77-FM-18
Vol. III, Rev. 1

JSC-12564

Flight Design System-1
System Design Document

RECEIVED
NASA STI FACILITY

Processor Library
Book 2

(NASA-TM-80838) FLIGHT DESIGN SYSTEM 1. SYSTEM DESIGN DOCUMENT. PROCESSOR LIBRARY, BOOK 2 (NASA) 681 p

N80-70717

Unclas
00/12 42199

Mission Planning and Analysis Division

October 1979

NASA
National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

77FM18:II/III

JSC-12564

77-FM-18
Vol. III, Rev. 1

SHUTTLE PROGRAM

FLIGHT DESIGN SYSTEM-1
SYSTEM DESIGN

PROCESSOR LIBRARY
BOOK 2

By Software Development Branch

Approved: Elric N McHenry
Elric N. McHenry, Chief
Software Development Branch

Approved: Ronald L. Berry
Ronald L. Berry, Chief
Mission Planning and Analysis Division

Mission Planning and Analysis Division

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Houston, Texas

October 1979

PREFACE

The Flight Design System-1 (FDS-1) is a pilot project to evaluate current concepts and to determine the hardware/software capability that will be required for the operational era to support Shuttle flight planning. This software system is being implemented on a Hewlett-Packard 21MX computer with a Daconics documentation system and will provide terminal-based interactive flight planning capability.

The System Design Document (SDD) for FDS-1 is the specification for and description of this hardware/software facility. The SDD is logically organized into 10 published volumes. This organization is presented in the accompanying table. The material in the early volumes is primarily presented from the user's point of view, whereas the latter material is software-developer oriented. The SDD will be published by volumes over a period of time, and various volumes will be updated and republished during the development of FDS-1.

FDS-1 SYSTEM DESIGN DOCUMENT

Volume I	Introduction, Overview, and User Interface
*Volume II	Utility Processor Library
*Volume III	Application Processor Library
Volume IV	System Architecture and Executive
Volume V	Documentation Support
Volume VI	Standards
Volume VII	Utility Support Software
Volume VIII	Build and Delivery Procedures; Software Development, Debug, and System Build Aids
Volume IX	Executive Logic Flow - Program Design Language
Volume X	Document Change Request Procedure and Submittal Form

*Combined as one volume with title: Volume III FDS-1 Processor Library

ACKNOWLEDGMENT

This document was written and prepared by a team consisting of Johnson Space Center civil service and contractor personnel. The following organizations made significant contributions to this document:

International Business Machines, Inc.
Federal Systems Division
Mission Analysis and Engineering

Lockheed Electronics Company, Inc.
Systems and Services Division

National Aeronautics and Space Administration/Johnson Space Center
Data Systems and Analysis Directorate
Mission Planning and Analysis Division
    Flight Planning Branch
    Software Development Branch

The Mission Planning and Analysis Division (MPAD) directed the effort and, in addition, developed software and submitted completed draft writeups of some sections. MPAD wishes to acknowledge the excellent support received from all organizations involved in preparing this document.

CONTENTS

Section		Page

Section | Page

Sections | Page

Section		Page

Section | Page

Section Page

Section		Page

PHASE TABLE PRINT PROCESSOR (PTP)

Section | Page

TABLES

Tables | Page

Book 1

Table Page

FIGURES

Figure Page

Book 1

BASETIME INITIALIZATION PROCESSOR (BASTM)

CONSUMABLES ANALYSIS FOR SHUTTLE KITS PROCESSOR (CASKU)

COASTING STATE VECTOR PREDICTOR (INCLUDING AEG) PROCESSOR (COAST)

DATA BOX DISPLAY PROCESSOR (DBDSP)

Figure | Page

Figure | Page

Figure | Page

Figure | Page

Book 3

1.0 INTRODUCTION

This volume presents the complete program documentation for the processors in the Flight Design System-1 (FDS-1). Two types of processors exist in the processor library; utility processors and application processors. Utility processors provide a general capability that is not particularly associated with flight design, such as Data Box Display (DBDSP), Data Element Definition (DEFIN), or Conditional Execution Processors (IF, ELSE, ENDIF). Application processors provide capability that is directly related to accomplishing flight design, such as Ascent (ASENT), General Purpose Maneuver Processor (GPMP), Groundtrack (GTRAK), or Mission Plan Table Processor (MPTP).

Both types of processors are presented in this volume in alphabetical order; thus, volume II of the SDD documentation is contained in this volume and will not exist separately. The only documentation that exists in addition to this volume is the software listings and their comment cards.

Because of the magnitude of this volume, it is published in three books.

2.0 PROCESSOR LIBRARY

This volume contains the complete software documentation for all FDS-1 processors. The processors are paged alphabetically according to the processor names as shown below.

Book 1

Ascent Processor	ASENT
Data Assignment Processor	ASSGN
Attitude Tables Maintenance Processor	ATM
Basetime Initialization Processor	BASTM
Consumable Analysis for Shuttle Kits Processor	CASKU
Coasting State Vector Predictor (including AEG) Processor	COAST
Data Box Display Processor	DBDSP
Data Box Extractor Processor	DBEXT
Data Box Interpolator Processor	DBINT
Data Element Definition Processor	DEFIN
Sequence Iteration Processors (DO, ENDDO)	DO
Document Processor	DOC
Deorbit Target Module Processor	DTM

Book 2

Early Repeating Groundtrack Orbits Processor	ERGO
* Finite Burn Processor	FINBN
* Fixed Magnitude Two-Burn Processor	FM2BN
Flight Plan Display Processor	FPD
General Purpose Maneuver Processor	GPMP
Groundtrack Processor	GTRAK
Conditional Execution Processors (IF, ELSE, ENDIF)	IF

*To be supplied.

Invariant Element Ephemeris Processor	INVAR
CASKU and QUIKU Output Data Display Processor	LKOUT
Landing Opportunities Processor	LOPT
Load State Vector Processor	LSV
Launch Window Processor	LWP
* Maneuver Iterator Processor	MANIT
Matrix and Attitude Support Table Processor	MAST
Master Data Temporary Print Processor	MDTP
Mission Plan Table Processor	MPTP
* Node Definer Processor	NODE
Orbital Maneuver Processor (Rendezvous)	OMP
Book 3	
Input/Output Units Specification Processor	PHYDM
* Placement Longitude Processor	PLLON
Print State Vector Processor	PSV
Phase Table Print Processor	PTP
Quick Investigation of Consumables Kits Processor	QUIKU
Parametric Scan Processors (SCAN, ENDSC)	SCAN
Sunrise/Sunset Time Predictor Processor	SRSS
* Shuttle/SUS Burn Relative Motion Processor	SSBRM
Sun Synchronous Orbits Processor	SSYN
Station Contacts Processor	STACN
Summary Table Print Processor	STP
State Vector Units Conversion Processor	SVUCP

*To be supplied.

