode injects a scan of flight times (using the FTA guidance law) into the execution time scan. It enables generation of all those midcourse solutions which arrive at a specific closest approach distance and inclination, thereby encompassing all the available guidance law uptions. At each midcourse execution time, flight times are scanned in KTF one-hour steps beginning at DTFIN. Ac each point of the scan, MDCORS and POST compute the maneuver and arrival characteristics and write them on MCUNIT for later printout by PROTO. By scanning flight times within the execution-time loop, gradients for targeting may be salvaged for use at several flight times and some trajectory computation may be avoided. Gradients are re-generated every fourth hour of the flight time scan.
$\underline{K T F}<0$ (Two-dimensional scan of midcourse times and delta $v$ ) Tnis is the fixed-attitude-guidance mode. At each midcourse execution time, FIXATG
is called. In FIXATG, it is assumed that the spin-axis direction or thrust direction is fixed at RAI and DECI. The impulsive velocity magnitude is varied systematically in FIXATG from zero in -KTF steps of size DINK (location INPUT(479) in $\mathrm{km} / \mathrm{sec}$ ). The post-ignition trajectory is propagated to target closest approach wnere arrival characteristics are computed and printed out. MDCORS is not called in this mode, neither is POST and no summary printout is performed in PROTO.


Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL PUTELS (ICOM, MSGERR, \$)
This subroutine writes a file describing the state for retrieval by the GTDS program.

ELMNT
None

Input / Output

|  | SYMBOLIC |  |  |  |
| :--- | :---: | :---: | :---: | :--- |
| I/O | NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| I | ICOM | 1 | Calling <br> Operand | Initialization flag <br> 1 <br> Initialize output data file <br> 0 add new element to set |
| I | MSGERR | 1 | Calling <br> Operand | Error message return number |
| $O$ | $\$$ | - | - | Statement number for <br> error return |

## Description:

This subroutine is used to output the state for retrieval by the GIDS program. The ICOM flag is used to initialize the output file. When this flag is set to 1 , the output file is initialized. No writes are made with this flag setting. When the ICOM flag is zero, writes are made on unit 27.

The data to be written on unit 27 is transferred into the subroutine via the SET array of ELMNT common. The set array is defined as follows:

| Location | Definition |
| :---: | :---: |
| 1 | Date of state in year, month and day written as YYMMDD. |
| 2 | Time of state in hours, minutes and seconds written as HHMMSS. SSS. |
| 3-5 | Cartesian position vector. |
| 6-8 | Cartesian velocity vector |
| 9-14 | Keplerian orbital elements. The order is (SMA, ECC, INCLIN, LAN, PA, MA) |
| 15-35 | Upper triangle of the state covariance matrix |

36

44

Start time of fitted data (year, month, day)
Start time of fitted data (hour, minute, second)
End of fitted data (year, month, day)
End of fitted data (hour, minute, second)
Root mean square of fit.
Satellite identification number
Reference coordinate system of state 1 for mean of 1950

Central body indicator
1 Earth 2 Moon 3 Sun
Element set number

## SUBROUTINE QUARTC

## Calling Sequence:

Purpose:

Common Blocks Used:
Subroutines Required:

CALL QUARTC (C, X, N)
To find the real roots of a quartic * equation,

$$
P=c_{1}+\sum_{i=1}^{N} c_{i+1} x^{i}=0
$$

None
None

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | DEFINITION |
| :---: | :---: | :---: | :---: |
| I | C | 5 | Coefficients of the polynomial |
| O | X | 4 | Solution vector if real solutions exist <br> (Upper-loaded if less than four exist) |
| $\mathrm{I} / \mathrm{O}$ | N | 1 | Input: Order of the polynomial $(3$ or 4$)$ <br> Output: Real solutions found $(0,1,2,3,4)$ |

* QUARTC may also be used to find the real roots of a cubic equation. The mathematical description which follows applies only to quartic equations.


## Description:

The Descartes technique is used to solve the equation,
$A X^{4}+B X^{3}+C X^{2}+D X+E=0$. First divide through by $A(A \neq 0)$ to obtain

$$
\begin{equation*}
X^{4}+B^{\prime} x^{3}+C^{\prime} X^{2}+D^{\prime} X+E^{\prime}=0 \tag{1}
\end{equation*}
$$

Substituting $X=y+h$, where $h=\frac{B^{\prime}}{4}$, into (1) obtain

$$
\begin{equation*}
y^{4}+P y^{2}+Q y+R=0 \tag{2}
\end{equation*}
$$

where

$$
\begin{align*}
& P=6 h^{2}+3 B^{\prime} h+C^{\prime}  \tag{3}\\
& Q=4 h^{3}+3 B^{\prime} h^{2}+2 C^{\prime} h+D^{\prime}  \tag{4}\\
& R=h^{4}+B^{\prime} h^{3}+C^{\prime} h^{2}+D^{\prime} h+E^{\prime} \tag{5}
\end{align*}
$$

Now (2) may be factored to produce equation (6)

$$
\begin{equation*}
\left(\mathrm{y}^{2}+\sqrt{\mathrm{R}^{\prime}} \mathrm{y}+\varepsilon\right)\left(\mathrm{y}^{2}-\sqrt{\mathrm{R}^{\prime}}+\beta\right)=0 \tag{6}
\end{equation*}
$$

which, in turn, yields four quadratic solutions if $\mathrm{R}^{\prime}, \xi$ and $\beta$ are defined.

$$
\begin{align*}
& \xi=\frac{1}{2}\left(P+R^{\prime}-\frac{Q}{\sqrt{R^{\prime}}}\right)  \tag{i}\\
& \beta=\frac{1}{2}\left(P+R^{\prime}+\frac{Q}{\sqrt{R^{\prime}}}\right) \tag{8}
\end{align*}
$$

$R^{\prime}$ is the maximum of the real roots of

$$
\begin{equation*}
z^{3}+a Z+b=0 \tag{Э}
\end{equation*}
$$

minus $\frac{2}{3} P$. The coefficients, $a$ and $b$, in (9) are given by

$$
\begin{equation*}
a=\frac{1}{3}\left[3\left(P^{2}-4 R\right)-4 P^{2}\right] \tag{10}
\end{equation*}
$$

and

$$
\begin{equation*}
b=\frac{1}{27}\left[16 P^{3}-18 P\left(P^{2}-4 R\right)-27 Q^{2}\right] \tag{11}
\end{equation*}
$$

To find the roots of (9), first compute the quantity

$$
\begin{equation*}
\Delta=\frac{\mathrm{a}^{3}}{27}+\frac{\mathrm{b}^{2}}{4} . \tag{12}
\end{equation*}
$$

If $\Delta>0$, the only real root is

$$
\begin{equation*}
\mathrm{z}_{1}=\sqrt[3]{-\frac{\mathrm{b}}{2}+\sqrt{\Delta}}+3 \sqrt{-\frac{\mathrm{b}}{2}-\sqrt{\Delta}} . \tag{13}
\end{equation*}
$$

If $\Lambda=0$, the real roots are

$$
\begin{equation*}
\mathrm{Z}_{1}=2 \sqrt[3]{-\frac{b}{2}}, \quad \mathrm{Z}_{2}=\sqrt[3]{\frac{b}{2}} \quad \text { and } \mathrm{Z}_{3}=\sqrt[3]{\frac{b}{2}} \tag{14}
\end{equation*}
$$

If $\Delta<0$ the roots are

$$
\begin{align*}
& z_{1}=2 \sqrt{-\frac{a}{3}} \cos (\phi / 3) \\
& z_{2}=2 \sqrt{-\frac{a}{3}} \cos \left(\phi / 3+120^{\circ}\right), \tag{15}
\end{align*}
$$

and

$$
z_{3}=2 \sqrt{-\frac{a}{3}} \cos \left(\phi / 3+240^{\circ}\right)
$$

where

$$
\begin{equation*}
\cos \phi=-\frac{b}{2 \sqrt{-\frac{a^{3}}{27}}} \tag{16}
\end{equation*}
$$

After computing $R^{\prime}$, equations (6), (7), and (8) yield $y_{i}, i=1,4$. Then $X_{i}=y_{i}+h$ are solutions to the quartic. Only real solutions are considered by this subroutine.

## SUBROUTINE QUIKIE

Calling Sequence: $\quad$ CALL QUIKIE (A, SI, CI, $\varnothing 0, S U N \varnothing, T F 1, T F 2$ )
Purpose: This subroutine calculates approximate shadow times for circular lunar orbits with constant inclinations. The times calculated are shadow-free time and time until next shadow (or time until no shadow, if orbit is presently in shadow)

Common Blocks Required: CONST
Subroutines Required: None
Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | A | 1 | $\begin{aligned} & \text { CALLING } \\ & \text { OPERAND } \end{aligned}$ | Semi-major axis |
| I | SI | 1 | CALLING <br> OPERAND | Sine (inclination) |
| I | CI | 1 | $\begin{aligned} & \text { CALLING } \\ & \text { OPERAND } \end{aligned}$ | Cosine (inclination) |
| I | $\varnothing_{0}$ | 1 | CALLING OPERAND | Initial longitude of node |
| I | SUN $\varnothing$ | 1 | $\begin{aligned} & \text { CALLING } \\ & \text { OPERAND } \\ & \hline \end{aligned}$ | Initial longitude of Sun |
| I | GM | 12 | CONST | Planet gravitational constants |
| I | PI, PI2 | 1,1 | CONST | $\pi, 2 \pi$ |
| O | TF1 | 1 | CALLING OPERAND | Time until next shadow or neg. time until no shadow |
| 0 | TF2 | 1 | CALLING OPERAND | Shadow-free time |

## Description:

This subroutine calculates the times when a given orbit around the Moon will cross the Moon's shadow. The orbit is assumed to have zero eccentricity and constant inclination. Input quantities should be expressed in an ecliptic coordinate system; however, the Moons equator is probably acceptable for most work.

The output quantity TF2 represents the time between shadows, (i.e., the maximum shadow free time.

Note: For low inclination orbits, part of the orbit is always in shadow. For this case, both TF1 and TF2 are output as 999.9.

Theory:
In an ecliptic coordinate system, let wee the longitude of the Sun, $\Omega$ be the longitude of the orbit's ascending node, and $\nu$ the angle between them.
Then $\gamma=\Psi-\Omega$, and $\dot{\gamma}=\dot{\Psi}-\stackrel{\circ}{\Omega}$, where $\dot{\Psi}$ is the angular velocity of the Sun about the Earth (. $998 \mathrm{deg} /$ day $)$, and $\dot{\Omega}$ is the precession rate of the orbit around the planet. $\left(\dot{\Omega} \approx 1.5 \mathrm{nc}_{2} \cos (\mathrm{i}) \mathrm{RM}^{2} / \mathrm{A}^{2}+.75 \mathrm{n}^{2}{ }^{2} \mu / \mathrm{n} \cos (\mathrm{i})\right)$
$\mathrm{n}=$ mean motion of spacecraft
$c_{2}=$ lunar oblateness coefficient
$\mathbf{i}=$ inclination
RM = Radius of the Moon

Since the orbit is circular, the radius A is constant. Thus, the angle from the shadow cone centerline to the edge of the cone at a distance $A$ from the planet is $\delta_{c}=\sin ^{-1}(\mathrm{RM} / \mathrm{A})-\alpha$, where $\alpha$ is the half shadow cone angle.

$\gamma_{c}=\sin ^{-1}\left(\frac{\sin \delta \mathrm{c}}{\sin \mathrm{i}}\right) \quad$ (Note: $|\operatorname{sm} \delta \mathrm{c}| \leq|\operatorname{smi}|$ )

This implies that the orbit is a shadow whenever

$$
-\gamma \leq \gamma \leq \gamma_{\mathrm{c}} \text { or } \pi-\gamma_{\mathrm{c}} \leq \gamma \leq \pi+\gamma_{\mathrm{c}}
$$

Thus the problem is reduced to finding out what quadrant $y$ is in and what the angle is until the orbit status changes. For instance, if $\boldsymbol{\gamma}$ is in the third quadrant and the orbit is not in shadow, then TF1 $=\frac{2 \pi-\nu \mathrm{c}}{\dot{\gamma}}$.
Similarly, if $\gamma$ is in the second quadrant and the orbit is in shadow, then
TF1 $=-\left(\pi+\gamma_{c}-\gamma\right) / \dot{\gamma} \cdot \mathrm{TF} 2$ is independent on $\gamma$ and is equal to $\frac{\left(\pi-2 \gamma_{c}\right)}{\dot{\gamma}}$.


## SUBROUTINE RANDM1

Calling Sequence:
Purpose:

CALL RANDM1 (IY, YFL)
This subroutine determines a random number uniformly distributed between 0 and 1 .

None
Subroutines Required: None

Input / Output

| I/O | SYMBOLIC <br> NANE | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| $I$ | $I Y$ | 1 | CALLING <br> ARGUMENT | Random number generator |
|  |  | 1 | CALLING <br> ARGUMENT | Random number |

## Description:

The random number is determined from

$$
\mathrm{YFL}=\mathrm{IY}(1027)(.465661 \mathrm{D}-9) .
$$

The random number generator is updated on each call as,

$$
I Y=I Y(1027)
$$

This random number generator is designed for the word length of an IBM 360 series computer.

Calling Sequence:
Purpose:

Common Blocks Required:

Subroutines Required:

CALL READE (JED, TSEC, TERR)
This routine takes the data from the ephemeris tape and interpolates for the planet's position and/or velocity at the desired time. It also transfers to the central planet.

CETBL2, CETBL3, CETBL9, PLNET

GETTAP

Input / Output

| I/O | SYMBOLIC NAME | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \\ & \hline \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ICENT | 1 | CETBL2(2) | Central Planet number |
| I | IERR | 1 | CALLING OPERAND | Error flag |
| I | IREQ | 13 | CETBL2 (3) | Planet's ephemeris flag $\qquad$ |
| I | JED | 1 | CALLING OPERAND | Reference julian ephemeris |
| 0 | NUT | 4 | PLNET(85) | Nutation output |
| I | NUTAT | 204 | CETBL3(830) | Nutation input data from ephemeris tape |
| 0 | TABOUT | 6,12 | PLNET(1) | Position and velocity vectors of planets with respect to the central planet |
| I | TAB3 | 829 | CETBL3(1) | Input planet position and velocity vectors from ephemeris tape |
| I | TSEC | 1 | $\begin{aligned} & \text { CALLING } \\ & \text { OPERAND } \end{aligned}$ | Seconds of ephemeris time past JED |

## Description:

This subroutine determines the positions and velocities of the planets with respect to the central planet from the JPL ephemeris tape. The times of the positions and velocities to be obtained are brought through the argument list as JED and TSEC. The output positions and velocities are put in TABOUT of PLNET common.

The positions and velocities of the planets are set up in the TAB3 array through a call to GETTAP. This routine searches for the proper block of data from the tape (or disk) and reads into TAB3 the eight-day block of data that encompasses the desired time. READE next interpolates for the positions and velocities at the desired time from the eight-day block of data in TAB3. After the interpolation is complete, READE translates the positions and velocities to the desired central planet.

## FUNCTION RETDV

Calling Sequence:
$\mathrm{DV}=\mathrm{RETDV}(\mathrm{DVMCM}, \mathrm{WT}, \mathrm{WTF}, \mathrm{WTAR})$

Purpose:
RETDV computes the retro velocity as a function of midcourse velocity.

CめNST, INPUT

Subroutines Called:
None

Input/Output

| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :--- | :---: | :---: | :--- |
| I | DVMCM | 1 | Argument <br> List | Midcourse velocity magnitude (km/sec) |
| I | WT $\varnothing$ | 1 | Argument <br> List | Spacecraft weight before midcourse (kg) |
| $\varnothing$ | WTF | 1 | Argument <br> List | Spacecraft weight after midcourse (kg) |
| $\varnothing$ | WTAR | 1 | Argument <br> List | Spacecraft weight after retro (kg) |
| $\varnothing$ | RETDV | 1 | Function | Retro velocity (km/sec) |
| I | G | 1 | CONST <br> $(45)$ | Earth's surface gravity (km/sec ${ }^{2}$ ) |
| I | ASPMC | 1 | INPUT <br> $(441)$ | Specific impulse of midcourse motor (sec) |
| I | ASPR | 1 | INPUT <br> $(442)$ | Specific impulse of retro motor (sec) |
| I | WRETR $\varnothing$ | 1 | INPUT <br> $(443)$ | Weight of retro fuel (kg) |

## Method:

The rocket equation is applied to simulate both the midcourse and retro motor burns. Attitude fuel expenditure is ignored.

$$
\begin{aligned}
\text { WTF } & =\mathrm{WT} \varnothing \cdot \mathrm{e}^{-[D V M C M /(G \cdot A S P M C)]} \\
\text { WTAR } & =\mathrm{WTF}-\mathrm{WRETR} \varnothing \\
\text { RETDV } & =-(\mathrm{G} \cdot \text { ASPR }) \ln \left(\frac{\text { WTAR }}{\text { WTF }}\right)
\end{aligned}
$$

Calling Sequence:

Purpose:

CALL RETRO (ELM, DVR, DVT, FFIRE, RAO, DECO, X)

RETRO finds the retro attitude which results in the minimum trim velocity.

Common Blocks Required: INPUT
Subroutines Required: MINV

## Input/Output

| I/O | $\begin{aligned} & \hline \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSTON | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ELM | 12 | ARGUMENT LIST | Orbital elements of the approach hyperbola |
| I | DVR | 1 | ARGUMENT LIST | Retro velocity impulse ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | DVT | 1 | ARGUMENT LIST | Trim velocity required (km/sec) |
| O | FFIRE | 1 | ARGUMENT LIST | Firing true anomaly on the approac hyperbola (rad) |
| I | RAO | 1 | ARGUMENT LIST | Initial estimate of retro right ascension (rad) |
| I | DECO | 1 | ARGUMENT LIST | Initial estimate of retro declination (rad) |
| O | X | 2 | $\begin{aligned} & \text { ARGUMENT } \\ & \text { LIST } \\ & \hline \end{aligned}$ | Right ascension(rad) \& declination of the solution retro orientation. |
| I | KROUT | 1 | INPUT(1072) | Extra output key (print if KROUT = 1) |

## Method:

A steepest descent hunting procedure is used to find the retro orientation resulting in the minimum trim velocity. The control vector, X , is defined by

$$
X=\left[\begin{array}{l}
\text { right ascension of the spin-axis } \\
\text { declination of the spin-axis }
\end{array}\right]
$$

(The velocity impulse imparted by the retro burn is anti-parallel to spin-axis). The trim velocity, $v_{t}$, is defined (through formulae in MINV and TRIM) as a function of $X$ and the firing true anomaly, $A$, on the approach hyperbola.

$$
v_{t}=f(X, \theta)
$$

RETRO obtains $v_{t}^{*}=f\left(X, \theta^{*}\right)$ by calling MINV. $\theta^{*}$ is the value of $\theta$ which minimizes $v_{t}$ for a given $X$ and is, generally, also an implicit function of $X . v_{t}^{*}$ is the $\theta$-minimized trim velocity. The objective of the logic in RETRO is to find $X^{*}$, the control (specifying spin-axis in orientation) for which $v_{t}$ is minimized.

$$
\min v_{t}=f\left(X^{*}, \theta^{*}\right)
$$

We will now denote $v_{t}^{*}$ by the typographically-simpler $\mathrm{v}(\mathrm{X})$. Approximations to the partial derivatives,

$$
P(X)=\frac{\partial v}{\partial X}
$$

are obtained by the secant method evaluated at $X$. The first order change, $\delta \mathrm{v}$, in $v(X)$ due to a change, $\delta X$, in $X$ is

$$
\delta v=P_{\delta} X
$$

which we cannot solve directly for $\delta \mathrm{X}$ given $\delta \mathrm{v}$. We choose, instead, to minimize a function, $\mathrm{S}^{2}$, of $\delta \mathrm{X}$ subject to the constraint $\mathrm{P} \delta \mathrm{X}=\delta \mathrm{v}$. Let $\mathrm{S}^{2}=\delta \mathrm{X}^{\mathrm{t}} \mathrm{G} \delta \mathrm{X}$, where $G$ is a positive - definite weighting matrix. We adjoin the constraint to $S^{2}$ and differentiate with respect to $\delta \mathrm{X}$ to solve for $\delta \mathrm{X}$ as a function of the undetermined multiplier, $\lambda$.

$$
\begin{gathered}
\frac{\partial}{\partial \delta \mathrm{X}}\left[\delta \mathrm{X}^{t} \mathrm{G} \delta \mathrm{X}-2 \lambda(\mathrm{P} \delta \mathrm{X}-\delta \mathrm{v})\right]=2 \delta \mathrm{X}^{t} \mathrm{G}-2 \lambda \mathrm{P}=0 \\
\delta \mathrm{X}=\mathrm{G}^{-t} \mathrm{P}^{\mathrm{t}} \lambda \\
\mathrm{P}_{\delta} \mathrm{X}=\mathrm{PG}^{-t} \mathrm{P}^{\mathrm{t}} \lambda=\delta \mathrm{v} \\
\lambda=\frac{\delta \mathrm{v}}{\mathrm{PG}^{-t} \mathrm{P}^{t}} \\
\delta \mathrm{X}=\frac{\mathrm{G}^{-t} \mathrm{P}^{t} \delta \mathrm{~V}}{\mathrm{PG}^{-t} \mathrm{P}^{t}}
\end{gathered}
$$

This is the steepest descent formulation for $\delta \mathrm{X}$. RETRO's weighting matrix is defined by

$$
\mathrm{G}^{-\mathrm{t}}=\left[\begin{array}{cc}
\mathrm{x}_{1}^{2} & 0 \\
0 & x_{2}^{2}
\end{array}\right]
$$

The hunting procedure used in RETRO to find $X^{*}$ is to ask for an improvement, $\delta v$, (negative) at the $k$-th iteration, where $X$ is $X_{r}$. This predicts a control change $\delta \mathrm{X}$ which leads to $\mathrm{X}_{\mathrm{r}+1^{\circ}}$

$$
X_{r+1}=X_{r}+\delta X
$$

If

$$
\Delta v=v\left(X_{r+1}\right)-v\left(X_{r}\right)
$$

is positive, $\delta v$ is halved and $\delta \mathrm{X}, \mathrm{X}_{\mathrm{r}+1}$ and $\mathrm{v}\left(\mathrm{X}_{\mathrm{r}+1}\right)$ are re-calculated (without recalculating $P$ ) until $\Delta v$ is negative. If the magnitude of $\delta v$ is less than the specified tolerance, $\mathrm{X}^{*}=\min \left(\mathrm{X}_{\mathrm{r}}, \mathrm{X}_{\mathrm{r}+1}\right)$ and the iteration is complete.

If $\Delta \mathrm{v}<\delta \mathrm{v}$ on the first trial at any $\mathrm{X}_{\mathrm{r}}$, $\delta \mathrm{v}$ is set to $1.5 \delta \mathrm{v}$ for the next step from $\mathrm{X}_{\mathrm{r}+1}$. If $\Delta v$ is negative on the first trial but $\Delta v>\delta \mathrm{v}$, another step is computed using $\delta v$ but starting at $X_{r+1}$. If $\Delta v_{2}=v\left(X_{r+2}\right)-v\left(X_{r}\right)$ is less than $\delta v$, the iteration proceeds from $\mathrm{X}_{\mathrm{r}+2}$. If $\delta \mathrm{v}<\Delta \mathrm{v}_{2}<0$, then $\delta \mathrm{v}$ is set to $\Delta \mathrm{v}_{2}$ - tolerance. This process is considered convergent when the magnitude of $\delta v$ is less than the pre-set tolerance.


## SUBROUTINE RKSEVN

Calling Sequence:
Purpose:

CALL RKSEVN (N, HO, XO)
This subroutine integrates a set of simultaneous differential equations using a seventh-order ten-cycle Runge-Kutta scheme.

Common Blocks Required: CNTRL, INPUT, INTVAR
Subrountines Called:
EQNS

Input / Output

| I/ $\varnothing$ | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | $\begin{aligned} & \text { PROGRAM } \\ & \text { DIMENSION } \end{aligned}$ | $\begin{aligned} & \hline \text { COMMON } \\ & \text { BLOCK } \\ & \hline \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ERRC | 1 | INPUT(1) | Error control |
| I/ $\varnothing$ | H | 1 | INTVAR(14) | Initial compute interval and next compute interval |
| $\emptyset$ | HO | 1 | Calling Operand | Actual compute interval used |
| $\emptyset$ | KHALT | 1 | CNTRL(6) | Error stop flag |
| I | N | 1 | Calling Operand | Number of differential equations |
| I/ $\varnothing$ | RATES | 6 | INTVAR(8) | First derivative of dependent variables. |
| $\emptyset$ | X | 1 | INTVAR(1) | Independent variable |
| I | XO | 1 | Calling Operand | Initial independent variable |
| I/ $\varnothing$ | Y | 6 | INTVAR(2) | Dependent variables |

## Theory:

The integration scheme shown below was developed by D. Sarafyan, Reference 1. In his scheme, a set of simultaneous differential equations are numerically integrated using a seventh-order ten-cycle Runge-Kutta method. In the following equations $f(x, y)$ refers to the value of the derivatives of the function at the independent variable X and dependent variable Y . This value is brought into the subroutine via RATES in INTVAR common. It is calculated in Subroutine EQNS.

The value of the independent variables, $Y$, at the end of the step is obtained from:

$$
\begin{equation*}
Y\left(X_{0}+h\right)=Y_{0}+\frac{1}{840}\left(41\left(K_{0}+K_{9}\right)+216\left(K_{4}+K_{8}\right)+27\left(K_{5}+K_{7}\right)+272 K_{6}\right) \tag{1}
\end{equation*}
$$

where

$$
\begin{align*}
\mathrm{K}_{0}=\mathrm{hf}\left(\mathrm{X}_{0}, \mathrm{Y}_{0}\right) \\
\mathrm{K}_{1}=\mathrm{hf}\left(\mathrm{X}_{0}+\frac{1}{3} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{3} \mathrm{~K}_{0}\right) \\
\mathrm{K}_{2}=\mathrm{hf}\left(\mathrm{X}_{0}+\frac{1}{2} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{8}\left[\mathrm{~K}_{0}+3 \mathrm{~K}_{1}\right]\right) \\
\mathrm{K}_{3}=\mathrm{hf}\left(\mathrm{X}_{0}+\mathrm{h}, \mathrm{Y}_{0}+\frac{1}{2}\left[\mathrm{~K}_{0}-3 \mathrm{~K}_{1}+4 \mathrm{~K}_{2}\right]\right) \\
\mathrm{K}_{4}=\operatorname{hf}\left(\mathrm{X}_{0}+\frac{1}{6} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{648}\left[83 \mathrm{~K}_{0}+32 \mathrm{~K}_{2}-7 \mathrm{~K}_{3}\right]\right) \\
\mathrm{K}_{5}=\mathrm{hf}\left(\mathrm{X}_{0}+\frac{1}{3} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{54}\left[-3 \mathrm{~K}_{0}-4 \mathrm{~K}_{2}+\mathrm{K}_{3}+24 \mathrm{~K}_{4}\right]\right)  \tag{2}\\
\mathrm{K}_{6}=\operatorname{hf}\left(\mathrm{X}_{0}+\frac{1}{2} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{5088}\left[-290 \mathrm{~K}_{0}-524 \mathrm{~K}_{2}+145 \mathrm{~K}_{3}\right.\right.
\end{align*} \quad \begin{array}{r}
\left.\left.\quad+1908 \mathrm{~K}_{4}+1305 \mathrm{~K}_{5}\right]\right) \\
\mathrm{K}_{7}=\mathrm{hf}\left(\mathrm{X}_{0}+\frac{2}{3} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{1431}\left[292 \mathrm{~K}_{0}+108 \mathrm{~K}_{2}+13 \mathrm{~K}_{3}-318 \mathrm{~K}_{4}\right.\right.
\end{array}
$$

$$
\begin{aligned}
& \mathrm{K}_{8}=\mathrm{hf}\left(\mathrm{X}_{0}+\frac{5}{6} \mathrm{~h}, \mathrm{Y}_{0}+\frac{1}{68688}\left[14042 \mathrm{~K}_{0}+11012 \mathrm{~K}_{2}-4477 \mathrm{~K}_{3}\right.\right. \\
&\left.\left.+5724 \mathrm{~K}_{4}-6903 \mathrm{~K}_{5}+6360 \mathrm{~K}_{6}+31482 \mathrm{~K}_{7}\right]\right) \\
& \mathrm{K}_{9}=\operatorname{hf}\left(\mathrm{X}_{0}+\mathrm{h}, \mathrm{Y}_{0}+\frac{1}{4346}\left[-2049 \mathrm{~K}_{0}-1836 \mathrm{~K}_{2}+839 \mathrm{~K}_{3}\right.\right. \\
&\left.\left.+5724 \mathrm{~K}_{4}-4692 \mathrm{~K}_{5}+12084 \mathrm{~K}_{6}-9540 \mathrm{~K}_{7}+3816 \mathrm{~K}_{8}\right]\right)
\end{aligned}
$$

## Description :

The value of $\mathbf{f}$ in the above equations is determined by repeated calls to EQNS with the indicated independent variable set in X of INTVAR common and the dependent variables set in $Y$ of INTVAR common. EQNS calculates the derivatives of the dependent variables and puts the answer in RATES of INTVAR common. The values of $f$ are used to calculate $K_{0}$ through $K_{9}$. These $K$ 's are used to determine variables at the end of the computing step from the first equation. If a fixed computing interval is used ( $E R R C=0$ ), the subroutine terminates.

When the automatic compute interval option is used (ERRC $\neq 0$ ) the subroutine calculates the fourth-order solution at the end of the step from

$$
\begin{equation*}
Y_{4}\left(X_{0} \text { th }\right)=Y_{0}+\frac{1}{6}\left(K_{0}+4 K_{2}+K_{3}\right) \tag{3}
\end{equation*}
$$

where

$$
\mathrm{K}_{0}, \mathrm{~K}_{2}, \mathrm{~K}_{3} \text { are the same as defined above. }
$$

Next the fourth-order solution is compared to the seventh-order solution. The largest relative difference between the two is used to determine the compute interval from

$$
\begin{equation*}
\mathrm{H}=\mathrm{H}(\mathrm{ERRC} / \mathrm{ERRELS})^{1 / 5} \tag{4}
\end{equation*}
$$

where ERRELS is the largest relative difference. H is the compute interval to be used in the next step; however, if the relative error is more than 4 ERRC, then the error in the current solution is considered too large. In this case, the whole process is repeated again with the compute interval just determined. A limit of 10 repeats is allowed. If the limit is reached, the error halt flag is set and the
subroutine terminates. The independent variable is not incremented inside this routine. It should be done after the call to RKSEVN as shown below.

$$
\begin{equation*}
T=T+H O \tag{5}
\end{equation*}
$$

where $T$ is the independent variable.

## Reference

1. Sarofyan, D.; "Seventh-Order Ten-Stage Runge-Kutta Formulas," Technical Report No. 38, Louisiana State University, Department of Mathematics, January 1970.


| Calling | nce: | CALL RO | (X, Y, S, C, U, V) |
| :---: | :---: | :---: | :---: |
| Purpo |  | To rotate subroutin and $S=\sin$ through | thonormal vectors in th putes $\mathrm{U}=\mathrm{CX}+\mathrm{SY}, \mathrm{V}=-\mathrm{SX}$ and $V$ are obtained by rota le $A$ in the sense $X$ into |
| Comm | ocks used | None |  |
| Subrou | Required | None |  |
| Input |  |  |  |
| I/O | $\overline{\mathrm{MBOLIC}}$ <br> NAME | DIMENSION | DEFINITION |
| I | X | 3 | Orthonormal input vect |
|  | Y | 3 | Orthonormal input vect |
| I | S | 1 | Sine of rotation angle |
| I | C | 1 | Cosine of rotation angle |
| 0 | U | 3 | Rotated vectors output |
| 0 | V | 3 | Rotated vectors output |

ROTAIT is coded in such a way that $U$ and $V$ may share the same storage as $X$ and $Y$.

## SUBROUTINE ROTATE

| Calling Sequence: | CALL ROTATE (M, A, B, C) |
| :--- | :--- |
|  | CALL MVTRN $(A, B, C, M, N)$ |
| Purpose: | To form the matrix product $C=A B$ or $C=A^{T} B$ |
|  | where $A$ is a $3 \times 3$ matrix and $B$ and $C$ are |
|  | $3 \times 1$ matrices $(3 \times N$ in MVTRN $)$. |
| Common Blocks Required: | None |
| Subroutines Called: | None |
| Input/Output |  |


| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :--- | :--- |
| I | A | 9 | Call List | Matrix multiplier |
| I | B | $3, N$ | Call List | Matrix multiplier |
| $\varnothing$ | C | $3, \mathrm{~N}$ | Call List | Product matrix |
| I | M | 1 | Call List | Indicator: $\mathrm{C}=\mathrm{AB}$ if $\mathrm{M}=1$ |
| $\mathrm{C}=\mathrm{A}^{\mathrm{B}}$ otherwise |  |  |  |  |
| I | N | 1 | Call List | Number of columns of B and C |

## SUBROUTINE SADOUT

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

CALL SADOUT
SADOUT outputs the times of umbral, penumbral, and occultation times.

INPUT, SHAD, STATE
CALEND

Input / Output

| SYMBOLIC |  |  |  | COMMON |  |  |
| :--- | :--- | :---: | :---: | :--- | :---: | :---: |
| I/O | NAME | DIMENSION | BLOCK |  |  |  |
| I | DJO | 1 | DEFINITION |  |  |  |
| I | DJ1 | 1 | INPUT(46) | Modified julian date of state epoch |  |  |
| I | JL | 1 | INPUT(37) | Modified julian date of liftoff |  |  |
| I | JT | 1 | INPUT(1015) | Launch planet number |  |  |
| I | KPLOT | 1 | INPUT(1031) | Target planet number |  |  |
| I | T | 1 | STATE(10) | Current time |  |  |
| I | TSX | 10 | SHAD(19) | Table of umbral, penumbral, and occultation <br> entrance and exit times |  |  |

## Description:

The shadow times to be output are stored in the TSX array. A test is made on the smallest time in the TSX array. If this time is greater than the current time, no output is presented. The TSX array is next ordered so that it is monotonically increasing. The KK array is included to keep track of the times in the ordered TSX array. Subroutine CALEND is used to determine the calendar date corresponding to the times in the TSX array. The dates are output on unit 6 if the time is less than T. If the KPLOT flag is set to 20 , information is written on unit 20 for later use in a plotting program.

## SUBROUTINE SENSO

## Calling Sequence:

Purpose:

## CALL SENS $\varnothing$

SENS $\varnothing$ computes the gradient of end constraints with respect to midcourse velocity. It can also be used to provide end state and constraint errors as a function of midcourse velocity.

Common Blocks Required: INPUT, MCC $\varnothing \mathrm{M}$
Subroutines Called: MCBURN, FめWARD, TARGET

Input/Output

| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DESCRIPTION |
| :--- | :--- | :---: | :---: | :--- |
| I | DINK | 1 | INPUT <br> $(479)$ | Midcourse velocity impulse for partials <br> (km/sec) |
| $\varnothing$ | KRASH | 1 | INPUT <br> $(1032)$ | Trajectory stop key for F $\varnothing$ WARD |
| I | DV | 3 | MCC $\varnothing \mathrm{M}$ <br> $(12)$ | Midcourse correction impulse (km/sec) |
| $\varnothing$ | DPT | 3,10 | MCC $\varnothing \mathrm{M}$ <br> $(50)$ | Gradient ( $\% \psi / \partial \Delta V$ ) |
| I/ $\varnothing$ | PSI | 10 | MCC $\varnothing \mathrm{M}$ <br> $(100)$ | End constraint error vector (see TARGET) |
| I/ $\varnothing$ | IR | 1 | MCC $\varnothing \mathrm{M}$ <br> $(158)$ | Gradient-or-not logic key ( $\geq 2$ for gradient) |
| $\varnothing$ | ISP | 1 | MCC $\varnothing \mathrm{M}$ <br> $(166)$ | Gradient-was-generated key (set 1$)$ |

## Method:

The secant method of computing approximate partial derivatives is used in generating the gradient of constraint errors with respect to control variations. The control vector, DV, is the impulsive midcourse correction velocity vector. This vector is fed to MCBURN, which outputs the post-burn state. This state is fed to FØWARD, which outputs the state at target closest approach. TARGET takes this end state and generates the constraint error vector, PSI. The transpose of the gradient is stored in DPT.

$$
\mathrm{DPT}=\left(\frac{\partial \mathrm{PSI}}{\partial \mathrm{DV}}\right)^{\mathrm{T}}
$$



## SUBROUTINE SETUP2

Calling Sequence: CALL SETUP2
\(\left.$$
\begin{array}{ll}\text { Purpose: } & \begin{array}{l}\text { This subroutine initializes flags and constants } \\
\text { before any of the program options are initiated. }\end{array} \\
\text { Common Blocks Required: }\end{array}
$$ \quad \begin{array}{l}ANKOR, CETBL2, CETBL3, CNTRL, CONST, <br>
ELMNT, INPUT, INTVAR, MCCOM, MOON, OBSIT, <br>
Subroutines Required: <br>
<br>