* State Vector Coordinate Transformation Processor	TFSV
Activity Timeline Processor	TMLNU

*To be supplied.

EARLY REPEATING GROUNDTRACK ORBITS PROCESSOR (ERGO)

1.0 PURPOSE

The Early Repeating Groundtrack Orbits Processor (ERGO) computes the semimajor axis required to achieve an orbit that will repeat its groundtracks in a specified number of days.

2.0 FUNCTIONAL DESCRIPTION

Given an initial circular altitude and the inclination as input, ERGO will, by holding the inclination constant and iterating on the altitude, compute a semimajor axis, eccentricity, and inclination combination that will produce a groundtrack repeating orbit. At execution, ERGO will perform one of three main functions. The route is chosen by the user through the use of the input parameter ERGID. With the first option ERGID = "NALT", ERGO will use the input altitude and inclination to determine the day (relative to the initial day zero) on which the groundtracks come closest to repeating (over the span of days designated by the user in the processor constants array), and the altitude will be incremented to have the groundtracks repeat on that day. Information about this altitude/inclination combination is then displayed to the user.

When ERGID = "SDAY", the input circular altitude will be incremented by a computed value in order to determine an orbital combination that will allow the groundtracks to repeat on the day supplied through the input parameter DAY. The resultant combination and its associated orbital parameters are displayed to the user.

When ERGID = "EDAY" is input, ERGO must first determine a value for DAY. ERGO will do this by finding the first day (from the base day zero) on which the input altitude and inclination combination comes within an input tolerance of repeating. Day will then be assigned that value, and ERGO will proceed in the same manner as in the case when ERGID = "SDAY" is input. The resultant altitude and inclination combination will be displayed to the user.

3.0 ASSUMPTIONS AND LIMITATIONS

Geosynchronous altitudes are the maximum altitudes that ERGO will consider. If an altitude larger than geosynchronous is input, ERGO will compute a geosynchronous orbit combination. The limit is determined by checking the initial nodal period to see if it is larger in value than the time required for the longitude of ascending node to recur.

ERGO does not generate a vehicle state vector.

The orbital rates parameters, apsidal precession, and nodal period, are calculated using invariant elements with the zonal harmonics incorporated. Similarly, the equation for nodal precession incorporates the zonal harmonics.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the ERGO processor interface table parameters is provided in table 4-I. GLOCON is a set of constants universal to all the processors that are maintained in the master data base. ERGO accesses this array for the constants that it requires. The default values are stored in !!GLCN.

ERGO accesses the SESCON array to obtain the session related constants that are generated by the user upon execution of the system utility processor BASTM. The default values are stored in !SESCN and are standard for all processors.

Specific constants and tolerances required by ERGO are maintained in PROCON. These parameters are used mainly for iteration tolerances and internal print analysis. Included in this set is the parameter to specify the maximum number of days over which ERGO will scan in order to find an early repeating groundtrack orbit. This parameter is input in PROCON(15).

ERGID is the identification code for the early repeating groundtracks options. If ERGID = "NALT", ERGO will find the day on which the groundtracks come closest to repeating, iterate on the altitude in order to have the groundtracks repeat on that day, and display the results. If ERGID = "SDAY", ERGO will iterate on the circular altitude required to have the groundtracks repeat on the input DAY. If ERGID = "EDAY", ERGO will determine the first day on which the input altitude/inclination combination comes within a specified tolerance (input in PROCON(9)) of repeating. The parameter DAY is assigned this value. ERGO will then iterate on the altitude required to have the groundtracks repeat on that day.

ALT is the circular altitude above a reference radius (the equatorial Earth radius for a Fischer ellipsoid that is input through GLOCON) that is used as the first iteration in the program. If ERGO determines that the groundtracks resulting from this altitude repeat within the input tolerance, then the input value for ALT is the final value. However, if the groundtracks do not repeat, then ALT is incremented to test a new altitude/inclination combination.

INC is the input inclination desired by the user. It is held constant throughout the run.

DAY is the day (from the initial day) on which it is desired to have the groundtracks repeat. If ERGID = "NALT", DAY is not used. If ERGID = "SDAY", DAY is to be input. If ERGID = "EDAY", DAY is determined by the program.

b. Interface table data array definitions - The definition of the input/output data arrays appearing in the ERGO interface table is provided in table 4-II.

c. Interface table data file definitions - None.

d. Processor solicited (prompted) inputs - None.

e. Processor displays and display parameter definition table - One main display is generated by ERGO. The format of the display is shown in table 4-III(a), an example of the display in table 4-III(b), and a definition of the display variables is provided in table 4-III(c). All the display parameters are in the user selected external units with the following exception: TSID, TNODE, the frequency of the recurring node, and nodal period are all in minutes. Also, the reference radius is in internal units. Note that R is the radius magnitude and A is semimajor axis.

In the groundtrack information portion of the display, CYCLES is the number of times the groundtrace cycles across the area bounded by consecutive groundtracks before the groundtracks repeat. CYCLES is nondimensional. The remaining six rows of the groundtrack information indicate the distance between various groundtracks. The first column of numbers is the central angle between the groundtracks. The second column of numbers is the distance between the groundtracks measured at the Equator along the Equator. The third column of numbers is the distance between the groundtracks measured at the reference latitude that is input in PROCON. Consecutive groundtracks are groundtracks that are one nodal period apart. Daily groundtracks are groundtracks that come closest to repeating after 1 day (e.g., the distance between daily groundtracks in a 1-day repeating orbit is 0° and the distance between daily groundtracks in a 2-day repeating orbit may be approximately 11°). Adjacent groundtracks are groundtracks that are closest to each other (once the groundtracks have repeated).

If ERGO reaches maximum iterations without converging on a solution, an error message display is shown, rather than the main display. The orbital parameters portion of the display is the same as on the main display. All the parameters are in external units except for TSID, TNODE, the frequency of the recurring node, and nodal period. They are in minutes. ITER is unitless, and the difference and the tolerance are both in orbits. The format of the display is shown in table 4-III(d), an example of the display in table 4-III(e), and a definition of the display parameters is provided in table 4-III(f).

f. Processor message table - In general, ERGO does not generate any messages to the user during execution. However, in the cases of certain types of user input errors, an error message will be displayed as information to the user. Depending on the severity of the error, execution may be terminated at that point. The messages the user may expect in those cases are shown in table 4-IV.

g. Interface table extended prompts - The processor extended prompts for each table parameter keyword are provided in table 4-V.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR ERGO

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
PROCON	AWA	Free	I	20	20		ERGO constants and tolerances
ERGID	AWA	6CH	I	3	1		Early repeating groundtracks option code "NALT" - Calculate the day on which the input altitude repeats (or an altitude close to the input altitude). "SDAY" - Calculate the altitude that repeats on a specified day (use the input altitude as the first guess). "EDAY" - Calculate the earliest day on which the orbit repeats within a repeat tolerance (use the input altitude as the first guess). Then calculate the altitude that repeats on that day.