PERT, PLNET, SAVE, STATE\end{array}\right\}\)| CLOSE, DATE, DVMAG, FIND, M50EPM, M50LEQ, |
| :--- |
| M50MDT, MVTRN, NUTAIT, NUTATE, OBLTY, OBSET, |
|  |
| ORBIT, PLANET, TRMN |


| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| 0 | ATT | 3 | STATE(11) | Attitude unit vector in mean of 1950 . |
| I | AU | 1 | CONST(4) | Astronomical unit |
| 0 | COV | 3,3 | INPUT(56) | State estimation covariance matrix. |
| I | DAYL | 1 | INPUT(21) | Launch day |
| I | DAYO | 1 | INPUT(51) | State epoch day |
| I | DEC | 1 | INPUT(48) | Initial declination of attitude |
| 0 | DJL | 1 | INPUT (37) | Modified julian <br> launch date |
| 0 | DJO | 1 | INPUT(46) | Modified julian date of state epoch. |
| 0 | DX | 3 | STATE (4) | Velocity |
| 0 | EJO | 1 | STATE(26) | Ephemeris date of state epoch |
| O | ELM | 6 | STATE(14) | Orbital eloments of initial state |
| 1/O | ETC | 1 | INPUT (39) | Ephemeris time correction |
| I | GM | 12 | CONST(5) | Gravitational constants of the planets |
| I | HRL | 1 | INPUT(23) | Hour of launch epoch |
| I | HRO | 1 | INPUT(53) | Hour of state epoch |
| 1 | IDATT | 1 | INPUT(1087) | Element set number of attitude desired from element set. |
| I | IDSAT | 1 | INPUT(1089) | Satellite identification number |
| I | IFIND | 1 | INPUT(1076) | Element set number of state desired from GTDS file |
| 0 | JC | 1 | CNTRL(07) | C $\epsilon$ ntral planet number |
| I | JL | 1 | INPUT(1015) | Launch planet number |
| I | JMN | 1 | INPUT(1017) | Planetary ephemeris flag |
| I | JT | 1 | INPUT(1031) | Target planet number |


| 1/0 | $\begin{gathered} \text { SYMBOLIC } \\ \text { NAME } \end{gathered}$ | DIMENSION | $\begin{gathered} \text { CONMON } \\ \text { BLOCE } \\ \hline \end{gathered}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | KINPT | 1 | INPUT(1019) | Input state coordinate system flag |
| I | KOBL | 1 | INPUT(1029) | Lunar oblateness flag |
| I | KP | 12 | INPUT(1001) | Planets in system to be integrated |
| I | MODLEN | 1 | INPUT(1035) | Lunar field model number |
| I | NATUNT | 1 | INPUT(1088) | I/O unit number of attitude file |
| I | OBSLAT | 10 | INPUT(480) | Observation sites latitudes |
| I | OBSL.ON | 10 | INPUT(410) | Observation sites longitudes |
| I | PCON | 1 | INPUT(5) | Position unit conversion factor |
| 1 | PI2 | 1 | CONST(3) | Twice pi. |
| I | RA | 1 | INPUT(47) | Initial right ascension of attitude |
| I | RAD | 1 | CONST(1) | Radian to degrees conversion factor. |
| 1 | RE | 12 | CONST(17) | Mean radius of the planets |
| I | RINIT | 1 | INPUT(444) | Desired orbit radius at target planet |
| I | SECL | 1 | INPUT(25) | Seconds of launch epoch |
| I | SECO | 1 | INPUT(55) | Seconds of state epoch |
| 0 | SOL | 1 | STATE (36) | Solar pressure constant |
| I | SOLARA | 1 | INPUT(433) | Spacecraft area |
| I | SPRESS | 1 | INPUT(197) | Solar pressure at 1 au |
| I/O | T | 1 | STATE(10) | Initial time since state epoch. |
| 0 | TCA | 1 | STATE(29) | Time of closest approach |
| I/O | TIG | 6 | INPUT(380) | Ignition and burnout times of the engines. |
| I | TMM | 1 | INPUT(196) | Epoch of lunar elements |
| 0 | UJT | 1 | STATE(32) | Current modified julian date |
| 1 | VCON | 1 | INPUT(6) | Velocity conversion factor |


| I/O | $\begin{aligned} & \text { SYMBOLI } \\ & \text { NAME } \end{aligned}$ | DIMENSION | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | VINTT | 1 | INPUT(445) | Circular velocity of desired target planet's final orbit |
| I | WO | 1 | INPUT (38) | Initial weight |
| I | WP | 12 | CONST(29) | Rotation rates of the planets |
| 0 | WT | 1 | STATE(34) | Current weight |
| 0 | WTI | 1 | STATE (35) | Weight after engine burn |
| 0 | X | 3 | STATE(1) | Position |
| I | XMINL. | 1 | INPUT(24) | Minutes of launch epoch |
| I | XMINO | 1 | INPUT(54) | Minutes of state epoch |
| I | XMONL | 1 | INPUT(20) | Month of launch epoch |
| I | XMONO | 1 | INPUT(50) | Month of state epoch |
| I | XMOON | 6 | INPUT(190) | Initial position of moon |
| 1 | YRL | 1 | INPUT(22) | Year of launch epoch |
| 1 | YRO | 1 | INPUT(52) | Month of launch epoch |

## Description:

The function of this subroutine is to initialize constants and flags and perform coordinate rotations when necessary. This subroutine does not have a complicated logic flow or theory. Thus, this description will consist of a chronological description of the functions performed.

1. Initialize Encke rectification factor.
2. Set up initial state, epoch, and covariance matrix from a call to FIND, if IFIND is non-zero.
3. Determine the modified julian date of the state epoch and the ephemeris time correction. If the ephemeris time correction is not input, it is calculated from ETC $=$ DJO $+(38.66+.0025921$ (DJO -40000) ), where ETC is the ephemeris time correction in seconds, and DJO is the modified julian date of the state epoch.
4. If the liftoff epoch is not input, set it equal to the state epoch.
5. Set up the solar.pressure constant as follows:

$$
\text { SOL }=10^{-8}(\text { SOLARA })(\text { SPRESS }) \mathrm{AU}^{2}
$$

where SOL is the solar pressure constant, SOLARA is the spacecraft area in $\mathrm{cm}^{2}$, SPRESS is the solara pressure at IAU in dynes $/ \mathrm{cm}^{2}$, and AU is the astronomical unit.
6. Set the central planet equal to the launch planet.
7. If the state is initialized through subroutine FIND (IFIND $=0$ ) skip to 10 , otherwise calculate the position and velocity vectors if orbital elements are input.
8. Convert input units to program units ( KM and $\mathrm{KM} / \mathrm{SEC}$ ).
9. Rotate from input coordinate system to Earth mean equator and equinox of 1950.
10. Initialize weights in STATE common for use in engine burns.
11. Set up ANKOR common and determine the circular velocity of the desired final orbit.
12. If IDATT is non-zero, determine the initial right ascension and declination of the attitude vector from a read from unit number NATUNT.
13. Initialize ATT, a unit vector in the direction of the attitude, from the initial right ascension and declination.
14. If the Moon is to be simulated by osculating elements, determine those elements from XMOON using subroutine ORBIT and set up MOON common.
15. If the tape or disk ephemeris is to be used, perform initial reads.
16. Set up the DST array for use in subroutine CLOSE to determine the central planet.
17. Set up the observation site common. The observation site common consists of the following information:
a. Vector from the center of the Earth to the site in Earth-fixed coordinates (XOBS).
b. The velocity of the site (DOBS).
c. Rotation matrix from the Earth equator and Greenwich to a site local coordinate system (OBSROT).

Calling Sequence:
Purpose:

CALL SHADOW

This routine calculates the times of umbral and penumbral passage with respect to the launch and target planets. It also calculates the times that the spacecraft is occulted by the target planet.

Common Blocks Required: CNTRL, CONST, INPUT, INTVAR, PLNET, SHAD, STATE

Subroutines Required: DVMAG, INTERP, PLANET

Input / Output

| I/O | SYMBOLIC <br> NAME | DTMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :--- |
| I/O | DSAD | 3,5 | SHAD(1) | Array of back distances to the <br> shadow cones |
| I | ELM | 6 | STATE(14) | ppacecraft's osculating <br> orbital elements |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | JL | 1 | INPUT(1015) | Launch planet number |
| I | KFIRST | 1 | CNTRL(12) | First pass flag |
| I | METH | 1 | INPUT(1013) | Trajectory propagation indicator |
| I | T | 1 | STATE(10) | Current time |
| $I$ | TSAD | 1 | SHAD (16) | Times of back shadow distances |
| $O$ | TSX | 1 | SHAD(19) | Array of shadow passage times |
| I | X | 6 | STATE(1) | Current position and velocity <br> vectors |

Theory:
The distance to the shadow cone, $D$, can be determined from trigometric relationships by examining Figure 1.


Figure 1

$$
\begin{equation*}
\mathrm{D}=[\mathrm{RSC} \sin \theta+(\mathrm{RSC} \cos \theta-\mathrm{RE} \sin \alpha) \tan \alpha-\mathrm{RE}] / \cos \alpha \tag{1}
\end{equation*}
$$

where $\alpha$ is the half cone angle obtained from

$$
\sin \alpha=\left(\mathrm{RE}_{\text {sun }}-\mathrm{RE}\right) / \mathrm{AXS}
$$

4 is the angle between the vector from the planet to the spacecraft and the Sun to the planet.

RE is the radius of the planet

The same type of cone is employed for occultation except that the Sun is replaced by the launch planet and the planet is the target planet.

The distance to the penumbral cone is obtained in a similar manner, (see Figure 2)


$$
\begin{equation*}
D=[R S C \sin \theta-(R S C \cos \theta-R E \tan a)-R E] / \cos \alpha \tag{2}
\end{equation*}
$$

where in this case, the half cone angle, $\alpha$, is obtained from

$$
\alpha=\left(R E{ }_{\text {sun }}+R E\right) / A X S
$$

## Description:

The times of umbral and prenumbral passage are determined in this subroutine. There are a total of ten times which may be determined in this routine. These times consist of entrance and exit of the following cones:

$$
\begin{aligned}
& \mathrm{KJ}= \text { 1. Launch planet umbra } \\
& \text { 2. Launch planet penumbra } \\
& \text { 3. Target planet umbra } \\
& \text { 4. Target planet penumbra } \\
& \text { 5. Occultation by target planet }
\end{aligned}
$$

There is a loop around the logic that determines the times. The internal flag, KJ, is the index of this loop and determines which shadow time is being calculated. (See above Table).

After the internal constants and flags are set up for loops and other purposes, the subroutine determines the distance from the shadow cone using equations (1) or (2). There are two criteria used to determine whether a shadow cone could have been passed on the last compute step.

1. Present position inside shadow cone while last position outside cone, or vice versa.
2. Spacecraft passed a minimum to the cone.

If neither of these criteria are satisfied, the spacecraft could not have flown through the shadow cone on the last step. At this point, flow is transferred to the location where the flags are set to loop on the next shadow time. If one of the criteria is satisfied, the spacecraft could have flown through the cone and the logic flow is transferred to the location where the time of shadow crossing is determined. The time of crossing is determined through a Newton-Raphson type iteration with subroutine INTERP. INTERP is used to determine the spacecraft state at the same time (T). This state is used in equations (1) or (2) to determine the distance to the shadow cone. $T$ is adjusted by the Newton-Raphson scheme to drive the distance from the shadow cone to zero. The N-R
iteration scheme, as implemented, determines the time of shadow crossing from

$$
\begin{align*}
& \mathrm{T}=\mathrm{T}+\mathrm{DELT} \\
& \mathrm{DELT}=\frac{\mathrm{dt}}{\mathrm{dD}} \mathrm{D} \tag{3}
\end{align*}
$$

where D is the distance to the shadow cone and

$$
\begin{aligned}
& \frac{\mathrm{dt}}{\mathrm{dD}} \quad \text { is the inverse of the derivative of the distance from } \\
& \text { the shadow cone with respect to time. }
\end{aligned}
$$

The derivative of the distance to the shadow cone with respect to time is determined analytically from the spacecraft's velocity vector and position with respect to the shadow cone. The derivative is equal to the component of the spacecraft's velocity along the direction normal to the shadow cone. This is represented vectorially by

$$
\begin{equation*}
\frac{\mathrm{d} D}{\mathrm{dt}}=\overline{\mathrm{V}} \cdot \hat{\mathcal{Z}} \tag{4}
\end{equation*}
$$

where $\quad \overline{\mathrm{V}}$ is the spacecraft's velocity and


Define the $X^{\prime}, Y^{\prime}, Z^{\prime}$ coordinate system with $X^{\prime}$ along the $X S$ vector, $Z^{\prime}$ along the $\bar{N}^{\prime}$ vector, and $Y^{\prime}$ defining a right-handed system, Figure $3 . \overline{\mathrm{N}}$ is obtained from

$$
\hat{\mathrm{N}}=(\overline{\mathrm{XSC}} \mathrm{x} \overline{\mathrm{XS}}) /|\overline{\mathrm{XSC}}| \quad|\overline{\mathrm{XS}}|
$$

Then, the vector normal to the shadow cone, $\hat{Z}$, in the primed system is expressed
as $\{\hat{z}\}=\left\{\begin{array}{c}-\sin \alpha \\ \cos \alpha \\ 0\end{array}\right\}$
$\hat{z}$ in the original coordinate system is $\{\hat{z}\}=[B]\{z\}$
where $[B]$ is the rotation matrix from the primed system to the original system.

The first column of the $B$ matrix is a unit vector along $\overline{\mathrm{XS}}$. The third column is the vector $\hat{N}$ while the middle column is a vector obtained by the cross-product of $\hat{N}$ and a unit vector along $\overline{\mathrm{XS}}$. Equations (4) to (6) define the derivative of the distance shadow cone with respect to time.

There is a limit to the time step allowed on each iteration. There are usually no convergence problems when one point is inside the shadow cone and the other outside. However, there is some convergence problem when a minimum to the shadow cone is passed. The problem is illustrated in Figure 4.


Figure 4
In this example, the spacecraft flight path just dips into the shadow cone. The problem arises when the derivative is calculated at point 1 . The $\mathrm{N}-\mathrm{R}$ technique would calculate point 2 as the entrance time. Since the minimum was skipped over, the iteration would converge on the shadow exit time instead of the entrance time it was expecting to determine. This problem was solved by testing on the sign of the derivative. Logic is included to reset the iteration to the last step and halve the calculated change in time, if the derivative changes sign. If the derivative changes sign more than five times, it is assumed that the spacecraft does not fly through the shadow cone. The shadow times determined are stored in the TSX array of SHAD common. The times are output from the array in subroutine SADOUT.

The DSAD array is updated on each of the passes in the KJ loop. On completion of this loop, the TSAD array is updated and the subroutine terminates.




250


Set large times for shadow times


## SUBROUTINE SHORB2

| Calling Sequence: | CALL SHORB2 (ELM, SUN, RC, RSUN, JUMB, <br> JPEN, UMBIN, UMBOUT, PENIN, PENOUT) |
| :--- | :--- |
| Purpose: | This subroutine calculates the true anomaly of a <br> given orbit that intersects the umbra and/or <br> penumbra of a planet with respect to the Sun or <br> another planet. |
| Common Blocks Required: $\quad$ None |  |
| Subroutine Required: $\quad$ ROTATE, QUARTC |  |

Input / Output

| I/ $\varnothing$ | SYMBOLIC <br> NAME | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | ELM | 6 | Calling <br> Operand | Orbital Elements |
| I | SUN | 3 | Calling <br> Operand | Position coordinates of Sun |
| I | RC | 1 | Calling <br> Operand | Radius of Planet |
| I | RSUN | 1 | Calling <br> Operand | Radius of Sun |
| I/O | JUMB/JPEN | 1 | Calling <br> Operand | Umbra and Penumbra Flag |
| $O$ | UMBIN | 1 | Calling <br> Operand | True anomaly when spacecraft <br> enters umbra. |
| $O$ | UMBOUT | 1 | Calling <br> Operand | True anomaly when spacecraft <br> leaves umbra. |
| $O$ | PENIN | 1 | Calling <br> Operand | True anomaly when spacecraft <br> enters penumbra. |
| $O$ | PENOUT | 1 | Calling <br> Operand | True anomaly when spacecraft <br> leaves penumbra. |

Note: ELM and SUN must be expressed in the same coordinate system. All input/ output angles are in radians.

Description: The true anomaly at which a spacecraft enters or leaves the UMBRA/PENUMBRA may be found by solving a quartic in the cosine of the true anomaly. For near circular orbits (E .LT. . 0015), a direct calculation is made for a close approximation of the intersections.

On input, the flags JUMB/JPEN must be set not equal to zero to calculate the UMBRAL/PENUMBRAL intersections respectively.

On output, the flags will have values $1-5$ as follows:
(1) Impact or escape central planet (no shadows calculated)
(2) No shadow possible
(3) No real solutions on night side of terminator
(4) Solutions found
(5) No real solutions found

If JUMB/JPEN .NE. 4 on return, the output quantities UMBIN, UMBOUT/PENIN, PENOUT will be zero.

Note: The output quantities UMBIN, UMBOUT/PENIN, PENOUT will have values $-2 \pi \leq \emptyset \leq 2 \pi$ with UMBOUT $\geq$ UMBIN and PENOUT $\geq$ PENIN.

Theory: The distance of the spacecraft from the central planet is given by

$$
r=\frac{P}{1+e \cos (\theta+\gamma)}
$$

where $P$ is the semi-latus rectum, $e$ the eccentricity, $\boldsymbol{\gamma}$ the angle between pericenter and the projection of the shadow cone on the orbit plane, and $\theta$ is the in-orbit plane angle from the shadow cone centerline projection to the spacecraft.

The distance to the shadow is given by $\rho=R / \sin (\alpha+\beta)$ where $R$ is the planet radius, $\mathcal{O}$ is half the shadow cone angle, and $\mathcal{D}_{\text {is }}$ the angle between the line from the planet's center to the point on the shadow cone and the shadow cone centerline. (See Figure SHORB2.1)

THETA and BETA can be related through spherical trigonometry by
$\cos \theta=\cos \theta \cos \theta$, where $\theta$ is the angle between the shadow cone centerline and the orbit plane.

Since $\delta, \mathcal{Y}, \mathrm{P}$ and e are known constants, r and $\rho$ can be equated, yielding the following equation

$$
\frac{\mathrm{P}}{1+e \cos (\theta+\gamma)}=\frac{\mathrm{R}}{\left(\sin \theta \cos \delta \cos \theta+\cos \theta \sqrt{\left.1-\cos ^{2} \theta \cos ^{2} \theta\right)}\right.}
$$

which can be squared twice to yield the following quartic in $\cos (8)$.
$\operatorname{COEF}(5) * \mathrm{x}^{4}+\operatorname{COEF}(4) * \mathrm{x}^{3}+\operatorname{COEF}(3) * \mathrm{x}^{2}+\operatorname{COEF}(2) * \mathrm{x}+\operatorname{COEF}(1)=0$ where
$\mathrm{x}=\cos (\theta)$
$\operatorname{COEF}(1)=\mathrm{C}^{2}-\mathrm{D}^{2}$
COEF (2) $=-2 \mathrm{BC}$
$\operatorname{COEF}(3)=\mathrm{B}^{2}+2 \mathrm{AC}+\mathrm{D}^{2}\left(1+\cos ^{2} \delta\right)$
$\operatorname{COEF}(4)=-2 \mathrm{AB}$
$\operatorname{COEF}(5)=\mathrm{A}^{2}-\mathrm{D}^{2} \cos ^{2} 8$
and
$A=(P \sin \theta \cos \delta-\mathrm{Re} \cos \gamma)^{2}+(\mathrm{Re} \sin \gamma)^{2}+(\mathrm{P} \cos \theta \cos \delta)^{2}$
$B=2 R(P \sin 0 \cos \delta-R e \cos \gamma)$
$C=R^{2}\left(1-e^{2} \sin ^{2} \gamma\right)-P^{2} \cos ^{2} \alpha$
$D=2 P R e \sin \gamma \cos \alpha$
This quartic is solved in subroutine QUARTC, which also outputs the number of roots found. If real roots exist,then the correct sign of 0 must be found by returning to the original equation (since cosine is an even function). [cf. check quadrants of solutions]. Once the sign is known, $\cos \theta$ is replaced by $\vartheta$. Since the angle $\gamma$ is the true anomaly of the shadow cone centerline projection in the orbit plane, the smallest absolute values of will be the desired intersections with the cone. Therefore the roots are ordered smallest first.

It is still possible that the intersections lie on the wrong side of the planet, so a check is made to make sure that $\theta$ is less than the angle from the centerline projection to the edge of the planet. [cf. check that solutions are on right side of umbral terminator].

The actual true anomalles of intersection are given by $\boldsymbol{\gamma} \boldsymbol{\gamma}$ and the output quantities are ordered so that the spacecraft enters the UMBRA/PENUMBRA before exiting (i.e. UMBOUT is greater than UMBIN).

If the orbit is circular, then $r$ is constant and $r=\frac{R}{\sin \left(\alpha+\beta_{c}\right)}$, where $\bar{\beta}_{\mathrm{C}}=\operatorname{arc} \sin (\mathrm{R} / \mathrm{r})-\mathbb{0}$ which implies that $\boldsymbol{\theta}_{\mathrm{c}}=\operatorname{arc} \cos \left(\cos \beta_{\mathrm{C}} / \cos \delta\right)$. Here the quadrant checks are unnecessary since there are no extraneous roots. The theory is exactly the same for PENUMBRAL cone intersections except for the angles $\mathbb{C}$ and $\boldsymbol{y}$, describing the cone. The half cone angle $\mathbb{C}$ is $\operatorname{arc} \sin$ ((radius of Sun + radius of planet)/distance to Sun.) The angle $\boldsymbol{\gamma}_{\text {PENUMBRA }}$ equals $\boldsymbol{\gamma}_{\text {UMBRA }}$ minus $\boldsymbol{\gamma}$, since the cone is on the opposite side of the planet for PENUMBRA. The quadrant checks are slightly different in that the maximum values of 9 are desired and $\theta$ must be greater than the angle from the Sun to the edge of the planet. The remainder of the calculations remain the same.

It is important to note that the assumption has been made that $\boldsymbol{\delta}, \boldsymbol{\gamma}, \mathrm{P}$ and e are relatively constant throughout one period of revolution of the spacecraft.

Note: Items in square brackets refer to comment cards in subroutine.


GEOMETRICAL RELATIONSHIPS OF ORBIT AND NHAHOW .

## SUBROUTINE SOL

Calling Sequence:

## Purpose:

Common Blocks Required:
Subroutines Required:

CALL SOL

This subroutine determines the position of the Sun with respect to the Earth.

CNTRL, CONST, INPUT, INTVAR, PLNET, STATE•
M50MDT, OBLTY, ROTATE

Input / Output

\left.| SYMBOLIC |  |  |  | COMMON |  |  |
| :--- | :---: | :---: | :---: | :--- | :---: | :---: |
| I/O | NAME | DIMENSION | BLOCK |  |  |  |
| I | EJO | 1 | STATE (26) | Ephemeris time of state epoch |  |  |
| I | EJ1900 | 1 | STATE(27) | Ephemeris time since 1900 |  |  |
| I | JC | 1 | CNTRL(7) | Central planet number |  |  |
| I | T | 1 |  | INTVAR(1) |  |  |$\right]$| Current time |
| :--- |
| $O$ |

Theory:
The position of the Sun with respect to the Earth is determined from the mean motion of the Sun as described in the Supplement to the Nautical Ephemeris. The mean anomaly of the Sun is obtained from

$$
\begin{equation*}
\mathrm{G}=358.47584^{\circ}+.985600267 \mathrm{~d}-.1 .12(10)^{-5} \mathrm{D}^{2}-7(10)^{-8} \mathrm{D}^{3} \tag{1}
\end{equation*}
$$

where
d is the number of days since 1900 , and
D is the number of Julian oenturies since 1900 .
The argument of perifocus is obtained from

$$
\begin{equation*}
G A M=281.22083+4.70684(10)^{-5} \mathrm{~d}+3.39(10)^{-5} \mathrm{D}^{2}+7(10)^{-8} \mathrm{D}^{3} \tag{2}
\end{equation*}
$$

The eccentric anomaly can be determined from the power series expansion of Kepler's equation since the eccentricity, $e$, is small. This equation is

$$
\begin{equation*}
E=G+e \sin G+\frac{e^{2}}{2} \sin 2 G+\frac{e^{3}}{8}\left(\dot{3} \sin ^{3} G-\sin G\right) \tag{3}
\end{equation*}
$$

The X and Y components of the Sun are obtained from the ellipse as

$$
\begin{aligned}
\mathrm{X}^{\prime} & =\mathrm{a}(\cos \mathrm{E}-\mathrm{e}) \\
\mathrm{Y}^{\prime} & =\mathrm{a} \sqrt{1-\mathrm{e}^{2}} \sin \mathrm{E} \\
\mathrm{Z}^{\prime} & =0
\end{aligned}
$$

where
a is the semi-major axis of Earth's orbit about the Sun.

This X and Y position is rotated about the Z axis to account for the argument of perifocus. Thus

$$
\begin{aligned}
& \mathrm{X}=\cos (\mathrm{GAM}) \mathrm{X}^{\prime}-\sin (\mathrm{GAM}) \mathrm{Y}^{\prime} \\
& \mathrm{Y}=\sin (\mathrm{GAM}) \mathrm{X}^{\prime}+\cos (\mathrm{GAM}) \mathrm{Y}^{\prime} \\
& \mathrm{Z}=0
\end{aligned}
$$

This vector represents the position of the Sun with respect to the Earth in the mean equinox and ecliptic of date. This vector is rotated to the mean equator and equinox of 1950 using subroutines M50MDT, OBLTY to establish the rotation matrices and ROTATE to perform the matrix multiplication.

Calling Sequence:
Purpose:
Common Blocks Required:

Subroutines Required;

## CALL SOLP

This subroutine calculates the acceleration due to solar pressure.

CONST, GRAVTY, INPUT, PLNET, PERT, STATE

CROSS, DVMAG, VNORM

Input / Output

| SYMBOLIC |  |  | COMMON |  |  |
| :--- | :--- | :--- | :--- | :--- | :---: |
| I/O | NAME | DIMENSION | BLOCK | DEFINITION |  |
| I | ATT | 3 | STATE(11) | Unit vector along spacecraft attitude |  |
| I | KP | 12 | INPUT(1001) | Planets in the system |  |
| I | KSOLP | 1 | INPUT(1084) | Solar pressure flag |  |
| I/O | RCART | 3 | PERT(1) | Perturbing acceleration |  |
| I | REFLEK | 1 | INPUT(198) | Spacecraft reflectivity constant |  |
| I | SOL | 1 | STATE(36) | Solar pressure constant |  |
| I | W | 1 | STATE(35) | Current spacecraft mass |  |
| I | $X$ | 3 | GRAVTY(1) | Current spacecraft position vector with |  |

Theory:
The acceleration due to solar pressure can be calculated using two different models. In the first model, the solar pressure force acts along the radial direction from the sun. This model assumes the spacecraft is a sphere or a flat plate perpendicular to the Sun's rays. Thus the acceleration can be obtained from,

$$
\begin{equation*}
\overline{\mathrm{a}}_{\mathrm{sp}}=\frac{\mathrm{SOL}_{\mathrm{S}}}{\mathrm{WR}_{\text {sun }}} \quad(1-\text { REFLEK }) \quad \hat{\mathrm{X}}_{\text {sun }} \tag{1}
\end{equation*}
$$

where

| W | is the spacecraft mass |
| :---: | :---: |
| $\mathrm{R}_{\text {sun }}$ | is the distance from the Sun |
| REFLEK | is the reflectivity coefficient |
| $\mathrm{X}_{\text {sun }}$ | is the unit vector from the Sun to the spacecraft, and |
| SOL | is a constant defined by |
|  | $\text { SOL }=\mathrm{Au}^{2}(\text { SOLARA })(\text { SPRESS })(10)^{-8}$ |
|  | where SOLARA is the spacecraft area |
|  | SPRESS is the solar pressure at 1 Au |

In the second model, the spacecraft is assumed to be an object spinning about its centerline. This kind of a model will contribute two components to the solar pressure force. The first component is due to the absorbed light and is along the radial direction while the second component is due to reflected light and is normal to the centerline, see Figure 1. . reflected light


Figure 1

For this model, the radial component is determined from

$$
\begin{equation*}
\left(\overline{\mathrm{a}}_{\mathrm{sp}}\right)_{\mathrm{rad}}=\frac{\mathrm{SOL}}{\mathrm{~W} \mathrm{R}_{\text {sun }}} \cos \theta(1-\mathrm{REFLK}) \hat{\mathrm{X}}_{\text {sun }} \tag{2}
\end{equation*}
$$

where $\theta$ is defined in Figure 1.
The reflective contribution to the solar pressure is obtained from

$$
\begin{equation*}
\left(\mathrm{a}_{\mathrm{SP}}\right)_{\text {nor }}=2 \frac{\mathrm{SOL}}{\mathrm{WR}_{\text {sun }}} \text { REFLEK } \cos \theta \hat{\mathrm{F}}_{\mathrm{n}} \tag{3}
\end{equation*}
$$

where $\quad \hat{F}_{\mathrm{n}}$ is a unit vector normal to the plane containing the centerline and the Sun.

The total acceleration is the sum of the radial and normal components.

## Description:

The subroutine determines if the spacecraft is in the umbral cone of any planet before the solar pressure acceleration is determined. The distance from the umbral cone is calculated from equation (1) of the SHADOW subroutine description. If the spacecraft is in an umbral cone,the solar pressure is assumed to be zero and the subroutine returns. If the spacecraft is not in shadows, the solar pressure is calculated from equation (1) or equations (2) and (3) according to the setting of the KSOLP flag. The solar pressure acceleration is added to the perturbing acceleration in RCART before the subroutine returns.

## SUBROUTINE SPER

## Calling Sequence:

Purpose:

CALL SPER (X, Y)
To convert the Cartesian coordinates $X(1), X(2)$, and $X(3)$ to spherical coordinates.
$Y(1)=\sqrt{X(1)^{2}+X(2)^{2}+X(3)^{2}}$
$\mathrm{Y}(2)=\tan ^{-1}\left(\mathrm{X}(3) / \sqrt{\left.\mathrm{X}(1)^{2}+\mathrm{X}(2)^{2}\right)},-90^{\circ}<\mathrm{Y}(2)<90^{\circ}\right.$
$Y(3)=\tan ^{-1}(X(2) / X(1)), \quad-180^{\circ}<Y(3)<180^{\circ}$
Units of $Y(2)$ and $Y(3)$ are degrees.
Common Blocks used: None

Subroutines Required: None

Input / Output

|  |  |  |  |
| :--- | :---: | :---: | :---: |
| I/O | SYMBOLIC <br> NAME | DIMENSION | DEFINITION |
| I | X | 3 | Input Cartesian Vector |
| O | Y | 3 | Output Spherical Coordinates |

## SUBROUTINE SPNM

Calling Sequence: $\quad$ CALL SPNM (NMAX, S, C, P)

Purpose:
SPNM calculates the Legendre polynomials from

$$
\begin{equation*}
P_{o}^{o}(S) \text { to } P_{\text {NMAX }} \mathrm{NMAX}+1 \tag{S}
\end{equation*}
$$

Common Blocks Required: None
Subroutines Required: None
Reference:
GULICK, L. J., (1970), 'A comparison of Methods for Computing Gravitational Potential Derivatives", ESSA Technical Report, C \& GS 40.

Input/Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON <br> BLOCK | DEFINITION |
| :--- | :---: | :---: | :---: | :---: |
| I | C | 1 | Calling <br> Operand | $\left(1-\mathrm{S}^{2}\right)^{1 / 2}$ |
| I | NMAX | 1 | Calling <br> Operand | Highest degree desired +1 |
| O | P | 17,19 | Calling <br> Operand | Array of polynomials |
| I | S | 1 | Calling <br> Operand | Argument of polynomials |

Theory:
The associated Legendre polynomials are calculated from the following recursion relationship:

$$
\begin{array}{lll}
P_{0}^{o}(x)=1 & P_{1}^{o}(x)=x & P_{1}^{1}(x)=\left(1-x^{2}\right)^{1 / 2} \\
P_{2}^{o}(x)=\frac{1}{2}\left(3 x^{2}-1\right) & P_{2}^{1}(x)=3 x\left(1-x^{2}\right)^{1 / 2} & P_{2}^{2}(x)=3\left(1-x^{2}\right) \\
P_{n}^{m}(x)=(2 m-1)\left(1-x^{2}\right)^{1 / 2} & P_{n-1}^{m-1}(x)+P_{n-2}^{m}(x) &
\end{array}
$$

For a derivation of this relationship see the above reference.

## Description:

SPNM calculates the Legendre polynomials of $S$ for all degrees from zero to NMAX-1 and for all orders up to NMAX +1 , and stores them in the array $P$. $C$ must be equal to $+\left(1-s^{2}\right)^{1 / 2}$. NMAX must be less than or equal to 17 . The indices are both one greater than the corresponding index used in the literature. Thus $P(3,1)=P_{2}^{0}$ in the usual notation for the associated Legendre polynomials.


## FUNCTION SUNMIN

Calling Sequence:
Purpose:

FUNCTION SUNMIN (X, Y, Z)
This function determines the minimum angle between vector $Z$ and a plane formed by vectors X and Y . The angle must lie between X and Y .

CONST

CROSS, ROTATE, VNORM

Input / Output

|  | SYMBOLIC |  |  |  |
| :--- | :---: | :---: | :---: | :---: |
| I/O | NAME | DIMENSION | COMMON |  |
| BLOCK | DEFINITION |  |  |  |
| I | PI2 | 1 | CONST(3) | Twice pi |
| I | RAD | 1 | CONST(1) | Radian to degree conversion factor |
| I | X | 3 | CALLING <br> OPERAND | Vector X as described in Purpose |
| I | Y | 3 | CALLING <br> OPERAND | Vector Y as described in Purpose |
| I | Z | 3 | CALLING <br> OPERAND | Vector $Z$ as described in Purpose |

Theory:
A rotation matrix is determined which transforms to a coordinate system with the X-axis along the X vector, the Y -axis normal to the plane formed by the vectors X and Y , and the Z-axis forming a right-handed system. This matrix is established by first alculating the unit vectors $R$ and $S$ defined by,

$$
\begin{aligned}
& \hat{R}=\bar{X} \times \bar{Y} /|\bar{X}| \quad|\bar{Y}| \\
& \hat{S}=\overline{\mathrm{R}} \times \overline{\mathrm{X}} /|\overline{\mathrm{X}}|
\end{aligned}
$$

Then the first column of the matrix is the unit vector along vector X . The second column is composed of $\hat{S}$ while the last column is $\hat{R}$. The longitude of vectors $Y$ and $Z$ are
determined in the new coordinate system. Note, the longitude of vector X is zero. If the longitude of $\hat{Z}$ is between zero and the longitude of $\vec{Y}$, the minimum angle is equal to

$$
\text { SUNMIN } \left.=\sin ^{-1} \hat{(\mathrm{R}} \cdot \mathrm{Z}\right) /|\overrightarrow{\mathrm{Z}}| .
$$

If the longitude of $\overline{\mathrm{Z}}$ is not between $\overline{\mathrm{X}}$ and $\overline{\mathrm{Y}}$, the minimum angle is equal to the minimum of

$$
\begin{aligned}
& \operatorname{SUNMIN}_{1}=\cos ^{-1}(\hat{Z} \cdot \hat{\mathrm{X}}) \\
& \text { SUNMIN } \left._{2}=\cos ^{-1} \hat{(\mathrm{Z}} \cdot \hat{\mathrm{Y}}\right)
\end{aligned}
$$



Calling Sequence:

Purpose:

CALL TABINT (X, K, J, N, R)

This subroutine determines the spacecraft thrust and mass at the current time from input thrust and mass flow tables.

INPUT

None

Input/Output

| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| 1/0 | NAME | DTMENSION | BLOCK | DEFINITION |
| I | J | 1 | CALLING OPERAND | Location of the beginning of the dependent variable table in the input array |
| I | K | 1 | CALLING OPERAND | Location of the beginning of the independent variable table in the input array |
| I | N | 1 | CALLING OPERAND | Number of points in the table |
| 0 | R | 1 | CALLING OPERAND | Output current thrust or mass |
| I | X | 1 | CALLING OPERAND | Time from ignition of the desired engine. |

## Description:

The tables that are used to determine thrust or mass values are in two parts. The first part is the independent variable beginning in location $K$ of the input array. This array consists of the times from engine ignition and must be monotonically increasing. The second part of the table is an array beginning in location J of the input array. This array contains values of the thrust or mass flow rate at the corresponding times of the first array. For example, the thrust in location $\mathrm{J}+2$ occurs at the time since ignition input in location $\mathrm{K}+2$. The value of $J$ is used to determine if thrust or mass values are to be determined. If $J$ is greater than 350 and less than 379 , then the mass is to be determined. Otherwise, a thrust table is being used.

If the thrust is to be determined, the thrust is obtained from

$$
\begin{equation*}
\mathrm{R}=\mathrm{A}(\mathrm{~L})+(\mathrm{A}(\mathrm{~L})-\mathrm{A}(\mathrm{~L}-1))(\mathrm{X}-\mathrm{A}(\mathrm{M})) /(\mathrm{A}(\mathrm{M})-\mathrm{A}(\mathrm{M}-1)) \tag{1}
\end{equation*}
$$

where the time X lies between

$$
A(M) \text { and } A(M-1)
$$

and $A(L)$ is the thrust at time $A(M)$

The mass is determined by a trapezoidal integration of the mass flow table up to the current time, X , from

$$
\begin{equation*}
M=\sum_{i=1}^{N} \frac{1}{2} \cdot(A(J+i)-A(J+i-1))(A(K+i)+A(K+i-1)) \tag{2}
\end{equation*}
$$

where
N is the location of the last time in the independent variable array less than $X$.

The final mass is given by

$$
\begin{equation*}
\mathrm{R}=\mathrm{M}+\frac{1}{2}(\mathrm{X}-\mathrm{A}(\mathrm{~J}+\mathrm{N}))(\mathrm{A}(\mathrm{~K}+\mathrm{N})+\mathrm{R}) \tag{3}
\end{equation*}
$$

where
$R$ is determined similar to equation (1) except the mass flow rate table is used.