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR ERGO

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
ALT	AWA	Real	I	2	1		Circular altitude above reference radius
INC	AWA	Real	I	2	1		Inclination
DAY	AWA	Intg	I	1	1	1	Day on which to have groundtracks repeat, relative to base day 0 (used when ERGID = "SDAY")

NOTES	CLASS	TYPE	USE
	AWA Disk	Free 2CH 72CH Intg 6CH Mix Real 18CH Symb Dubl 36CH	I = Input O = Output I/O = Input/Output

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR ERGO

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array, master data base element; see table 7.2-III of JSC IN 78-FM-60, vol. I, for definition of contents.
SESCON	(1) . . . (90)	!SESCN	Input/output session constants array; see table 7.2-II of JSC IN 78-FM-60, vol. I, for definition of contents.
PROCON	(1)	0	Output unit on which to write all print. The default value is a flag to use the user terminal. A positive value identifies the output unit.
	(2)	0	Not used
	(3)	0	Internal print flag option = 0 No internal print = 1 Internal print
	(4)	0	Not used
	(5)	20	Maximum iterations limit
	(6)	0	Not used
	(7)	0	Not used
	(8)	0	Not used
	(9)	3.0E-2	Orbit miss tolerance (used when ERGID = "EDAY".\ORBITS)
	(11)	6.0E-4	Eccentricity tolerance
	(13)	1.0E-3	Convergence tolerance
	(15)	30	Maximum number of days over which to look for repeating groundtracks
	(16)	0	Not used

TABLE 4-II.- Concluded

PROCESSOR ERGO

Array name	Index location	Default value	Definition
PROCON	(17)	30.	Reference latitude for groundtrack information, degrees
	(19)	250.	Radius tolerance, feet

TABLE 4-III.- PROCESSOR DISPLAYS AND DISPLAY PARAMETER DEFINITION TABLE

(a) Repeating groundtrack orbit display

PROCESSOR ERGO

```
                          REPEATING GROUNDTRACK ORBIT

****** INITIAL INPUT TO PROGRAM ******
ALTITUDE,INCLINATION      +XXXXXXXXXX.XX      +XXX.XXXXX
DAY,ERGID                         XXXXXX
ALTITUDE MEASURED ABOVE SPHERE WITH RADIUS OF +XXXXXXXX.X

                   ******** ORBITAL PARAMETERS ********
HA,HP                             +XXXXXX.XXXX    +XXXXXX.XXXX
INCLINATION                       +XXXX.XXXXXX
R,ALTITUDE,ECCENTRICITY           +XXXXXX.XXXX    +XXXXX.XXXXX    +.XXXXXXXXE+YY
A,MEAN A                          +XXXXXX.XXXX    +XXXXXX.XXXX
MEAN MOTION                       +XXX.XXXXXXX
OMEGAD,FREQ.RECUR.NODE            +X.XXXXXXXXX    +XXXXXX.XXXX
TSID,TNODE                        +XXXX.XXXXXX    +XXXX.XXXXXX

                 ******** GROUNDTRACK INFORMATION ********
CYCLES                                        XX
DIST. BETWEEN CONSEC. G.T.           +XXX.XXXXX  +XXXX.XXX  +XXXX.XXX
DIST. BETWEEN DAILY G.T.             +XXX.XXXXX  +XXXX.XXX  +XXXX.XXX
DIST. BETWEEN ADJACENT G.T.          +XXX.XXXXX  +XXXX.XXX  +XXXX.XXX
PERPEND. DIST. BET. CONCEC. G.T.     +XXX.XXXXX             +XXXX.XXX
PERPEND. DIST. BET. DAILY G.T.       +XXX.XXXXX             +XXXX.XXX
PERPEND DIST BET ADJACENT G T        +XXX XXXXX             +XXXX XXX
```

TABLE 4-III.- Continued

(a) Concluded

PROCESSOR <u>ERGO</u>

```
*****************************************************
******************* FINAL SOLUTION *******************
*****************************************************

 THE GROUNDTRACKS WILL REPEAT IN :         XXX ORBITS
                                  +XXXXXXX.XXX AAAA
                                  +XXXXX.XXXXX HR
                                  +XXXX.XXXXXX DAYS
                                  +        XXX DAYS
 FREQUENCY OF THE RECURRING NODE :+XXXXXX.XXXX MIN

MEAN ALTITUDE   INC.      MEAN A    OSCULATING A   ECCENTRICITY    NODAL PERIO

±XXXXX.XXXXX  +XXX.XX  +XXXXX.XXXX  +XXXXX.XXXX  .XXXXXXXXE+YY    +XXXX.XXXXX
```

TABLE 4-III.- Continued

(b) Example display

```
                          REPEATING GROUNDTRACK ORBIT

****** INITIAL INPUT TO PROGRAM ******
 ALTITUDE,INCLINATION                235.00        50.00000
 DAY,ERGID                           1             NALT
 ALTITUDE MEASURED ABOVE SPHERE WITH RADIUS OF  20925740.0

                    ******** ORBITAL PARAMETERS ********
 HA,HP                               234.0377       234.0377
 INCLINATION                         50.000000
 R,ALTITUDE,ECCENTRICITY             3677.9717      234.03769     .86605549E-03
 A,MEAN A                            3681.1602      3677.6689
 MEAN MOTION                         3.8612046
 OMEGAD,FREQ.RECUR.NODE             -.003537593     1416.0852
 TSID,TNODE                          93.235138      93.164566

                  ******** GROUNDTRACK INFORMATION ********
 CYCLES                                           1
 DIST. BETWEEN CONSEC. G.T.              23.68448   1423.627   1231.860
 DIST. BETWEEN DAILY G.T.                 4.73272    284.474    246.155
 DIST. BETWEEN ADJACENT G.T.              4.73272    284.474    246.155
 PERPEND. DIST. BET. CONSEC. G.T.        15.63065              812.970
 PERPEND. DIST. BET. DAILY G.T.           3.16964              164.857
 PERPEND. DIST. BET. ADJACENT G.T.        3.16964              164.857


*****************************************************
****************** FINAL SOLUTION *******************
*****************************************************

 THE GROUNDTRACKS WILL REPEAT IN :            76 ORBITS
                                       7080.425 MIN
                                      118.00708 HR
                                       4.916962 DAYS
                                              5 DAYS
 FREQUENCY OF THE RECURRING NODE :    1416.0852 MIN

MEAN ALTITUDE   INC.     MEAN A    OSCULATING A   ECCENTRICITY    NODAL PERIOD

   233.73486    50.00   3677.6689   3681.1602   .86605549E-03      93.164566
*****************************************************
```

TABLE 4-III.- Continued

(c) Display parameter definition for the repeating groundtrack orbit display

PROCESSOR ERGO

REPEATING GROUNDTRACK ORBIT	
Display parameter label	Parameter definition
ALTITUDE	Altitude measured above reference radius
HA	Height of apogee
HP	Height of perigee
R	Radius magnitude
A	Semimajor axis
MEAN A	Mean semimajor axis
ONEGAD	Nodal precession rate
TSID	Inertial period, min
TNODE	Nodal period, min
CYCLES	Number of times groundtrace will cross area bounded by two consecutive groundtracks before groundtracks repeat
G.T.	Groundtracks
DIST.	Distance
INC.	Inclination

TABLE 4-III.- Continued

(d) ERGO error message display

PROCESSOR ERGO

```
                            ERGO ERROR MESSAGE DISPLAY


DIFFERENCE BETWEEN THE CALCULATED VALUE AND THE
NEAREST INTEGER NUMBER OF ORBITS = +.XXXXXXXXXE+YY.
THE TOLERANCE = +.XXXXXXXXXE+YY.

****** INITIAL INPUT TO PROGRAM ******
ALTITUDE,INCLINATION       +XXXXXXXXXX.XX        +XXX.XXXXX
DAY,ERGID                           XXXXXX           AAAAAA
ALTITUDE MEASURED ABOVE SPHERE WITH RADIUS OF +XXXXXXXX.X

THE LAST ORBITAL PARAMETERS CALCULATED FOLLOW :

                  ******** ORBITAL PARAMETERS ********
HA,HP                       +XXXXXX.XXXX     +XXXXXX.XXXX
INCLINATION                 +XXXX.XXXXXX
R,ALTITUDE,ECCENTRICITY     +XXXXXX.XXXX     +XXXXX.XXXXX     +.XXXXXXXXXE+YY
A,MEAN A                    +XXXXXX.XXXX     +XXXXXX.XXXX
MEAN MOTION                 +XXX.XXXXXXX
OMEGAD,FREQ.RECUR.NODE      +X.XXXXXXXXX     +XXXXXX.XXXX
TSID,TNODE                  +XXXX.XXXXXX     +XXXX.XXXXXX
```

TABLE 4-III.- Continued

(e) Example error message display

```
                    ERGO ERROR MESSAGE DISPLAY

DIFFERENCE BETWEEN THE CALCULATED VALUE AND THE
NEAREST INTEGER NUMBER OF ORBITS = -.29769897E-01
THE TOLERANCE =  .99999993E-03

****** INITIAL INPUT TO PROGRAM ******
 ALTITUDE,INCLINATION              235.00        50.00000
 DAY,ERGID                           1           NALT
 ALTITUDE MEASURED ABOVE SPHERE WITH RADIUS OF  20925740.0

THE LAST ORBITAL PARAMETERS CALCULATED FOLLOW :

                    ******** ORBITAL PARAMETERS ********
 HA,HP                             234.0377        234.0377
 INCLINATION                      50.000000
 R,ALTITUDE,ECCENTRICITY          3677.9717       234.03769       .86545944E-03
 A,MEAN A                         3682.1211       3678.6313
 MEAN MOTION                      3.8612027
 OMEGAD,FREQ.RECUR.NODE         -.003534354       1416.1028
 TSID,TNODE                       93.235168       93.164658
```

TABLE 4-III.- Concluded

(f) Display parameter definition table for the ERGO error message table

PROCESSOR ERGO

ERGO ERROR MESSAGE DISPLAY	
Display parameter label	Parameter definition
ALTITUDE	Altitude measured above reference radius
HA	Height of apogee
HP	Height of perigee
R	Radius magnitude
A	Semimajor axis
MEAN A	Mean semimajor axis
OMEGAD	Nodal precession rate
TSID	Inertial period, min
TNODE	Nodal period, min

TABLE 4-IV.- PROCESSOR MESSAGE TABLE

PROCESSOR <u>ERGO</u>

MSG no.	Message ID block	Message text block and explanation
1	*ERGO*	*ERGIN* "AAAAAA" IS INVALID INPUT FOR BBBBBB. Meaning: The user has supplied incorrect character code. Severity: The processor will terminate, and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to correct erroneous code; then resume execution.
2	*ERGO*	*REPET* MAXIMUM ITERATIONS. ITER = "XX". Meaning: REPET did not converge on a solution for semimajor axis. Severity: Processor will print pertinent information and terminate. Action required by user: Check input data, including tolerances, for discrepancies.
3	*ERGO*	*EARLY* THE TOLERANCE FOR FINDING AN EARLY GROUNDTRACK - REPEATING DAY WAS NOT MET. Meaning: For ERGID = "EDAY", ERGO did not find a day on which groundtracks came within the user supplied tolerance of repeating. Severity: ERGO will set DAY = 1; error flag will be set; processor will continue. Action required by user: Check repeat tolerance, input in PROCON.

TABLE 4-V.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR ERGO

Processor name	Processor abstract prompt (maximum 256 characters)
ERGO	ERGO computes the semimajor axis required to achieve an orbit that will repeat its groundtracks in a specified number of days.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
GLOCON	Global constants array, master data base element,- normally defaulted to !!GLCN.
SESCON	Session constants array; normally defaulted to !SESCN.
PROCON	ERGO processor constants and tolerances
ERGID	Early repeating groundtracks option "NALT" - Calculate day on which input altitude repeats "SDAY" - Calculate altitude that repeats on specified day "EDAY" - Calculate altitude that repeats on earliest day within repeat tolerance
ALT	Circular altitude desired (above reference radius)
INC	Inclination desired
DAY	Day (relative to base day) on which to have groundtracks repeat

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

FINITE BURN PROCESSOR (FINBN)

TO BE SUPPLIED

FIXED MAGNITUDE TWO-BURN PROCESSOR (FM2BN)

TO BE SUPPLIED

FLIGHT PLAN DISPLAY PROCESSOR (FPD)

1.0 PURPOSE

The FPD processor presents graphically station contacts, sunrise/sunset, communications satellites rise/set, groundtrack, and events time-line information.

2.0 FUNCTIONAL DESCRIPTION

Based on optional flag settings (STAFLG, SUNFLG, TWFLG, and TEFLG), the FPD processor obtains the station contacts, sunrise/sunset, and TDRS data from disk files generated by the STACN, SRSS, AND TRTS processors. The FPD processor then formats and outputs these data to a TEKTRONIX terminal.