## SUBROUTINE TARGET

## Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Called:

CALL TARGET

TARGET computes the end constraint error vector for midcourse guidance calculations.

CONST, CNTRL, INPUT, MCCOM, PLNET, STATE

M50LEQ, MVTRN, BVE, ORIENT, DVMAG, RETDV, ORBIT, RETRO, CROSS, VNORM, ROTAIT

Input / Output

| I/O | $\begin{gathered} \text { SYMBOEIC } \\ \text { NAME } \end{gathered}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | JC | 1 | CNTRL(7) | Central body number of end state |
| I | RTD | 1 | CONST(1) | Radians-to-degrees conversion factor |
| I | UM | 12 | CONST(5) | Gravitational constant array ( $\mathrm{km}^{3} / \mathrm{sec}^{2}$ ) |
| I | GKS | 1 | CONST(45) | Earth's surface gravity (km/sec ${ }^{2}$ ) |
| I | WTO | 1 | INPUT(38) | Initial spacecraft weight (kg) |
| I | DJo | 1 | INPUT(46) | Julian date of anchor epoch (days) |
| I | ASPMC | 1 | INPUT(441) | Specitic impulse of the trim motor (sec) |
| I | RCIRC | 1 | INPUT(444) | Final desired orbit radius (km) |
| I | WDROP | 1 | INPUT(473) | Post-retro drop weight (kg) |
| I | JTARG | 1 | INPUT(1031) | Target body number |
| I | IVTI | 1 | INPUT(1078) | Overburn strategy key |
| I | NORMIN | 1 | INPUT(1080) | Retro optimization key |
| I | DV | 3 | MCCOM(12) | Midcourse correction impulse ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | DVS | 3 | MCCOM(15) | Spherical components of DV |


| 1/0 | SYMBOL NAME | DIMENSION $\begin{array}{ccc}\text { COMMON } \\ \text { BLOCK } & \text { DEFINITION }\end{array}$ |  |  |
| :---: | :---: | :---: | :---: | :---: |
| I | BVD | 2 | $\operatorname{MCCOM}(19)$ | Desired miss-vector if IBTR $=1$ (km) |
| I | DVB4 | 1 | $\mathrm{MCCOM}(24)$ | Previously expended midcourse velocity ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | DVRET | 1 | MCCOM $(25)$ | Retro velocity magnitude (km/sec) |
| 0 | TV | 3 | MCCOM(27) | Anti-retro unit vector (lunar equator ref) |
| 0 | DVT | 1 | MCCOM(30) | Trim velocity ( $\mathrm{km} / \mathrm{sec}$ ) |
| 0 | FUELT | 1 | MCCOM(31) | Trim fuel (kg) |
| 0 | XS | 6 | MCCOM(32) | Cartesian end state ( $\mathrm{km}, \mathrm{km} / \mathrm{sec}$, lunar equator) |
| O | WTF | 1 | $\mathrm{MCCOM}(47)$ | Post-midcourse spacecraft weight (kg) |
| I | PSID | 10 | MCCOM ${ }^{\text {(80) }}$ | Desired end constraint vector |
| 0 | PST | 10 | MCCOM(100) | End constraint error vector |
| 1 | IBTR | 1 | MCCOM(167) | Miss vector type key $(1=\mathrm{BT}, \mathrm{BR}, 2=\mathrm{RCA}, \mathrm{INC})$ |
| I | XP | 6,12 | PLNET(1) | Celestial body states at T $(\mathrm{km}, \mathrm{km} / \mathrm{sec})$ |
| I | X | 6 | STATE(1) | End state (km, km/sec) |
| I | T | 1 | STATE(10) | End time corresponding to X (sec) |

Description:
TARGET is a very important subroutine in the computation of midcourse guidance corrections. It computes the end constraint error vector, $\psi$, as a function of the desired end conditions and the end state. In the normal mode of operation, each guidance law is formulated to constrain radius of closest approach and inclination. Desired values of these quantities are combined with actual arrival energy and the direction of the approach asymptote to arrive at equivalent "desired" miss-vector components, $B \cdot T_{d}$ and $B \cdot R_{d}$. This computation is described in Appendix $A$ of Reference 1. The first two constraint error vector components are then formulated:

$$
\begin{aligned}
& \operatorname{PSI}(1)=\mathrm{B} \cdot \mathrm{~T}_{\mathrm{d}}-\mathrm{B} \cdot \mathrm{~T}_{\text {actual }} \\
& \operatorname{PSI}(2)=\mathrm{B} \cdot \mathrm{R}_{\mathrm{d}}-\mathrm{B} \cdot \mathrm{R}_{\text {actual }}
\end{aligned}
$$

where the actual miss-vector is computed in subroutine BVE. The third and fourth constraint error components are for time of flight and hyperbolic excess speed, respectively.

$$
\begin{aligned}
& \operatorname{PSI}(3)=\mathrm{TFS}-\mathrm{T} \\
& \operatorname{PSI}(4)=\operatorname{VINFD}-\operatorname{VINF}
\end{aligned}
$$

TFS is the desired time at closest approach measured from anchor epoch and VINFD is the square root of the desired arrival energy $\left(C_{3}\right)$. The fifth constraint error component is post-retro circular excess velocity, assuming the retro to burn at periapsis antiparallel to the velocity there.

$$
\operatorname{PSI}(5)=\text { VDAR }-\mathrm{VAR}
$$

VDAR, velocity desired after retro, is supplied by subroutine MCSET, having been computed to be circular velocity at the desired arrival radius plus an optional input increment. VAR, velocity after retro, is computed as the scalar difference between the arrival periapsis speed and the retro velocity magnitude, DVRET. DVRET is, in turn, computed from the rocket equation and the midcourse correction velocity in function RETDV. The sixth component of PSI is not an error, but rather is the total correction fuel.

$$
\operatorname{PSI}(6)=\text { WTO - WTF + FUELT }
$$

WTO-WTF is the midcourse correction fuel and FUELT is the trim correction fuel. FUELT is computed by the rocket equation from the trim velocity, DVT, about which more will be said later. The seventh and eighth components of PSI are the errors in achieving the desired periapsis radius, PRD, and the desired inclination, OINC.

$$
\begin{aligned}
& \operatorname{PSI}(7)=\mathrm{PRD}-\mathrm{PR} \\
& \operatorname{PSI}(8)=\text { OINC }-\mathrm{SINC}
\end{aligned}
$$

PR is the actual closest approach radius as computed from the end state in BVE. SINC is the post-retro inclination, which differs from the approach orbit's inclination only when the variable target inclination procedure is used.

## Retro and Trim Calculations

The retro-strategy implemented in TARGET depends on:

1. arrival energy, $\mathrm{C}_{3}$
2. radius of closest approach, $\mathbf{r}_{\mathbf{p}}$
3. inclination, $\mathbf{i}$
4. retro-velocity impulse, $\delta \mathrm{v}$
5. input option keys, NORMIN, IVTI and IBTR.

TARGET computes the "desired" $\mathrm{B} \cdot \mathrm{T}$ and $\mathrm{B} \cdot \mathrm{R}$ values as functions of these parameters as well as predicting:
a. firing true anomaly, $\theta$, on the approach hyperbola.
b. direction, TV, of the spin-axis at retro-fire.
c. trim velocity and fuel for in-plane trim.

## (Case 1: NORMIN = 2)

In this case, subroutine RETRO is used to optimize 9 and TV and to provide the trim velocity. Overburn strategies are ignored. Desired miss vector computation depends on IBTR and input miss parameters.
(Case 2: NORMIN $<2$, IBTR $=1$ )
Overburn strategies are ignored and retro is anti-velocity at periapsis.
(Case 3: NORMIN $<$, IBTR $=2$, IVTI $=0$ )
Same as case 2 except for computation of desired miss vector.
(Case 4: NORMIN $<2 ;$ IBTR $=2 ;$ IVTI $= \pm 1$ )
Same as case 3 for underburns*, otherwise variable target inclination computations are invoked.
(Case 5: NORMIN $<2$, IBTR $=2, \cdot I V T I= \pm 2$ )
Same as case 3 for underburns*, otherwise variable approach distance computations are performed for the desired miss vector, retro direction, and trim velocity.

* An underburn is defined as the condition when the retro impulse would be sufficient for circularizing at the desired radius if the closest approach radius were indeed the desired circular orbit radius.


## Variable Target Inclination

The VTI procedure is derived in Appendix B of Reference 1. It amounts to defining desired miss vector components, $B \cdot T$ and $B \cdot R$, in such a way that an out-of-plane retro-burn at periapsis of the approach hyperbola can render a circular orbit of the desired radius and inclination. This procedure is applicable only for overburns. The sketch shows the retro velocity impulse, $\Delta V$, in the plane of the approach periapsis velocity, $V_{p}$, and the approach angular momentum vector, $H$.


Geometry of the VTI Procedure
The direction of $\Delta V$ is given by

$$
\hat{\Delta V}=\hat{V}_{p} \cos \alpha+\hat{H} \sin \alpha
$$

and the new angular momentum vector's direction is

$$
\hat{\mathrm{H}}=\hat{\mathrm{H}} \cos \beta-\hat{\mathrm{V}}_{\mathrm{p}} \sin \beta_{0}
$$

The angles, $\alpha$ and $\beta$, are defined in the derivation of the VTI procedure as functions of $\mathrm{C}_{3}, \delta \mathrm{v}$, and the desired final orbit radius. The desired miss parameters are computed in such a way that the approach orbit's track differs by $\beta$ from the desired orbit's tract at $r_{p}$. If the closest approach and inclination constraint errors are exactly zero, the post-retro orbit will require no time.

## Variable Approach Distance

The VAD procedure, like the VTI, is applicable only to overburns. The basic idea is to define the desired miss vector magnitude in such a way that it the retro is fired appropriately in-plane at the desired circular orbit radius, $r_{c}$, the post-retro orbit will be circular. It is assumed, first of all, that closest approach radius, $\mathbf{r}_{\mathbf{p}}$, of the arrival hyperbola can be varied without significantly changing the arrival energy, $\mathrm{C}_{3}$, or the retro velocity impulse, $\delta v$. Assuming further that $r_{p}$ is less than the desired circular orbit radius, $r_{c}$, we can write

$$
\mathrm{V}\left(\mathrm{r}_{\mathrm{c}}\right)=\mathrm{v}_{\mathrm{r}} \hat{R}_{\mathrm{c}}+\mathrm{v}_{\theta} \hat{\theta}
$$

where $\mathrm{v}_{\mathrm{r}}$ is the radial component of velocity and $\mathrm{v}_{8}$ is the tangential component. The circular, post-retro velocity is:

$$
\mathrm{V}_{\mathrm{c}}\left(\mathrm{r}_{\mathrm{c}}\right)=\mathrm{v}_{\mathrm{c}} \hat{\theta}
$$

where $v_{c}=\sqrt{\frac{\mu}{r_{c}}}$. The required retro impulse, $\Delta V$, is

$$
\begin{aligned}
\Delta V & =v_{c}-V=v_{r} \hat{R}+\left(v_{c}-v_{\theta}\right) \hat{\theta} \\
\delta v^{2} & =v_{r}^{2}+\left(v_{c}-v_{\theta}\right)^{2}=v^{2}-2 v_{c} v_{\theta}+v_{c}^{2} \\
& =c_{3}+\frac{2 \mu}{r_{c}}-2-\frac{\mu}{r_{c}} \frac{h}{r_{c}}+\frac{\mu}{r_{c}} \\
& =c_{3}+\frac{3 \mu}{r_{c}}-2 \frac{\mu}{r_{c}} \frac{r_{p}}{r_{c}} \quad c_{3}+\frac{2 \mu}{r_{p}}
\end{aligned}
$$

and

The only unknown in the above equation is $r_{p}$. We can solve for $r_{p}$ as follows,

$$
\begin{aligned}
c_{3} r_{p}^{2}+2 \mu r_{p} & =\left\{\left[\delta v^{2}-c_{3}-\frac{3 \mu}{r_{c}}\right]\right. \\
r_{p} & \left.=\sqrt{\left.\frac{r_{c}}{2 v_{c}}\right\}^{2}+c_{3} b^{2}}\right\}^{2}-\frac{\mu}{c_{3}}
\end{aligned}
$$

The periapsis constraint error is re-defined for this case as:

PST (7) $=$ RP $-\operatorname{PR}$.

The desired miss-vector magnitude is easily formulated from $r_{p}$.

$$
\begin{aligned}
& \mathrm{BMAG}=\left(\frac{\mu}{\mathrm{c}_{3}}+\mathrm{r}_{\mathrm{p}}\right) \sin \left(\tan ^{-1} \frac{\mathrm{v}_{\mathrm{p}} \mathrm{v}_{\infty} r_{p}}{\mu}\right. \\
& \mathrm{v}_{\mathrm{p}}=\sqrt{\mathrm{c}_{3}+\frac{2 \mu}{r_{p}}}, \quad \mathrm{v}_{\infty}=\sqrt{\mathrm{c}_{3}}
\end{aligned}
$$

The true anomaly at which the retro motor is fired is found from

$$
\cos \theta=\left(\frac{p}{r_{c}}-1\right) / e
$$

where the sign of 9 is determined by input: if IVTI is positive, $\theta$ is negative and vice versa. Defining unit vectors $\hat{P}$ and $\hat{Q}$ along periapsis position and velocity vectors, respectively,

$$
\begin{aligned}
& \hat{\mathrm{R}}=\hat{\mathrm{P}} \cos \theta+\hat{\mathrm{Q}} \sin \theta \\
& \hat{\theta}=\hat{\mathrm{P}} \sin \theta+\hat{\mathrm{Q}} \cos \theta
\end{aligned}
$$

We define $\Delta \mathrm{V}$ in terms of an out-of-plane angle, $\alpha$, and a flight path angle, $\gamma$.

$$
\Delta V=\delta v[\hat{\theta} \cos \alpha+\hat{H} \sin \alpha) \cos \gamma+\hat{\mathrm{R}} \sin \gamma]
$$

If the periapsis constraint error is zero, $\alpha=\pi$, and the maneuver will be in-plane. If $\delta \mathrm{V}$ is too large to circularize at $\mathrm{r}_{\mathrm{c}}$ for the actual $\mathrm{r}_{\mathrm{p}}$ and $\mathrm{c}_{3}$, the maneuver will be out-of-plane. The direction out-of-plane is chosen to minimize the resultant inclination error. If $\delta v$ is too small, $\gamma$ will be chosen so that the radial component of post-retro velocity will be nulled, with the tangential component falling wherever it may as a result.

$$
\begin{aligned}
& \sin \gamma=\frac{\mathrm{v}_{\mathrm{r}}}{\delta \mathrm{v}} \\
& \cos \alpha=\frac{\mathrm{v}_{\mathrm{c}}-\mathrm{v}_{\theta}}{\delta \mathrm{v} \cos \gamma}, \text { or } \alpha=\pi \text { if }|\cos \alpha|>1
\end{aligned}
$$

## Trim Velocity

The trim maneuvers in TARGET do not attempt to remove inclination errors. They are treated simply as two-impulse Hohmann transfers from the post-retro orbit to the desired circular orbit. We require only the specification of periapsis radius and apo-apsis radius of the post-retro orbit to compute the required trim velocity. (See TRIM) These may be computed, given the post-retro energy, $C_{3 \dot{e}}$, and angular momentum, he. The general form of the post-retro velocity is

$$
\begin{aligned}
\dot{\mathrm{V}}_{\mathrm{a}} & =\mathrm{V}+\Delta \mathrm{V} \\
& =\left(\mathrm{v}_{\mathrm{r}} \hat{\mathrm{R}}+\mathrm{v}_{\theta} \theta\right)+\delta_{\mathrm{v}}[(\theta \cos \alpha+\mathrm{H} \sin \alpha) \cos \gamma+\hat{\mathrm{R}} \sin \gamma]
\end{aligned}
$$

so that the energy ( $\mathrm{C}_{3 \mathrm{e}}$ ) is

$$
\begin{aligned}
& C_{3 e}=v_{a}^{2}-\frac{2 \mu}{r}=\left(v_{r}+\delta v \sin \gamma\right)^{2}+\left(v_{\theta}+\delta v \cos \alpha \cos \gamma\right)^{2} \\
& +(6 \mathrm{v} \sin \alpha \cos \gamma)^{2}-\frac{2 \mu}{r}
\end{aligned}
$$

where $\mathbf{r}$ is the radius of retro-fire. The tangential post-retro velocity is the magnitude of $\hat{R} \times V_{a}$.

$$
\begin{aligned}
& \hat{\mathrm{R}} \times \mathrm{V}_{\mathrm{a}}=\left(\mathrm{v}_{\theta}+\delta \mathrm{v} \cos \alpha \cos \gamma\right) \hat{\mathrm{H}}-(\delta \mathrm{v} \sin \alpha \cos \gamma) \hat{\theta} \\
& \mathrm{h}_{\mathrm{e}}=\mathrm{r} \sqrt{\left(\mathrm{v}_{\theta}+\delta \mathrm{v} \cos \alpha \cos \gamma\right)^{2}+(\delta \mathrm{v} \sin \alpha \cos \gamma)^{2}}
\end{aligned}
$$ and Planetary Orbiting Missions, AMA 71-16, March, 1971.




## SUBROUTINE TIMEC

Calling Sequence:
Purpose:

## CALL TIMEC

This subroutine controls the time logic during numerical integration. Its primary functions are to determine the compute interval, discontinuity times, and stopping criteria.

Common Blocks Required: CONST, CNTRL, INPUT, INTER, INTVAR, PERT, SAVE, STATE

Subroutines Required:

CLOSE, CRASH, DOPLER, INTEG, MOTORS, OUT1, ORBIT, OUTPUT, SADOUT, SHADOW, TRMN, UPDATE.

Input / Output

| SYMBOLIC |  | COMMMON |  |  |
| :---: | :---: | :---: | :---: | :---: |
| I | DELT | 10 | INPUT(180) | Compute intervals |
| I | ERRC | 1 | INPUT(1) | Error control limit for automati compute interval determination |
| I | DELTMN | 1 | INPUT(3) | Minimum compute interval |
| I | DELTO | 1 | INPUT( 2 ) | Initial compute interval |
| I | DX | 3 | STATE(4) | Spacecraft velocity |
| 1/O | ELM | 6 | STATE(14) | Spacecraft orbital elements |
| I | GM | 12 | CONST (5) | Gravi tational constants |
| I | JC | 1 | CNTRL(7) | Central planet number |
| 1 | JT | 1 | INPUT(1031) | Target planet number |
| I/O | KCA | 1 | CNTRL(11) | Counter used in closest approach iteration |



## Description:

This subroutine controls the flow of logic during trajectory propagation by numerical integration. When the subroutine is initiated, the initial spacecraft state is in STATE common. The state at the final time resides in STATE common when the subroutine terminates.