If groundtrack data is requested (GTFLG), the trajectory information may be input either through a trajectory DRDE (VECFIL), which had been generated by the INVAR processor, or by inputting a single invariant elements state vector (FPVEC). Based on this trajectory information, the processor computes and plots a history of latitude as a function of time.

If an events time line is requested, the processor obtains the time-line information through a disk file (TMLFIL) and correlates the events codes to Hollerith data by using another disk file (EVTFIL). The FPD outputs this information graphically.

3.0 ASSUMPTIONS AND LIMITATIONS

a. Orbital propagation using invariant elements does not reflect changes to the orbit due to atmospheric drag between the states stored in the trajectory DRDE. Periodic perturbations to the orbit are not considered. Only elliptical orbit formulation is provided.

b. If the station contacts, sunrise/sunset, or TDRS information is requested, then the STACN, SRSS, or TRTS processors must be run first to generate the desired data.

c. If the events time-line data are requested, then files that contain the events and time-line information would have to exist.

d. The processing may be initiated at terminals other than the TEKTRONIX. However, the user should ensure that there are no other users on the TEKTRONIX. In this case, a warning message and a prompt will be issued to the originating terminal before processing continues.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The interface table for the FPD processor is contained in table 4-I. The type of information displayed is controlled by the flags: STAFLG, SUNFLG, TWFLG, TEFLG, GTFLG, and TMLFLG. The data required to generate the display are contained in the disk files: STAFIL, SUNFIL, TWFIL, TEFIL, VECFIL, TMLFIL, and EVTFIL. Each file is not referenced unless the appropriate flag is turned on. Therefore, the files do not need to exist or contain valid data unless the flag is turned on. The parameters GETS and GETF define the time interval to be processed. The NOHRS parameter defines the number of hours to be displayed for each page of the display. The PAGFLG parameter defines whether the paging is done automatically or manually. If paging is done automatically, the processor automatically generates a hardcopy and continues processing on the next page. If paging is done manually, the processor issues a continue prompt to allow the user to determine whether a hardcopy is desired before processing the next page.

b. Interface table data array definitions - The interface table data arrays are contained in table 4-II.

c. Interface table data file definitions - The format of the FPD input files are contained in table 4-III.

d. Processor solicited (prompted) inputs - The format for the continue prompt is provided in table 4-IV.

e. Processor display and display parameter definition table - Tables 4-V and 4-VI contain the processor display and display parameter definition table.

f. Processor message table - The processor messages are contained in table 4-VII.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VIII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR FPD

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
STAFLG	AWA	2CH	I	1	1		Flag that indicates whether station contacts data are to be output "ON" = Output station contacts "OF" = No station contacts output
STAFIL	Disk	Free	I	--	—		Name of DRDE file that contains station contacts data
TWFLG	AWA	2CH	I	1	1		Flag that indicates whether TDRS west data are to be output "ON" = Output TDRS-west data "OF" = No TDRS-west data output
TWFIL	Disk	Real	I	--	—		Name of DRDE file that contains TDRS west data
TEFLG	AWA	2CH	I	1	1		Flag that indicates whether TDRS-east data is to be output "ON" = Output TDRS-east data "OF" = No TDRS-east data output
TEFIL	Disk	Real	I	—	—		Name of DRDE file that contains TDRS-east data
SUNFLG	AWA	2CH	I	1	1		Flag that indicates whether sunrise/sunset data is to be output "ON" = Output sunrise/sunset data "OF" = No sunrise/sunset output

NOTES	CLASS	TYPE	USE
	AWA Disk	Free 2CH 72CH Intg 6CH Mix Real 18CH Symb Dubl 36CH	I = Input O = Output I/O = Input/Output

TABLE 4-I.- Continued

PROCESSOR FPD

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
SUNFIL	Disk	Real	I	--	--		Name of DRDE file that contains sunrise/ sunset data
GTFLG	AWA	2CH	I	1	1		Flag that indicates whether groundtrack data is to be output "ON" = Output groundtrack data "OF" = No groundtrack output
IVEC	AWA	2CH	I	1	1		Vector source flag for groundtrack data "IN" = Single input vector (FRVEC) "DF" = Vector DRDE file as specified by VECFIL
FPVEC	AWA	Real	I	30	15		Input invariant elements state vector; required when IVEC = "IN"
VECFIL	Disk	Real	I	--	--		Name of invariant elements DRDE file; required when IVEC = "DF"
TMLFLG	AWA	2CH	I	1	1		Flag that indicates whether event time-line data are to be output "ON" = Output event time line "OF" = No event time-line output
TMLFIL	Disk	Free	I	-	-		Name of time-line DRDE; required when TMLFLG = "ON"

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR FPD

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
EVTFIL	AWA	6CH	I	3	1		Name of events file; required when TMLFLG = "ON"
GETS	AWA	Real	I	6	3		Start time (hr, min, sec) relative to base time
GETF	AWA	Real	I	6	3		Final time (hr, min, sec) relative to base time
NOHRS	AWA	Intg	I	1	1		Number of hours to be output per page
PAGFLG	AWA	2CH	I	1	1	"MA"	Flag to indicate whether paging is to occur automatically or manually "MA" = Manual "AU" = Automatic
GMTR	AWA	Real	I	2	1	!SESCN(11)	Reference time for GET computations
CFA	AWA	Real	I	2	1	!SESCN(27)	Conversion factor for angles
CFD	AWA	Real	I	2	1	!SESCN(29)	Conversion factor for distance
CFT	AWA	Real	I	2	1	!SESCN(31)	Conversion factor for time

NOTES:

CLASS	TYPE			USE
AWA	Free	2CH	72CH	I = Input
Disk	Intg	6CH	Mix	O = Output
	Real	18CH	Symb	I/O = Input/Output
	Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR FPD

Parameter keyword name	Class	Type	Use	Size	Array dimension (I, J)	Values stored in default interface table	Definition
ROMEGE	AWA	Real	I	2	1	!SESCN(81)	Rotation rate of Earth
PI	AWA	Real	I	2	1	!!GLCN(59)	Ratio of circle circumference to diameter
TWOPI	AWA	Real	I	2	1	!!GLCN(61)	Twice pi
DTWOPI	AWA	Dubl	I	3	1	!!GLCN(68)	Double precision TWOPI
AMIN	AWA	Real	I	2	1	!!GLCN(97)	Seconds per minute
AHOUR	AWA	Real	I	2	1	!!GLCN(99)	Seconds per hour
PROCON	AWA	Free	I	6	6		FPD constants array