The primary function of this subroutine is to determine the compute interval, discontinuity times, and stopping criteria. The discontinuity times consist of engine ignition and burnout times, final time, and time of closest approach. The compute interval is adjusted so that the integrator stops at those times exactly. The state and its derivatives are saved at the beginning of each step. If a discontinuity time is passed during the step, the state is restored to the saved value and the compute interval adjusted to integrate to the discontinuity time.

The compute interval is obtained from the input compute interval table, from the discontinuity logic to hit a discontinuity time, or, if the automatic compute interval option is used, the compute interval is determined by subroutine RKSEVN and passed through INTVAR common.

The time of closest approach to the target planet is determined by an iteration using subroutines CRASH and TIMEC. If the time of closest approach, determined by subroutine CRASH, is less than the current time, the state is restored to the last step and the state propagated to the time of closest approach. At this point, the time of closest approach is recalculated by subroutine CLOSE. If the last time of closest approach is within a specified tolerance of the new closest approach time, the iteration is converged and the time of closest approach determined. If the difference of the times is larger than a specified tolerance, the iteration has not converged and the process is repeated. A total of seven iterations are allowed with a conversion tolerance of ten seconds.

Besides its primary functions, TIMEC also performs many other functions. These consist of:

1. Calls to subroutine SHADOW to determine umbral, penumbral occultation times.
2. Determining the trajectory propagation method. If the method changes, logic is included to change the integration variables to the new set of integration variables.
3. Calls to UPDATE to set up arrays for interpolation.
4. Calls to OUT1 for trajectory output.
5. Logic to rectify the reference orbit when propagating the state using Encke.
6. Calls to subroutine CLOSE to determine the central planet.
7. Calls to subroutine DOPLER to get doppler output.
8. Calls to subroutine MOTORS to simulate an engine burn by impulsive velocity.





## SUBROUTINE TOBODY

| Calling Sequence: | CALL TOBODY (JC) |
| :--- | :--- |
| Purpose: | This subroutine flies along a Keplerian conic <br> an increment of time, and determines the <br> cartesian state at the end of the step. |
| Common Blocks Required: | CONST, DUM, STATE |
| Subroutines Required: | ORBIT, TRMN |

Input / Output

| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I | DT | 1 | DUM (10) | Input time step along conic |
| 0 | ELM | 12 | STATE(14) | Osculating orbital elements along with sines and cosines of $i, u, \Omega$ |
| I | GM | 12 | CONST(5) | Gravi tational constants |
| 1 | JC | 1 | CALLING OPERAND | Central planet number |
| I/O | X | 6 | STATE(1) | Position and velocity vectors of the state at the beginning of the step and, on output, these vectors at the end of the step |

## Description:

First, the orbital elements are determined from the initial position and velocity vectors using subroutine ORBIT. Next the mean motion is determined from

$$
P V=G M(J C) / \operatorname{SEM}^{3}
$$

where

$$
\begin{aligned}
& \text { GM(JC) is the gravitational constant of the central } \\
& \text { planet, and } \\
& \text { SEM } \quad \text { is the semi-major axis }
\end{aligned}
$$

The initial mean anomaly, AMO, is determined from the initial true anomaly using
subroutine TRMN. Next, the mean anomaly at the end of the time step, DT, is determined from

$$
\mathrm{AM}=\mathrm{AMO}+\mathrm{PV}(\mathrm{DT})
$$

The true anomaly at the end of the time step is determined from the mean anomaly using subroutine TRMN. Finally, the position and velocity at the end of the time step is determined using subroutine ORBIT using the final true anomaly in the orbital elements.

## SUBROUTINE TRIM

Calling Sequence: CALL TRIM (ELM, DELV)
Purpose: TRIM calculates the trim velocity to circularize at a desired radius and correct inclination

Common Blocks Required: CONST, INPUT
Subroutines Required: None

Input / Output


## Method:

Once the retro motor has fired, the spacecraft is in lunar orbit. Assuming this orbit to be elliptical with energy, $c_{3 e}$ and angular momentum $h_{e}$, we will proceed to define the trim sequence and the trim fuel, cost. The in-plane trim is accomplished with a two-impulse Hohmann transfer to circular orbit at the desired radius, $r_{d}$.

(2)

Case 1: $r_{a}>r_{d}$
(2)

(1)

Case 2: $\mathrm{r}_{\mathrm{a}}<\mathrm{r}_{\mathrm{d}}$

The periapsis radius, $r_{p}$, and apoapsis radius, $r_{a}$, can be defined from $c_{3 e}$ and $h_{e}$.

$$
\begin{array}{ll}
p=\frac{h^{2}}{\mu} & \text { semi-latus rectum } \\
e=\sqrt{1+\frac{c_{3 e} e^{p}}{\mu}} & \text { eccentricity } \\
r_{p}=\frac{p}{1+e} & \text { periapsis radius } \\
r_{a}=\frac{p}{1-e} \\
a=\frac{r p+r}{2} & \text { apoapsis radius } \\
2
\end{array}
$$

semi-major axis

Case 1: $\quad r_{a}>r_{d}$ The semi-major axis, $a_{i}$, and energy, $c_{3 i}$ of the intermediate orbit must be

$$
\begin{aligned}
& a_{i}=\frac{r_{a}+{ }_{d}}{2} \\
& \text { and } c_{3 i}=-\frac{\mu}{a_{i}}=-\frac{2 u}{r_{a}+r_{d}}
\end{aligned}
$$

The velocity at $r_{a}$ before the first impulse is

$$
r_{a}^{-}=\sqrt{c_{3 e}+\frac{2 \mu}{r_{a}}}=\sqrt{\frac{2 \mu r_{p}}{r_{a}\left(r_{a}+r_{p}\right)}}
$$

and after the impulse,

$$
v_{a}^{+}=\sqrt{c_{3 i}+\frac{2 \mu}{r_{a}}}=\sqrt{\frac{2 u r_{d}}{r_{a}\left(r_{a}+r_{d}\right)}}
$$

so that the first maneuver requires an impulsive velocity

$$
\delta v_{1}=\sqrt{\frac{2 \mu}{r_{a}}}\left|\sqrt{\frac{r_{d}}{r_{a}+r_{d}}}-\sqrt{\frac{r_{p}}{r_{a}+r_{p}}}\right|
$$

The velocity on the intermediate orbit at $r_{d}$ is

$$
\mathrm{v}_{\mathrm{d}}^{-}=\sqrt{\mathrm{c}_{3 \mathrm{i}}+\frac{2 \mu}{\mathrm{r}_{\mathrm{d}}}}=\sqrt{\frac{2 \mu \mathrm{r}_{\mathrm{a}}}{\mathrm{r}_{\mathrm{d}}\left(\mathbf{r}_{\mathrm{a}}+\mathbf{r}_{\mathrm{d}}\right)}}
$$

while the desired circular velocity is

$$
\mathrm{v}_{\mathrm{d}}^{+}=\sqrt{\frac{\mu}{r_{d}}}
$$

so that the second impulse is

$$
\delta v_{2}=\sqrt{\frac{\mu}{r_{d}}}\left|\sqrt{\frac{2 r_{a}}{r_{a}+r_{d}}}-1\right|
$$

and the total in-plane trim velocity is

$$
\delta v=\delta v_{1}+\delta v_{2}
$$

Case 2: $r_{a}<r_{d}$. In this case, it is slightly cheaper to transfer first from $r_{p}$ to $r_{d}$. By a derivation similar to that for Case 1 , we obtain

$$
\begin{aligned}
& \delta v_{1}=\sqrt{\frac{2 \mu}{r_{p}}}\left|\sqrt{\frac{r_{d}}{r_{p}+r_{d}}}-\sqrt{\frac{r_{a}}{r_{a}+r_{p}}}\right| \\
& \delta v_{2}=\sqrt{\frac{\mu}{r_{d}}}\left|\sqrt{\frac{2^{r_{p}}}{r_{p}+r_{d}}}-1\right|
\end{aligned}
$$

It should be noted that the two cases can be computed with the same equations by interchanging roles between $r_{a}$ and $r_{p}$.

## Inclination Change

The inclination is changed by a third impulse executed at the node of the orbit on the equator. The impulse is applied on the intermediate orbit of the in-plane adjustment at the nodal crossing of larger radius. Although this strategy is not usually optimal, it is convenient: it leaves the node invariant, it always permits a solution, and it separates the in-plane and out-of-plane trim costs. Let the larger nodal radius vector (of either the pre-trim or intermediate orbit, actually) be $R$ and its corresponding velocity be $\mathrm{V}^{-}$. If the inclination is to be changed by $\delta i$, the rotated post-impulse velocity, $\mathrm{V}^{+}$will be

$$
\mathrm{V}^{+} \cdot=\cos \delta \mathrm{i} \mathrm{~V}^{-}+(1-\cos s i) \hat{\mathrm{R}} \cdot \mathrm{~V}^{-} \hat{\mathrm{R}}+\sin \delta \mathrm{i} \hat{\mathrm{R}} \times \mathrm{V}^{-}
$$

and

$$
\begin{aligned}
\delta \mathrm{V}_{3} & =\mathrm{V}^{+}-\mathrm{V}^{-} \\
& =(\cos \delta \mathrm{i}-1) \mathrm{V}^{-}+(1-\cos \delta \mathrm{i}) \hat{\mathrm{R}} \cdot \mathrm{~V}^{-} \hat{\mathrm{R}}+\sin \delta \mathrm{i} \hat{\mathrm{R}} \times \mathrm{V}^{-} \\
\delta \mathrm{v}_{3}^{2} & =\left[(1-\cos \delta)^{2}+\sin ^{2} \delta \mathrm{i}\right]\left[\mathrm{V}^{-} \cdot \mathrm{V}^{-}-\left(\hat{\mathrm{R}} \cdot \mathrm{~V}^{-}\right)^{2}\right] \\
& =2(1-\cos \delta \mathrm{i})\left|\hat{\mathrm{R}} \times \mathrm{V}^{-}\right|^{2} \\
& =4 \sin ^{2}\left(\frac{\delta \mathrm{i}}{2} \quad\left|\hat{\mathrm{R}} \times \mathrm{V}^{-}\right|^{2}\right. \\
\delta \mathrm{v}_{3} & =2 \sin \left(\frac{\delta \mathrm{i}}{2}\right)\left|\hat{\mathrm{R}} \times \mathrm{V}^{-}\right|=2 \sin \left(\frac{|\delta \mathrm{i}|}{2}\right) \mathrm{v}_{\text {tangential }}
\end{aligned}
$$

To roughly estimate the cost of correcting inclination, assume $\left|\hat{R} \times V^{-}\right|=\sqrt{\frac{\mu}{r_{d}}}$ and $\sin \frac{\delta \dot{i}}{2}=\frac{\delta \mathbf{i}}{2}$.

$$
\delta \mathrm{v} \approx \sqrt{\frac{\mu}{\mathrm{r}_{\mathrm{d}}}} \delta \mathrm{i}=22.94\left(\frac{\mathrm{~m}}{\mathrm{sec}} / \mathrm{deg}\right) \delta \mathrm{i}^{\mathrm{o}}
$$

Determination of the larger-radius node is somewhat tricky and deserves some explanation.


The argument of pericenter, $\omega$, is either $\omega_{1}$ or $\omega_{2}$ as shown in the sketch, depending on which nodal sense is the ascending one. The true anomaly of the larger-radius node is computed according to the following scheme.

$$
\begin{aligned}
& -\pi / 2 \leq \omega \leq \pi / 2: f=\pi-\omega \\
& \frac{\pi}{2}<\omega<\frac{3 \pi}{2} \quad: \quad \mathrm{f}=2 \pi-\omega
\end{aligned}
$$

In either case, $\cos f=-\dot{\cos } \omega \mid$.

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

and the tangential velocity (needed in the plane-change formula) is

$$
v_{\text {tangential }}=\frac{h}{r}=\frac{\sqrt{\mu p}}{r}
$$

If $r$ is smaller than $R D$, the plane-change is executed at a node of the circularized orbit. An inclination tolerance, TRINC, offers a band within which no inclination correction is made. The correction is made only over to the tolerance band.

Calling Argument:
Purpose:

Common blocks required:
Subroutines required: MVTRN, ORBIT
Inputs/Outputs

| SYMBOLIC |  |  | COMMON |  |
| :---: | :---: | :---: | :---: | :---: |
| I/O | NAME | DIMENSION | BLOCK | DEFINITION |
| I | CINF | 1 | INPUT(446) | Desired inclination of final orbit |
| 0 | DV | 1 | Calling Operand | Magnitude of the first and second trim maneuvers |
| 0 | DX | 3,2. | PIT (3) | First and second trim maneuvers in same system as ELMI |
| I | ELM I | 12 | Calling Operand | Orbital elements of initial orbit |
| I | JC | 1 | CNTRL( 7 ) | Central planet number |
| I | LOPT | 1 | Calling Operand | Flag used to calculate trim velocity components. Non-zero to calculate components |
| I | F | 1 | Calling Operand | True anomaly of the first maneuver on the initial orbit |
| I | RFINAL | 1 | INPUT(444) | Desired final orbit radius |
| 1 | VFINAL | 1 | INPUT(445) | Desired final orbit velocity |
| 0 | XS | 6 | PIT(12) | State before firsttrim in same coordinate system as ELMI |

## Reference:

F. T. Sun, "Analytic Solution for Optimal Two-Impulse $180^{\circ}$ Transfer Between Noncoplanar Orbits and the Optimal Orientation of the Transfer Plane, AIAA Journal, Vol 7, No. 10, 1969.

## Theory:

This subroutine determines the optimal two-impulse $180^{\circ}$ transfer between noncoplanar orbits using the method described in the reference. Since a $180^{\circ}$ transfer is specified, the first impulse must be applied at the intersection of the initial and final orbit planes. Thus, the angle between the initial and final orbit planes and the position on the initial orbit where the maneuver is made can be obtained from the spherical trigonometric relationships, see Figure 1.


Figure 1

The angle from the reference plane to the common line of nodes in the initial orbit, $\lambda$, can be determined from the input initial true anomaly, $f$, and the argument of the ascending node of the initial orbit as,

$$
\begin{equation*}
\lambda=f+\omega \tag{1}
\end{equation*}
$$

Then, the angle between the two orbit planes, $\sigma$, can be determined from,

$$
\begin{align*}
\sin \beta= & \sin \mathrm{i}_{\mathbf{i}} \sin \mathbf{i}_{\mathrm{d}} / \sin \lambda  \tag{2}\\
\tan \frac{\sigma}{2}= & \frac{\sin \left(\frac{\lambda-\beta}{2}\right)}{} \begin{aligned}
\sin \left(\frac{\lambda+\beta}{2} \tan \left(\frac{i_{d}-i_{\mathbf{i}}}{2}\right)\right.
\end{aligned}
\end{align*}
$$

where i is the inclination
$\beta$ is the angle from the reference plane to the common line of nodes in the final orbit plane.
and the subscripts $i$ and $d$ refer to the initial and desired orbits, respectively.

The radius and velocity components can be determined from standard orbital relationships at the initial true anomaly, f.

The orientation of the transfer plane with respect to the initial and final orbit planes is described in Figure 2.


Figure 2

The angies $\mu_{1}$ and $\omega_{2}$ describe the orientation of the transfer plane with respect to the initial and final orbit planes, respectively.

If the inclination of the transfer plane with respect to the initial plane is specified, then the optimal velocity can now be determined using Sun's equation 10.

$$
\begin{align*}
\Delta V=\sqrt{\frac{\mu}{r_{1}}} & \left\{v_{r 1}^{2}+\left[\left(v_{T 1}^{2}-2 \sqrt{\frac{2 n}{n+1}} V_{T 1} \cos \omega_{1}+\frac{2 n}{n+1}\right)\right.\right. \\
& \left.\left.+\left(1-2 \sqrt{\frac{2}{n+1}} \quad \cos \omega_{2}+\frac{n}{n+1}\right)^{1 / 2}\right]\right\}^{2} \tag{3}
\end{align*}
$$

Where $:$ is the gravitationai constant, $r_{I}$ is tie radius at $f=\lambda-\mu_{d}, V_{R_{i}}$ and $V_{T I}$ are the rasial and transversal velocity components on the initial orbit relative to local circular velocity ( $\mathrm{V} / \mathrm{n} / \overline{\mu / r}$ ), and n is $\mathrm{r}_{\mathrm{d}} / \mathrm{r}_{\mathrm{i}}$.

The equation above is somewhat simplified from Sun's equation since the final orbit is circular. Thus,

$$
\begin{align*}
& \mathrm{V}_{\mathrm{R} 2}=0  \tag{4}\\
& \mathrm{~V}_{\mathrm{T} 2}=1
\end{align*}
$$

If the velocity of each trim maneuver is desired, then

$$
\begin{equation*}
\frac{\Delta V_{i}}{\Delta V}=1+\frac{1}{\sqrt{n}}\left[\frac{1-2 \sqrt{\frac{2}{n+1}} \cos \omega_{2}+\frac{2}{n+1}}{V_{T 1}^{2}-2 \sqrt{\frac{2 n}{n+1}} V_{T 1} \cos \omega_{1}+\frac{2 n}{n+1}}\right]^{1 / 2} \tag{5}
\end{equation*}
$$

and

$$
\frac{\Delta V_{2}}{\Delta V^{2}}=1+\sqrt{n}\left[\frac{V_{T 1}^{2}-2 \sqrt{\frac{2 n}{n+1}} V_{T 1} \cos u_{1}+\frac{2 n}{n+1}}{1-2 \sqrt{\frac{2}{n+1}} \cos u_{2}+\frac{2}{n+1}}\right]
$$

where $\Delta V_{1}$ and $\Delta V_{2}$ denote the magnitudes of the first and second trim maneuvers.

The direction these impulses are applied can be determined by noting the following relationships,

$$
\begin{align*}
& \Delta \mathrm{V}_{\mathrm{R} 1}=\mathrm{V}_{\mathrm{RT} 1}-\mathrm{V}_{\mathrm{R} 1} \\
& \Delta \mathrm{~V}_{\mathrm{N} 1}=-\mathrm{V}_{\mathrm{T} 1} \sin \omega_{1}  \tag{6}\\
& \Delta \mathrm{~V}_{\mathrm{T} 1}=\mathrm{V}_{\mathrm{TT} 1}-\mathrm{V}_{\mathrm{T} 1} \cos \mu_{1}
\end{align*}
$$

The second trim is determined in a similar manner as

$$
\begin{align*}
& \Delta \mathrm{V}_{\mathrm{R} 2}=\mathrm{V}_{\mathrm{R} 2}-\mathrm{V}_{\mathrm{RT} 2} \\
& \Delta \mathrm{~V}_{\mathrm{N} 2}=-\mathrm{V}_{\mathrm{T} 2} \sin \omega_{2}  \tag{7}\\
& \Delta \mathrm{~V}_{\mathrm{T} 2}=\mathrm{V}_{\mathrm{T} 2} \cos \omega_{2}-\mathrm{V}_{\mathrm{TT} 2}
\end{align*}
$$

In the above equations the components of velocity are defined as follows,

| $\mathrm{V}_{\mathrm{R1} 1,2}$ | radial component of velocity of the <br> initial or final orbit. |
| :--- | :--- |
| $\mathrm{V}_{\mathrm{T} 1,2}$ | Transversal component of velocity of the initial or <br> final orbit. |
| $\mathrm{V}_{\mathrm{TT} 1,2}$ | Transversal component of velocity of the initial or <br> final orbit written in the transfer plane. |
| $\mathrm{V}_{\text {RT1,2 }}$ | Radial component of velocity of the transfer orbit at <br> initial and final orbit crossings. |

All components of velocity, except the radial components of the transfer orbit in equations 6 and 7, are fixed by specifying the initial and final orbits. The radial component of velocity is determined from the condition that the tatal trim velocity is to be minimized. The total trim velocity is

$$
\begin{align*}
\Delta \mathrm{V}=\Delta \mathrm{V}_{1}+\Delta \mathrm{V}_{2}= & \sqrt{\Delta \mathrm{V}_{\mathrm{T} 1}^{2}+\Delta \mathrm{V}_{\mathrm{N}}^{2}+\left(\mathrm{V}_{\mathrm{RT} 1}-\mathrm{V}_{\mathrm{R} 1}\right)^{2}}  \tag{8}\\
& +{\sqrt{\Delta \mathrm{V}_{\mathrm{T} 2}^{2}}+\Delta \mathrm{V}_{\mathrm{N} 2}^{2}\left(\mathrm{~V}_{\mathrm{R} 2}-\mathrm{V}_{\mathrm{RT} 2}\right)^{2}}^{2}
\end{align*}
$$

Also,

$$
\begin{align*}
& \mathrm{V}_{\mathrm{RT} 1}=-\mathrm{V}_{\mathrm{RT} 2} \\
& \mathrm{~V}_{\mathrm{R} 2}=0 \tag{9}
\end{align*}
$$

since a $180^{\circ}$ transfer is specified and the final orbit is to be circular. Now, the partial derivative of the trim velocity with respect to $\mathrm{V}_{\mathrm{RT} 1}$ can be written as,

$$
\frac{\partial \Delta V}{\partial V_{R T 1}}=\frac{V_{R T 1}-V_{R 1}}{\Delta V_{1}}+\frac{V_{R T 1}}{\Delta V_{2}}=0
$$

or

$$
\begin{equation*}
\mathrm{V}_{\mathrm{RT} 1}=\frac{\Delta \mathrm{V}_{2} \mathrm{~V}_{\mathrm{Bl}}}{\Delta \mathrm{~V}_{1}} \tag{10}
\end{equation*}
$$

The components of the trim velocity obtained from equations 6, 7, and 10 describe the trim velocity with respect to the transfer plane. The trim velocity vector in the same coordinate system of the initial orbit is obtained through a three Euler angle rotation pictured in the figure below.


In the figure above, $x$ corresponds to the radial direction, $y$ to the transversal direction, and z to the normal direction.

The angular elements of the reference orbit define tive Euler angles. Thus the transformation from the $\mathrm{x}, \mathrm{y}, \mathrm{z}$ system to the $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ system is

$$
\left\{\begin{array}{l}
\mathrm{X}  \tag{11}\\
\mathrm{Y} \\
\mathrm{Z}
\end{array}\right\}=\left[\begin{array}{cll}
\cos \psi \cos \Omega & -\sin \psi \cos \Omega & \sin i \sin \Omega \\
-\cos i \sin \Omega \sin \psi & -\cos i \sin \Omega \cos \psi & \\
\cos \psi \sin \Omega & -\sin \psi \sin \Omega & -\sin i \cos \Omega \\
+\cos i \cos \Omega \sin \psi & +\cos i \cos \Omega \cos \psi & \\
\sin i \cos \psi & \sin i \sin \psi & \cos i
\end{array}\right] \quad\left\{\begin{array}{l}
x \\
y \\
z
\end{array}\right\}
$$

where $\psi=\mathrm{f}+\omega$

The above equation is used to transform the trim velocity components from an orbit plane coordinate system to the system of the reference orbit.

## Optimum Inclination of the Transfer Plane

The condition for the optimal orientation of the transfer plane is expressed by

$$
\begin{equation*}
\frac{\sin \omega_{2}}{\sin \omega_{1}}=-\mathrm{n}\left(\frac{1-2 \rho_{2} \cos \omega_{2}+\rho_{2}^{2}}{1-2 \rho_{1} \cos \omega_{1}+\rho_{1}^{2}}\right)^{1 / 2} \tag{12}
\end{equation*}
$$

where $f_{1}=\left(\frac{2 \mu r_{d} / r_{1}}{r_{d}+r_{1}}\right) \quad \frac{1}{V_{T 1}}$

$$
\rho_{2}=\left(\frac{2 \mu r_{1 / r_{d}}}{r_{d}+r_{1}}\right)^{1 / 2}
$$

Equation 12 along with the condition

$$
\begin{equation*}
\omega_{2}=\sigma+\omega_{1} \tag{13}
\end{equation*}
$$

yields a set of equations which can be solved for $u_{1}$ or $u_{2}$ to yield the optimum orientation of the transfer plane. The solutions to equation 12 resulted in a sixth order polynomial in $\sin \omega$. The equation was solved numerically in order to avoid the cumbersome task of solving a sixth order equation. A Newton-Raphson procedure was employed to determine the solution to equation 13 . Sun, in the reference, states that the solution is unique. Thus, the task of finding multiple solutions with the Newton-Raphson method is not required.

## Description:

The initial orbital elements and argument of the ascending node of the final orbit is brought into the subroutine via the argument list. The angular components of the initial and final orbits are used in equations 1 and 2 to determine the position on the initial orbit where the trim maneuver will take place. Next, the transversal and radial components of velocity and other quantities required for Sun's equations are determined. The Newton-Raphson technique described by equations 12 and 13 are used to determine the inclination of the transfer orbit with respect to the initial orbit.

The magnitude of the velocity is determined from Sun's equation 10. If only the magnitude of the velocity is required, the subroutine returns. This option is executed through the LOPT flag brought in through the argument list. If the flag is equal to zero, only the velocity magnitude is calculated. Otherwise, the components of the trim velocity in the same coordinate system as the initial orbital elements are determined.

Equations 6, 7, and 10 are used to determine the components of the trim velocity in the transfer plane. The first trim velocity is rotated from the transfer plane to the initial
orbit piane by a rotation through an angle uis about the radial velocity component. This rotation is expressed by,

$$
\left\{\begin{array}{l}
D X  \tag{14}\\
D Y \\
D Z
\end{array}\right\}=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \cos \omega_{1} & \sin \omega_{1} \\
0 & -\sin \omega_{1} & \cos \omega_{1}
\end{array}\right] \quad\left\{\begin{array}{c}
\Delta V_{R} \\
\Delta V_{T} \\
\Delta V_{N}
\end{array}\right\}
$$

Next, the rotation defined by equation 11 is employed to obtain the first trim velocity in the desired system. The angular quantities in this rotation are obtained from the initial orbit.

The trim velocity from equation 14 is added to the initial state to determine the transfer orbit initial conditions. These conditions are used in ORBIT to determine the elements of the transfer orbit. The normal, radial, and transversal components from equation 7 are used directly in equation 11 to determine the second trim velocity in the desired system. The elements used in equation 11 are from the transfer orbit for the second trim. The logic to rotate the trim components are accomplished in a loop. The IK flag is used to determine the current maneuver.

## SUBROUTINE TRIM2



Determine initial trim position from equations 1 and 2 .

Determine transversal components and quantities for velocity determination.

Newton-Raphson iteration to determine orientation of transfer orbit using equations 12 and 13.


Calculate magnitude from equation 3


Get magnitudes of each maneuver from equation 5.


Determine velocity components of first trim in transfer plane from equations 6 and 10.


Rotate to initial orbit plane using $\omega_{y}$.


Calling Sequence:

Purpose:

Common Blocks Required:
Subroutines Required:

$$
\mathrm{Y}=\mathrm{TRMN}(J, Q, \dot{\mathrm{E}})
$$

This function calculates the mean anomaly from the true anomaly and vice-versa.

CONST
None

Input / Output

| I/O | $\begin{aligned} & \text { SYMBO } \\ & \text { NAME } \end{aligned}$ | DIMENSTON BLOCK ${ }^{\text {COMMON }}$ BLOC ${ }^{\text {a }}$ DEFINITION |  |  |
| :---: | :---: | :---: | :---: | :---: |
| I | E | 1 | Calling Operand | Eccentricity |
| I | J | 1 | Calling Operand | Flag to determine the computation =1 Calculate mean from true anomaly $=-1$ Calculate true from mean anomaly |
| I | PI | 1 | CONST(2) | pi, $\pi$ |
| I | PI2 | 1 | CONST(3) | twice pi |
| I | Q | 1 | Calling Operand | Input mean or true anomaly |

Theory:
I. Calculation of the mean anomaly from the true anomaly. The equations to determine the mean anomaly are dependent on the eccentricity, e. If e is less than 1 , the mean anomaly is obtained as follows,

$$
E=\cos ^{-1}[(\cos f+e) /(1+e \cos f)]
$$

where $E$ is the eccentric anomaly and
$f$ is the input true anomaly.
Then the mean anomaly, $M$, is obtained from Kepler's equation, as

$$
M=E-e \sin E
$$

When hyperbolic motion is encountered e>1, the mean anomaly is obtained by first calculating the auxiliary variable, F, from

$$
\begin{equation*}
\tanh \left(\frac{\mathrm{F}}{2}\right)=\sqrt{\frac{\mathrm{e}-1}{\mathrm{e}+1}} \quad \tan \left(\frac{\mathrm{f}}{2}\right) \tag{3}
\end{equation*}
$$

Then the mean anomaly is determined from

$$
\begin{equation*}
\mathrm{M}=\mathrm{e} \sinh \mathrm{~F}-\mathrm{F} \tag{4}
\end{equation*}
$$

II. Calculating the true anomaly given the mean anomaly. Kepler's equation, equation (2), is solved for the eccentric anomaly given the mean anomaly. This is a transcendental equation and an iterative procedure is required to solve for $E$.

If a first-order Taylor series is used to represent Kepler's equation, it becomes

$$
\begin{align*}
& \mathrm{M}=\varnothing\left(\mathrm{E}_{0}\right)+\emptyset^{\prime}\left(\mathrm{E}_{0}\right) \Delta \mathrm{E}_{0}  \tag{5}\\
& \text { where } \quad \emptyset\left(\mathrm{E}_{0}\right)=\mathrm{E}_{0}-\mathrm{e} \sin \mathrm{E}_{0} \\
& \quad \emptyset^{\prime}\left(\mathrm{E}_{0}\right)=1-\mathrm{e} \cos \mathrm{E}_{0}
\end{align*}
$$

If we assume that

$$
\emptyset\left(\mathrm{E}_{0}\right)=\mathrm{M}_{0}
$$

then

$$
\mathrm{M}-\mathrm{M}_{0}=\Delta \mathrm{M}=\emptyset^{\prime}\left(\mathrm{E}_{0}\right) \Delta \mathrm{E}_{0}
$$

or

$$
\begin{equation*}
\Delta \mathrm{E}_{0}=\frac{\dot{\mathrm{M}}-\mathrm{M}_{0}}{1+\mathrm{e} \cos \mathrm{E}_{0}} \tag{6}
\end{equation*}
$$

A corrective term is added to the derivative of $\emptyset$ in order to help convergence. Thus the change in eccentric anomaly is given by

$$
\begin{equation*}
\Delta E_{0}=\frac{M-M_{0}}{1+e \cos E_{0}+.01 e \cos ^{3} E_{0}} \tag{7}
\end{equation*}
$$

The iterative process uses equations (2) and (7) as follows:

1. The initial mean anomaly is input, $\mathrm{M}_{0}$.
2. A guess is made for the corresponding eccentric anomaly, $\mathrm{E}_{0}$.
3. Equation (2) is used to determine the mean anomaly corresponding to $\mathrm{E}_{0}$.
4. The change in eccentric anomaly, $\Delta \mathrm{E}_{0}$, is determined from equation (7).
5. A new eccentric anomaly is determined from $\mathrm{E}=\mathrm{E}+\Delta \mathrm{E}$.
6. Equation (2) is used to obtain a new mean anomaly $M$.
7. If ( $\mathrm{M}-\mathrm{M}_{0}$ ) is sufficiently small, the solution has converged. If not, steps 4 through 7 are repeated until it is small.

The true anomaly is calculated from the eccentric anomaly using

$$
\begin{equation*}
\sin f=\frac{a}{r} \sqrt{1-e^{2}} \quad \sin E \tag{8}
\end{equation*}
$$

and

$$
\cos f=\frac{a}{r} \cos E-e
$$

The solution of Kepler's equation for hyperbolic orbits is essentially the same except that equation (2) is replaced by equation (4) and equation (7) is replaced by

$$
\begin{equation*}
\Delta F=-\frac{M-M_{0}}{1+e \cosh F_{0}} \tag{9}
\end{equation*}
$$

The true anomaly is calculated from the auxiliary variable, $F$, using the inverse tangent

$$
\begin{equation*}
\tan f=\frac{\sqrt{e^{2}-1} \quad \sinh F}{\cos F-e} \tag{10}
\end{equation*}
$$




## SUBROUTINE TUBE1

Available from NASA Goddard Space Flight Center.

Calling Sequence:
Purpose:
CALI TWELVE (T)
This subroutine integrates a set of simultaneous differential equations using a twelfth-order predictor-corrector type method.

CONST, CNTRL, INPUT, INTVAR, PERT
DVMAG, EQNS, RKSEVN

Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | H | 1 | INTVAR(14) | Compute interval |
| I | JC | 1 | CNTRL(7) | Central planet number |
| I | KDIS | 1 | CNTRL(5) | Discontinuity flag |
| I | METH | 1 | INPUT(13) | Trajectory propagation indicator |
| I | RATES | 6 | INTVAR(8) | Derivatives of the dependent variables |
| I/O | T | 1 | CALLING OPERAND | Initial time on input and time on end of step or return |
| I | TOL | 1 | INPUT(8) | Corrector convergence tolerance |
| 1 | X | 1 | INTVAR(1) | Independent variable |
| I/O | Y | 6 | INTVAR(2) | Dependent variables |

Theory:

A set of simultaneous differential equations are numerically integrated over the time step, H. The scheme employed is a predictor-corrector scheme of summed ordinate form. Two sets of predictor and corrector equations are contained in this routine. When second derivatives of the quantities to be integrated are available, a Störmer predictor and a Cowell corrector are used. An AdamsBashforth predictor and an Adams-Moulton corrector are employed when only the first derivatives are available. A predictor-corrector numerical integration scheme determines the state at the end of the step as follows:

1. The predictor equation is used to extrapolate for the state at the end of the step ( $\mathrm{T}+\mathrm{H}$ ) using the back derivatives (or second derivatives) at times $\mathrm{T}, \mathrm{T}-\mathrm{H}, \mathrm{T}-2 \mathrm{H}$, etc.
2. The derivatives are determined at time $\mathbf{T}+\mathrm{H}$ using the state determined in 1.
3. The corrector equation is used to determine the state at $\mathrm{T}+\mathrm{H}$ using the back values of the derivatives at $T, T-H, T-2 H$, etc. and the derivative at $\mathrm{T}+\mathrm{H}$ from 2 .
4. The state obtained from the corrector equation is compared to the predictor state. If the difference is less than TOL, the solution is converged.
5. If the solution has not converged, the derivative is determined using the corrector state at $\mathrm{T}+\mathrm{H}$ and flow is transferred to Step 3. A total of 3 iterations are allowed.

The predictor equations are

$$
\begin{equation*}
\mathrm{Y}_{\mathrm{K}+1}^{\mathrm{p}}=\mathrm{h}^{2}\left[\Pi_{S_{k}}+\sum_{\mathrm{i}=1}^{10} \alpha_{\mathrm{i}} \quad \ddot{\mathrm{Y}}_{11-\mathrm{i}}\right] \quad \text { Störmer } \tag{1}
\end{equation*}
$$

$$
\begin{equation*}
\mathrm{Y}_{\mathrm{K}+1}^{\mathrm{p}}=\mathrm{h}\left[\mathrm{I} \mathrm{~S}_{\mathrm{k}}+\sum_{\mathrm{i}=1}^{10} \alpha_{\mathrm{i}}^{*} \dot{\mathrm{Y}}_{11-\mathrm{i}}\right] \quad \text { Adams-Bashforth } \tag{1}
\end{equation*}
$$

where $\alpha_{i}$ and $\alpha_{i}^{*}$ are constants described in Table I.
and $\mathrm{I}_{\mathrm{S}_{k}}$ and ${ }^{\mathrm{M}} \mathrm{S}_{\mathrm{k}}$ are the first and second sums defined by

$$
\begin{aligned}
& I_{S_{k+1}}=I_{S_{k}}+\ddot{Y}_{k+1} \\
& \Pi_{S_{k+1}}=\Pi_{S_{k}}+I_{S_{k+1}}
\end{aligned}
$$

or if first derivatives only are available

$$
\mathrm{I}_{\mathrm{S}_{\mathrm{k}+1}}=\mathrm{I}_{\mathrm{S}_{\mathrm{k}}}+\stackrel{\circ}{\mathrm{Y}}_{\mathrm{k}+1}
$$

The Cowell and Adams-Moulton corrector equations used are,

$$
\begin{array}{lll}
\mathrm{Y}_{\mathrm{K}+1}^{\mathrm{c}}=\mathrm{h}^{2}\left[{ }^{\mathrm{I}_{\mathrm{S}}}+\sum_{\mathrm{i}=1}^{10} \beta_{\mathrm{i}} \ddot{\mathrm{Y}}_{11-\mathrm{i}}\right] & \text { Cowell } \\
\mathrm{Y}_{\mathrm{K}+1}^{\mathrm{c}}=\mathrm{h} \cdot\left[\mathrm{I}_{\mathrm{S}_{\mathrm{k}}}+\sum_{\mathrm{i}=1}^{10} \beta_{\mathrm{i}}^{*}\right. & \left.\dot{\mathrm{Y}}_{11-\mathrm{i}}\right] & \text { Adams-Moulton }
\end{array}
$$

where $\beta_{i}$ and $\beta_{i}^{*}$ are constants described in Table I
The table of back derivatives is established using a seventh-order.ten-cycle Runge-Kutta integration scheme when the discontinuity flag is set to 1 . This flag is set on the first step, or at engine ignition or burnout times. The initial first and second sums are obtained from the corrector equation as follows,

$$
\begin{equation*}
\Pi_{S_{9}}=\frac{Y_{10}}{h^{2}}-\sum_{i=1}^{10} \beta_{i} \ddot{Y}_{11-i} \tag{3}
\end{equation*}
$$

$$
\begin{align*}
& I_{S_{9}}=\frac{\dot{Y}_{10}}{h}-\sum_{i=1}^{10} \beta_{i}^{*} \ddot{Y}_{11-i} \\
& I_{S_{10}}={ }^{I_{S}}{ }_{9}+\ddot{Y}_{10}  \tag{3}\\
& \Pi_{S_{10}}+{ }^{\Pi}{ }_{S_{9}}+{ }^{I_{S}}{ }_{9}
\end{align*}
$$

If the first derivative is used, the sum is obtained from

$$
\begin{aligned}
& I_{S_{9}}+\frac{Y_{10}}{h}-\sum_{i=1}^{10} \beta_{i}^{*} \dot{Y}_{11-i} \\
& I_{S_{10}}={ }^{I_{S}}{ }_{9}+\dot{Y}_{10}
\end{aligned}
$$

Only the first sum is needed.