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR FPD

Array name	Index location	Default value	Definition
FPVEC	1-15		Invariant elements vector (standard state vector format)
GETS	1		Start time relative to reference time (hours)
	2		Start time relative to reference time (minutes)
	3		Start time relative to reference time (seconds)
GETF	1		Final time relative to reference time (hours)
	2		Final time relative to reference time (minutes)
	3		Final time relative to reference time (seconds)
PROCON	1	0	Debug print flag = 0 No print = 1 Debug print
	2	0	Output destination flag = 0 Terminal Positive number = LU for output device
	3	31	Logical unit for plotting output
	4	20	Cartridge reference number for data files
	5-6	225.	Step size (seconds) to be used for groundtrack output

TABLE 4-III.- INTERFACE TABLE DATA FILE DEFINITIONS

PROCESSOR FPD

DRDE DATA FILE STAFIL

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file
	4-6	Interface table variable creating file
	7-9	Processor last changing file
	10-12	Interface table variable last changing file
2-N	1-3	Station contact site identification
	4	Orbit number
	5-6	GMT of acquisition of signal
	7-8	GMT of loss of signal
	9-10	Maximum elevation angle
	11-12	Azimuth at acquisition of signal
	13-14	Range at acquisition of signal
	15-16	Minimum range

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE SUNFIL

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file
	4-6	Interface table variable creating file
	7-9	Processor last changing file
	10-12	Interface table variable last changing file
2-N	1	Orbit number of terminator set
	2-3	GMT of terminator set
	4-5	Latitude of terminator set
	6-7	Longitude of terminator set
	8-9	Altitude of terminator set
	10-11	LVLH pitch angle to Sun
	12-13	LVLH yaw angle to Sun
	14	Orbit number of effective sunset
	15-16	GMT of effective sunset
	17-18	Latitude of effective sunset
	19-20	Longitude of effective sunset
	21-22	Altitude of effective sunset
	23-24	LVLH pitch angle to Sun
	25-26	LVLH yaw angle to Sun
	27	Orbit number of sunset
	28-29	GMT of sunset

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE SUNFIL - Continued

Record number	Integer word allocations	Content and definition
2-N (cont'd.)	30-31	Latitude of sunset
	32-33	Longitude of sunset
	34-35	Altitude of sunset
	36-37	LVLH pitch angle to Sun
	38-39	LVLH yaw angle to Sun
	40	Orbit number of orbital midnight
	41-42	GMT of midnight
	43-44	Latitude of orbital midnight
	45-46	Longitude of orbital midnight
	47-48	Altitude of orbital midnight
	49-50	LVLH pitch angle to Sun
	51-52	LVLH yaw angle to Sun
	53	Orbit number of sunrise
	54-55	GMT of sunrise
	56-57	Latitude of sunrise
	58-59	Longitude of sunrise
	60-61	Altitude of sunrise
	62-63	LVLH pitch angle to Sun
	64-65	LVLH yaw angle to Sun
	66	Orbit number of effective sunrise

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE SUNFIL - Concluded

Record number	Integer word allocations	Content and definition
2-N (conc'd.)	67-68	GMT of effective sunrise
	69-70	Latitude of effective sunrise
	71-72	Longitude of effective sunrise
	73-74	Altitude of effective sunrise
	75-76	LVLH pitch angle to Sun
	77-78	LVLH yaw angle to Sun
	79	Orbit number of terminator rise
	80-81	GMT of terminator rise
	82-83	Latitude of terminator rise
	84-85	Longitude of terminator rise
	86-87	Altitude of terminator rise
	88-89	LVLH pitch angle to Sun
	90-91	LVLH yaw angle to Sun
	92	Orbit number of orbital noon
	93-94	GMT of orbital noon
	95-96	Latitude of orbital noon
	97-98	Longitude of orbital noon
	99-100	Altitude of orbital noon
	101-102	LVLH pitch angle to Sun
	103-104	LVLH yaw angle to Sun

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE TEFIL/TWFIL

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file
	4-6	Interface table variable creating file
	7-9	Processor last changing file
	10-12	Interface table variable last changing file
2-N	1-2	GMT of TDRS rise time, sec
	3-4	GMT of TDRS set time, sec

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE VECFIL

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file
	4-6	Interface table variable creating file
	7-9	Processor last changing file
	10-12	Interface table variable last changing file
2-N	1-60	Position/velocity phase table column; see sec. 1.2 in vol. VI for form and content; each record con tains one column of an invariant element state phase table

TABLE 4-III.- Continued

PROCESSOR FPD

DRDE DATA FILE TMLFIL

Record number	Integer word allocations	Content and definition
1	1-3	"TMLNU" - Name of the processor that created the file
	4-6	NEWFIL - Interface table variable name through which the file was created
	7-9	"TMLNU" - Name of the processor that last updated the file
	10-12	NEWFIL - Interface table variable name through which the file was changed
2-N	1	IACT(I) - Activity number
	2-3	TIM(I,1) - GET of activity start time, hr
	4-5	TIM(I,2) - GET of activity stop time, hr

TABLE 4-III.- Concluded

PROCESSOR FPD

DRDE DATA FILE EVTFIL

Record number	Integer word allocations	Content and definition
1-N	1-2	ASCII code for event number
	3-6	ASCII code for event identifier

TABLE 4-IV.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR FPD

Prompt	Meaning	Valid responses
CONTINUE?:	Determine if processing should be continued. Used after warning message 14 and when paging is done manually	Space (b̸) - Indicates that processing should continue. Any other character indicates that processing should terminate.

TABLE 4-V.- PROCESSOR DISPLAY FORMAT

PROCESSOR FPD

```
                              FLIGHT PLAN DISPLAY
STAFIL-        TWFIL-       TEFIL-      SUNFIL-        VECFIL-        TMLFIL-
  GET   | GSTDN TDRS      SUN    LATITUDE              EVENTS        |  GET
  0 00  | ELEV                                                       |  0 00

  1 00                                                                  1 00

  2 00                                                                  2 00

  3 00                                                                  3 00

  4 00                                                                  4 00

  5 00                                                                  5 00

  6 00                                                                  6 00
CONT?!
                                                                   PAGE # 1
```

TABLE 4-VI.- DISPLAY PARAMETER DEFINITION TABLE FOR THE FLIGHT PLAN DISPLAY

PROCESSOR FPD

Display name FLIGHT PLAN DISPLAY

Display parameter label	Parameter definition
STAFIL	Name of station contacts input file; blank if STAFLG is off.
TWFIL	Name of TDRS-west input file; blank if TWFLG is off.
TEFIL	Name of TDRS-east input file; blank if TEFLG is off.
SUNFIL	Name of sunrise/sunset input file; blank if SUNFLG is off.
VECFIL	Name of invariant elements input file; blank if not used.
TMLFIL	Name of time-line input file; blank if TMLFLG is off.
GET	GET for plot labels.
GSTN ELEV	Times of station contacts and the station name are plotted as a function of the maximum elevation angle.
TDRS	Times are plotted from TDRS-rise to TDRS-set for TDRS-west and TDRS-east.
SUN	Times are plotted from sunset to sunrise.
LATITUDE	A time history of latitude is plotted.
EVENTS	A time history of the events time line is plotted.
PAGE #	Current display page number.