## Description:

Logic to numerically integrate either first or second derivative equations is contained in the routine. The METH flag is tested to determine which set of equations to use. A series of flags are internally set according to the value of METH. These flags are used to determine the size of DO LOOPS and values of subscripts. The purpose of the flags are
\(\left.$$
\begin{array}{lccc}\text { FLAG } & \begin{array}{c}\text { 1st } \\
\text { Deriv: }\end{array} & \begin{array}{c}\text { 2nd } \\
\text { Deriv: }\end{array} & \\
\text { NK } & 6 & 3 & \begin{array}{l}\text { Number of quantities integrated }\end{array} \\
\text { NO } & 0 & 3 & \end{array}
$$ \begin{array}{l}Used to pick out the first derivatives <br>
when integrating second derivative <br>

equations.\end{array}\right]\)| IN |
| :--- |
| IS |

KDIS is tested to determine if it is necessary to proceed into the start-up logic. This logic is used to set up the back values of the derivatives in the XDD array. The values are obtained by integrating the set of equations with the seventh-order Runge-Kutta scheme. The compute interval is halved and the integration accomplished with RKSEVN. The derivatives at the end of every second step are
loaded into the XDD array. After the start-up integration, the first and second sums are initialized using equations (3).

Next, the predicted value is obtained from equation (1). Subroutine EQNS is used to get the derivatives of the function using the predicted state.

The corrector equation is employed to obtain the corrected value and the test for convergence is made. The integration is complete if convergence is obtained. If not, the derivation is obtained with the corrected state and the corrector equation employed again. Three iterations are allowed.

The derivative array, XDD, and the sums, S , are updated after convergence is achieved.

Note: This subroutine must be used with a constant, fixed step compute interval as there are no provisions for modifying the back value table.

## SUBROUTINE TWELVE



## TABLE I

## INTERPOLATION COEFFICIENTS

$\frac{\text { Stormer Predictor }}{\alpha}$
.709333000140291806
-2.94977592767957351
7.56536563552188551
-12.9585332742103575
15.3441157607824274
-12.6715074479918229
7.19372036335578001
-2.68438193943402276
.594237726972101971
-.0592405641233766233

| Adams-Bashforth |
| :--- |
| Predictor |

$\alpha^{*}$
3.4519884004562823
-13.8168372652617444
35.4336533489658489
-60.8353967251883918
72.1837392401194484
-59.7076813146344396
33.9395391414141414
-12.6774970438512105
2.80868181442400192
-.280189596443936721

Cowell Corrector
$\beta$
. 0592405641233766233
. 116927358906525573
-. 283950542127625460
.456997940716690716
-. 518014808301266634
. 415493601691513357
-. 230988982082732082

- 0848526685505852171
-. 0185565538820747153
.00183208573833573833 . 00678584998463470685


## SUBROUTINE TWOPIT

| Calling Sequence: | CALL TWOPIT(ELM, DUT) |
| :--- | :--- |
| Purpose: | This subroutine is used in conjunction with <br> TRIM 2 to determine the minimum two- <br> impulse, plane-change post-injection trim <br> maneuver. |
|  | CONST |
| Common Blocks Required: | TRIM2 |

Input/Output

|  |  |  |  |  |
| :--- | :---: | :---: | :---: | :---: |
| I/O | SYMBOLIC <br> NANE | PROGRAM <br> DIMENSION | COMMON <br> BLOCK | DEFINITION |
| 0 | DVT | 1 | Calling Argument | Total velocity requirement |
| $I$ | ELM | 6 | Calling Argument | Pre-trim orbit |
| $I$ | PI | 1 | CONST(2) |  |

## Description:

This subroutine is used in conjunction with TRIM2 to determine the optimum place on the initial orbit to initiate the two-impulse plane-change maneuver. Subroutine TRIM2 determines the optimum maneuver between two orbits. However, the position on the initial orbit must be specified. This subroutine uses a half-interval search to find the position on the initial orbit which results in the minimum trim requirements.

The search is accomplished by stepping the true anomaly around the initial orbit. On each step, subroutine TRIM 2 is called to determine the velocity requirement.

As TWOPIT steps along the true anomaly, the true anomalies on the two previous steps and their corresponding velocity requirements are saved in the OS and DVS arrays. The DVS array and the current velocity, DVT, is used to determine if a minimum velocity requirement has been passed. If a minimum is not passed the logic continues to step true anomaly. The subroutine goes into its halfinterval search phase when the minimum criterion is satisfied. The LPASS flag is non-zero when in the half-interval phase.

It is known that the minimum occurs between the current true anomaly and the true anomaly stored in OS(1) when the half-interval search phase is begun, see Figure 1.


TRUE ANOMALY
The true anomaly interval is halved and added onto the value in $\operatorname{OS}(1)$. Thenthe velocity requirement at this point is determined (1) in the figure). If the velocity at (1) is less than $\operatorname{OS}(2)$, the minimum must occur between $\operatorname{OS}(1)$ and OS(2). If this condition is true, the interval is halved again and the process repeated between the points $\operatorname{OS}(1)$ and (1). If the velocity at (1) is greater than $\operatorname{OS}(2)$, the minimum lies between(1) and $\phi$. The velocity requirement is then determined at (2) . If the velocity at (2) is less than DVS(2), the minimum must occur between OS(2) and $\phi$. Then,

$$
\begin{aligned}
& \operatorname{OS}(1)=\operatorname{OS}(2) \\
& \operatorname{OS}(2)=(1) \\
& \operatorname{DVS}(1)=\operatorname{DVS}(2) \\
& \operatorname{DVS}(2)=\operatorname{DV}(1))
\end{aligned}
$$

and flow transfers back to the beginning of the half-interval search. If the velocity at 2 is greater than DVS (2), the minimum occurs between (1) and (2) . For this case

$$
\begin{aligned}
& \mathrm{OS}(1)=\text { (1) } \\
& \mathrm{OS}(2)=\operatorname{OS}(2) \\
& \operatorname{DVS}(1)=\operatorname{DV}((1)) \\
& \mathrm{OS}(2)=\operatorname{DVS}(2)
\end{aligned}
$$

The flow returns to the beginning of the search.
The search is terminated when the interval in true anomaly has been halved enough times to cause it to become less than a stopping value. Logic is included to search the entire orbit for minimums. This is required because there are many local minimums. The subroutine terminates when the final true anomaly is more than $2 \pi$ greater than the initial value. Prior to termination, TRIM2 is called with the value of true anomaly corresponding to the absolute minimum and the input flag set to obtain components of the trim along with its magnitude.



## SUBROUTINE UPDATE

Calling Sequence:
Purpose:

Common Blocks Used:

Subroutines Used:

CALE UPDATE
This subroutine sets up the array of back derivatives used for interpolation in subroutine INTERP.

INPUT, INTER, INTVAR, STATE
None

Input / Output

| I/O | $\begin{gathered} \text { SYMBOI } \\ \text { NAME } \\ \hline \end{gathered}$ | DIMENSTON | COMMON BLOCK | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| 0 | ACL | 6,10 | INTER(71) | Array of back derivatives of the integration variables |
| I | INT | 1 | INTER(131) | Counter used to indicate the current value in arrays |
| 0 | POS | 6,10 | INTER(11) | Array of back values of the integration variables |
| I | RATES | 6 | INTVAR(8) | Current derivatives of the integration variables |
| I | T | 1 | STATE(10) | Current time |
| 0 | X | 10 | INTER(1) | Times of the back values |
| I | Y | 6 | INTVAR(2) | Current integration variables |

## Description:

This subroutine sets up arrays which contain the back values of the integration variables, derivatives of the integration variables and times at which the back values are stored in the arrays. The INT flag is used as an indicator to determine which back value is current. The back value arrays can hold up to ten back values. Only five of these back values are used. The longer table decreases the number of times components in the table must be moved. If INT is larger than 10, the last five back values of the table are set in the first five slots and INT reset to 5 . If INT is zero, the back values arrays are restored to zero.

## SUBROUTINE VIEW

Calling Sequence:

Purpose:

Common Blocks Required:

Subroutines Required:

CALL VIEW (X,Y,J, A)

To calculate the lighting characteristics of a planet as seen by the spacecraft.

CONST

None

Inputs / Outputs

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DESCRIPTION |
| :---: | :---: | :---: | :---: | :---: |
| O | A | 1 | CALLING <br> ARGUMENT | Percent of planet lighted |
| I | J | 1 | CALLING ARGUMENT | Planet number |
| I | RE | 12 | CONST(17) | Equatorial radius of the planets |
| I | X | 3 | CALLING ARGUMENT | Vector from spacecraft to the planet. |
| I | Y | 3 | CALLING ARGUMENT | Vector from Sun to planet. |

Theory:
Let $\theta$ be the s/c-planet-sun-angle
$\mathrm{x}, \mathrm{y}, \mathrm{z}$ system centered at the planet with
$x$ toward the $s / c, z$ normal to the $s / c-p l a n e t-s u n$ plane, and $y$ in the plane
$\mathrm{x}^{\prime}, \mathrm{y}^{\prime}, \mathrm{z}^{\prime}$ system same as $\mathrm{x}, \mathrm{y}, \mathrm{z}$, except rotated about z an angle $\theta$ such that $x^{\prime}$ is toward the Sun.

Then the equation for the terminator circle formed on the planet by the Sun is described

$$
\begin{align*}
& y^{\prime^{2}}+z^{\prime 2}=r_{s}^{2}  \tag{1}\\
& x^{\prime}=-x_{s}
\end{align*}
$$

where

$$
\begin{aligned}
& \mathrm{r}_{\mathrm{s}}=\mathrm{r}_{\mathrm{e}} \cos a \\
& \mathrm{x}_{\mathrm{s}}=\mathrm{r}_{\mathrm{e}} \sin a
\end{aligned}
$$


and $r_{e}$ is the planet's radius
$a$ is the half cone angle $y^{\prime}$


The rotation from the $x, y, z$ system to $x^{\prime}, y^{\prime}, z^{\prime}$ is obtained from

$$
\left\{\begin{array}{c}
\mathrm{x}^{\prime} \\
\mathrm{y}^{\prime} \\
\mathrm{z}^{\prime}
\end{array}\right\} \quad\left[\begin{array}{ccc}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{array}\right] \cdot\left\{\begin{array}{l}
\mathrm{x} \\
\mathrm{y} \\
\mathrm{z}
\end{array}\right\}
$$

Equation (1) written in the s/c system becomes

$$
\begin{align*}
& x \cos \theta+y \sin \theta=-x_{s} \\
& (-x \sin \theta+y \cos \theta)^{2}+z^{2}=r_{s}^{2} \tag{2}
\end{align*}
$$

The s/c also sees a circle on the planet described by

$$
\begin{align*}
& y^{2}+z^{2}=r_{c}^{2} \\
& x=x_{c} \tag{3}
\end{align*}
$$

where


$$
\begin{aligned}
& \mathrm{x}_{\mathrm{c}}=\mathrm{r}_{\mathrm{e}} \sin \beta \\
& \mathrm{r}_{\mathrm{c}}=\mathrm{r}_{\mathrm{e}} \cos \beta \\
& \beta=\sin ^{-1} \frac{\mathrm{r}}{\mathrm{r}}
\end{aligned}
$$

The terminator ellipse will intersect the viewing circle where the plane $x_{c}=x_{c}$ intersects the ellipse, or

$$
x_{c} \cos \theta+y \sin \theta=-x_{s}
$$

and

$$
\begin{equation*}
\left(-x_{c} \sin \theta+y \cos \theta\right)^{2}+z^{2}=r_{s}^{2} \tag{4}
\end{equation*}
$$

thus the intersection points are

$$
\begin{align*}
& y_{1}=\frac{-x_{s}-x_{c} \cos \theta}{\sin \theta}  \tag{5}\\
& z_{1,2}= \pm \sqrt[8]{r_{s}^{2}-y_{c}^{\prime 2}}
\end{align*}
$$

where

$$
\begin{aligned}
\mathrm{y}_{\mathrm{c}}^{\prime} & =-\mathrm{x}_{\mathrm{c}} \sin \theta+\left[\frac{-\mathrm{x}_{\mathrm{s}}-\mathrm{x}_{\mathrm{c}} \cos \theta}{\sin \theta}\right] \cos \theta \\
& =\frac{-\mathrm{x}_{\mathrm{c}} \sin ^{2} \theta-\mathrm{x}_{\mathrm{s}} \cos \theta-\mathrm{x}_{\mathrm{c}} \cos ^{2} \theta}{\sin \theta}=\frac{-\mathrm{x}_{\mathrm{c}}-x_{\mathrm{s}} \cos }{\sin \theta}
\end{aligned}
$$

The sun ellipse can be rewritten as

$$
\begin{align*}
& x=\frac{-x_{s}+y \sin \theta}{\cos \theta} \\
& {\left[\left(\frac{-x_{s}+y \sin \theta}{\cos \theta}\right) \sin \theta+y \cos \theta\right]^{2}+z^{2}=r_{s}{ }^{2}} \\
& {\left[\frac{-x_{s} \sin \theta+y \sin ^{2} \theta+y \cos ^{2} \theta}{\cos \theta}\right]^{2}+z^{2}=r_{s}{ }^{2}} \tag{6}
\end{align*}
$$

$$
\left(\frac{-x_{s} \sin \theta+y}{\cos \theta}\right)^{2}+z^{2}=r_{s}^{2}
$$

The shaded area of the clisk in the figure below is obtained from


$$
\begin{align*}
A & =\int_{z_{1}}^{z}\left(y-y_{1}\right) d z \\
= & \int\left\{\cos \theta \sqrt{r_{s}{ }^{2}-z^{2}}+x_{s} \sin \theta-y_{1}\right\} d z \\
& =\left[\frac{\cos \theta}{z}\left[z \sqrt{r_{s}{ }^{2}-z^{2}}+r_{s}^{2} \sin ^{-1} \frac{z}{r_{s}}\right]+z\left(x_{s} \sin \theta-y_{1}\right)\right]_{2}^{z_{2}} \tag{7}
\end{align*}
$$

The area of the rest of the circle (the crossed-hatched area) is determined from

$$
\begin{equation*}
\frac{\pi r^{2}}{z}+\left[y \sqrt{r_{c}^{2}-y_{1}^{2}}+r_{c}^{2} \sin ^{-1} \frac{y_{1}}{r_{c}}\right] \tag{8}
\end{equation*}
$$

## Description:

The areas determined in equations (7) and (8) are determined in a straightforward manner. The initial vectors are input via the argument list while the planet's radius is in CONST common. The area determined by the sum of equations (7) and (8) is divided by the total area to determine the percent lighted. If the spacecraft-planet-Sun angle is greater than 90 degrees, the percent calculated is the darkened area. Thus, the percent is subtracted from 100 to determine the lighted area.

Calling Sequence:
Purpose:
Common Blocks Required: CONST, INPUT, OBSIT, PLNET.
Subroutines Required: DOT, DVMAG, MVTRN, M50EPM

Input / Output

| I/O | SYMBOLIC <br> NAME | DIMENSION | COMMON |
| :--- | :---: | :---: | :---: | :--- |
| BLOCK |  |  |  |$\quad$| DEFINITION |
| :--- |
| I |
| TIME |
| I |
| XIN |

## Method:

VISIB is called both for visibility at midcourse time and for visibility at retro ignition. In either case, the spacecraft's position, X , is transformed into Earth's equator and prime meridian coordinates where the trackers' positions, $R_{t}$, reside. The slant range vector, $S=X-R_{t}$, is used to compute the elevation angle, E .

$$
\mathrm{El}=90^{\circ}-\cos ^{-1}\left(\frac{\mathrm{~S}^{\bullet} \mathrm{R}_{t}}{\mathrm{Sr})_{t}^{o}}\right.
$$

The KEL element corresponding to El is the integer part of

$$
\frac{\mathrm{E} 1}{10}+1
$$

if El is positive and zero otherwise. Thus, if the elevation is negative at the I-th tracker, KEL (I) $=0$. If elevation is between $0^{\circ}$ and $10^{\circ}, \operatorname{KEL}(\mathrm{I})=1$, if it is between $10^{\circ}$ and $20^{\circ}, \mathrm{KEL}(\mathrm{I})=2$, etc.

If the central body is not the Earth, the spacecraft's position is first translated to be Earth-centered before computing the slant range vector. In addition, occultation by the central body is checked. If the central body indeed occults the spacecraft from the I-th tracker, KEL ( I$)=0$, even if the elevation is positive. The criterion for occultation is as follows. Let $R$ be the spacecraft's position relative to JC and $r_{m}$ be the physical radius of JC. Then if

$$
\mathrm{R} \cdot \mathrm{~S} \geqq \mathrm{~s} \sqrt{\mathrm{r}^{2}-\mathrm{r}_{\mathrm{m}}^{2}}
$$

the spacecraft is occulted. Otherwise it is not. If the spacecraft is occulted while the state is Earth-centered (way out past the Moon), it would not be detected because the test would not be made.

## FUNCTION VNORM

Calling Sequence:
Purpose:

Common Blocks Required:
Subroutines Required:

## CALI VNORM (X, Y)

This function determines the magnitude of a vector and a unit vector in the same direction as the input vector.

None
None

Input / Output

| I/O | $\begin{aligned} & \text { SYMBOLIC } \\ & \text { NAME } \end{aligned}$ | DIMENSION | $\begin{aligned} & \text { COMMON } \\ & \text { BLOCK } \end{aligned}$ | DEFINITION |
| :---: | :---: | :---: | :---: | :---: |
| I | X | 3 | CALLING OPERAND | Input vector |
| 0 | Y | 3 | CALLING OPERAND | Unit vector along X |
| 0 | VNORM | 1 | $\begin{aligned} & \text { FUNCTION } \\ & \text { NAME } \\ & \hline \end{aligned}$ | Absolute value of X |

## Description:

The absolute value of the input vector, X , is determined from

$$
\mathrm{R}=\sqrt{\mathrm{x}_{1}^{2}+\mathrm{x}_{2}^{2}+\mathrm{x}_{3}^{2}}
$$

where $X_{i}, i=1,3$, are the components of $\bar{X}$. The components of the unit vector $\bar{Y}$ are determined from

$$
\mathrm{Y}_{\mathrm{i}}=\mathrm{X}_{\mathrm{i}} / \mathrm{R} \quad \dot{\mathrm{i}=1,3}
$$


[^0]:    *See Scarborough, James B, Numerical Mathematical Analysis, The Johns Hopkins Press, 1930.

[^1]:    *** See Table 3.3 of User's Manual

[^2]:    ＊＊＊See Table 3.3