TABLE 4-VII.- PROCESSOR MESSAGE TABLE

PROCESSOR FPD

MSG no.	Message ID block	Message text block and explanation
1	*FPD*	"VECFIL OPEN FILE ERROR=", IIII, "NAME=", AAAAAA Meaning: Error occurred when attempt was made to open file. Severity: Processor terminates. Action required by user: Check VECFIL name for validity.
2	*FPD*	"STAFIL OPEN FILE ERROR=", IIII, "NAME=", AAAAAA Meaning: Error occurred when attempt was made to open file. Severity: Processor terminates. Action required by user: Check STAFIL parameter for validity.
3	*FPD*	"STAFIL READ ERROR=", IIII Meaning: Error occurred when attempt was made to read file. Severity: Processor terminates. Action required by user: Consult SDB representative.
4	*FPD*	"SUNFIL OPEN ERROR=", IIII "NAME=", AAAAAA Meaning: Error occurred when attempt was made to open SUNFIL. Severity: Processor terminates. Action required by user: Check SUNFIL parameter for validity.
5	*FPD*	"SUNFIL READ ERROR=", IIII Meaning: Error occurred when attempt was made to read SUNFIL. Severity: Processor terminates. Action required by user: Consult SDB representative.
6	*FPD*	"TEFIL OPEN ERROR=", IIII Meaning: Error occurred when attempt was made to open TEFIL. Severity: Processor terminates. Action required by user: Check TEFIL parameter for validity.

TABLE 4-VII.- Continued

PROCESSOR FPD

MSG no.	Message ID block	Message text block and explanation
7	*FPD*	"TWFIL OPEN ERROR=", IIII Meaning: Error occurred when attempt was made to open TWFIL. Severity: Processor terminates. Action required by user: Check TWFIL parameter for validity.
8	*FPD*	"TDRS FILE READ ERROR=", IIII Meaning: Error occurred when attempt was made to read TEFIL or TWFIL. Severity: Processor terminates. Action required by user: Consult SDB representative.
9	*FPD*	"VECFIL READ FILE ERROR=", IIII Meaning: Error occurred when attempt was made to read VECFIL. Severity: Processor terminates. Action required by user: Check VECFIL for validity of contents.
10	*FPD*	"TMLFIL OPEN ERROR=", IIII Meaning: Error occurred when attempt was made to open TMLFIL. Severity: Processor terminates. Action required by user: Check TMLFIL parameter for validity.
11	*FPD*	"TMLFIL READ FILE ERROR=", IIII Meaning: Error occurred when attempt was made to read TMLFIL. Severity: Processor terminates. Action required by user: Check TMLFIL for validity of contents.
12	*FPD*	"EVTFIL OPEN ERROR=", IIII Meaning: Error occurred when attempt was made to open EVTFIL. Severity: Processor terminates. Action required by user: Check EVTFIL parameter for validity.

TABLE 4-VII.- Concluded

PROCESSOR FPD

MSG no.	Message ID block	Message text block and explanation
13	*FPD*	"EVTFIL READ ERROR=", IIII Meaning: Error occurred when attempt was made to read EVTFIL. Severity: Processor terminates. Action required by user: Check EVTFIL for validity of contents.
14	*FPD*	"PLOT OUTPUT WILL BE TO TEKTRONIX" Meaning: The plot output for the FPD must go to the TEKTRONIX terminal; however, processing may be initiated from another terminal. Severity: Warning. Action required by user: The user should ensure that there are no current users of the TEKTRONIX terminal. If the user desires to continue, the response to the continue prompt should be b̸; any other response will result in termination of the processing.

TABLE 4-VIII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR FPD

Processor name	Processor abstract prompt (maximum 256 characters)
FPD	FPD presents (graphically) station contacts, Sunrise/Sunsets, TDRS rise/set, groundtracks, and event time-line information.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
	Use definitions on interface table definitions (table 4-I), except for the following:
PROCON	Processor constants (1) Debug (2) Logical unit for print output (3) Logical unit for plot output (4) Cartridge reference for data files (5) Stepsize for groundtracks output (seconds)

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

GENERAL PURPOSE MANEUVER PROCESSOR (GPMP)

1.0 PURPOSE

The GPMP computes a single vehicle impulsive maneuver that is initiated at a specified maneuver point in the vehicle orbit and is targeted to a specified final orbit condition.

2.0 FUNCTIONAL DESCRIPTION

The GPMP has the capability of processing 12 different maneuvers. The maneuvers are: input delta-V, plane change, node shift, circularization, circularization with plane change, circularization with node shift, height maneuver, height maneuver with plane change, height maneuver with node shift, line of apsides shift, line of apsides shift while maintaining HA and HP, and change HA and HP. These maneuvers can be performed at a variety of specified locations such as latitude, longitude, time, apogee, perigee, height, radius, argument of latitude or optimum time. Given the initial state vector (position, velocity, time, CD, area, and weight), a maneuver option, maneuver location and value, and threshold time, the GPMP will execute a single vehicle impulsive burn to effect the specified maneuver.

After the user-provided input is received by the GPMP and converted to proper form, a routine to search for a maneuver time associated with a particular location condition is used. After the time is computed, the input vector is updated to that time. Then the maneuver targeting routine is called to compute the delta-V associated with the maneuver. Following the computation of the delta-V required for the maneuver, the inertial delta-V components will be applied to the preburn state vector to obtain the postburn vector. This vector, plus associated orbital quantities, and maneuver parameters will then be displayed to the user. This vector and some displayed quantities are also stored by the processor so that any other processor may use them.

3.0 ASSUMPTIONS AND LIMITATIONS

The GPMP is intended to include as many impulsive maneuver types (other than rendezvous maneuvers) as possible that are commonly used by the mission design engineer. However, a particular Shuttle mission may require a maneuver type not present in the targeting options. Sometimes, a combination of maneuvers from the GPMP list will suffice.

The logic of the GPMP requires that the specified maneuver threshold time be equal to, or greater than, the input vector time. This means, in effect, that no vector will be propagated backwards in time. Specifically, this means that the input vector will always be updated forward in time from input time to computed or input maneuver time.

When the input delta-V maneuver is selected, the delta-V is positive, and the angle rotation sequence is yaw, then pitch.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the GPMP processor interface table parameters is provided in table 4-I. GLOCON is a set of constants universal to all the processors that will be maintained in the master data base. The GPMP will access this array for the constants it requires. The default values will be stored in !!GLCN.

The GPMP will access the SESCON array to obtain the session-related constants generated by the user on execution of the system utility processor BASTM. The default values will be stored in !SESCN and will be standard for all processors.

Specific constants and tolerances required by the GPMP will be maintained through default values stored in !PROCN. These parameters are used mainly for iteration tolerances and internal print analysis.

CVEC is the position/velocity input state vector of the maneuvering vehicle. This vector is updated to the time of the burn by the GPMP. All internal GPMP calculations involving the state vector use the true Equator and Greenwich meridian of epoch coordinate system (TEG) and the Cartesian element set. Thus, if the input vector is not in this element set and coordinate system the vector conversion processor must be used prior to using the GPMP in order to place the state vector in the acceptable coordinate set.

The following seven parameters completely define to the GPMP the type of maneuver to be calculated. MANID is the maneuver identification code to describe the maneuver option. MNSTOP is the maneuver location code, and STOPVL is the maneuver location stop condition. Thus, these first three parameters describe the type, place, and time of the chosen maneuver. THRT is the threshold GET (or maneuver time, if the maneuver time is input) at which the GPMP begins searching for the maneuver stop condition. MANOPT, APSOPT, and MNSPEC are not necessarily used in each case. MANOPT is a maneuver information option used when MANID = "DELV," "HITE," "HTPC," or "HTNS." APSOPT is a line of apsides option used when MANID = "HITE," "HTPC," "HTNS," or "SHKP." MNSPEC is an array accepting up to three parameters that complete the description of the chosen maneuver.

A GPMP input matrix is provided in table 4-II as additional information describing the input options.

The following explanation may be helpful. The letters in parentheses indicate the type of input required from the user: "6CH" = six characters; "R" = real number; "3R" = three real numbers separated by commas. The character "b̸" indicates that the parameter is not used for that particular option. However, a value other than a blank must be entered. The character "R" indicates that a real number must be entered and is used. The word "SAME" indicates that the same options are available as listed under the MANID = "DELV" option.

Finally SVPROP is the state vector propagation flag. Either a conic, two-body, propagation or the analytic ephemeris generator (AEG) may be chosen.

An example interface table response for the previous seven parameters and SVPROP is given in figure 4-1.

PVTAB is the position/velocity state vector phase table output quantities. The initial input vector, the preburn vector at burn initiation time, and the postburn vector at burn initiation time (impulsive burn) will be output to the table in the TEG coordinate system and the Cartesian element set. The vector times are in Greenwich mean time.

SUMTAB is the output summary table data array. Information placed there by the GPMP is available for recall at a later time in the user session. By executing SMPRT, the information will be listed in tabular form.

b. Interface table data array definitions - The definition of the input/output data arrays appearing in the GPMP interface table is provided in table 4-III.

c. Interface table data file definitions - None

d. Processor solicited (prompted) inputs - The processor solicited (user prompts) are provided in table 4-IV. These prompts allow the user to cause a pause in the execution of the processor while the present results are considered.

e. Processor displays and display parameter definition tables - One display is generated by the GPMP. The format of this display is shown in table 4-V, and a definition of the display variables is provided in table 4-VI. All the display parameters are in the user selected external units with the following exceptions. GMTI, GETI, GETA, and GETP are in hours, minutes, seconds. DELTB MAN is in seconds, and PERIOD is in minutes.

f. Processor message table - In general, the GPMP does not generate any messages to the user during execution. However, in the cases of certain types of user input errors, an error message may be displayed as a warning to the user. The messages the user may expect in those cases are shown in table 4-VII.

One set of messages not listed in the table are those generated by the COAST processor.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VIII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR GPMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array, master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
PROCON	AWA	Free	I	20	20	--	GPMP constants and tolerances
BVEC	AWA	Real	I	30	15	--	Position/velocity input state vector
THRUST	AWA	Real	I	2	1	--	Vehicle thrust
ISP	AWA	Real	I	2	1	--	Vehicle specific impulse, seconds
ENGID	AWA	2CH	I	1	1		Engine identification code: "SR" - Solid rocket motors "ME" - Main engine "OM" - Orbital maneuvering system "RA" - Reaction control system - AFT "RF" - Reaction control system - FWD "US" - Shuttle upper stage
MANID	AWA	6CH	I	3	1	--	Maneuver identification code "DELV" - input ΔV "PLCH" - plane change "NOSH" - node shift "CIRC" - circularization "CRPC" - circularization with plane change "CRNS" - circularization with node shift

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR GPMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
MANID	AWA	6CH	I	3	1	--	"HITE" - height "HTPC" - height with plane change "HTNS" - height with node shift "SHFT" - shift line of apsides "SHKP" - shift line of apsides, keep same apogee/perigee "HAHP" - change apogee/perigee
MNSTOP	AWA	6CH	I	3	1		Maneuver location code "CUR" - use current vector "APS" - N^{th} apsidal crossing "TIM" - update state to a time "ALT" - search for altitude "LAT" - search for latitude "LON" - search for longitude "RAD" - search for radius "APO" - search for N^{th} apogee "PER" - search for N^{th} perigee "OPT" - optimum node shift point "ARG" - argument of latitude
STOPVL	AWA	Real	I	2	1		Value of maneuver location stop condition. When MNSTOP = "CUR"; STOPVL = N/A = "APS"; = N^{th} crossing = "TIM"; = N/A = "ALT"; = altitude = "LAT"; = latitude = "LON"; = longitude = "RAD"; = radius = "APO"; = N^{th} apogee = "PER"; = N^{th} perigee

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR GPMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
STOPVL	AWA	Real	I	2	1		= "OPT"; = N/A = "ARG"; = argument of latitude
THRT	AWA	Real	I	6	3		Threshold ground elapsed time to begin search for MNSTOP; or maneuver ground elapsed time; hr, min, sec
MANOPT	AWA	6CH	I	3	1		Maneuver option = "LVLH" - Input ΔVx,ΔVy,ΔVz in MNSPEC = "DVYP" - Input ΔV, yaw pitch in MNSPEC for MANID = "DELV". = "H" - Input altitude in MNSPEC = "DH" - Input delta altitude in MNSPEC for MANID = "HITE", "HTPC", "HTNS"
APSOPT	AWA	6CH	I	3	1		Line of apsides option = "NO" - Do not create apsis at maneuver point = "YES" - Create apsis at maneuver point for MANID = "HITE", "HTPC", "HTNS" = "PER" - Perform maneuver near perigee = "APO" - Perform maneuver near apogee for MANID = "SHKP"
MNSPEC	AWA	Real	I	6	3		Maneuver specifications

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR GPMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
SVPROP	AWA	6CH	I	3	1		State vector propagator selection flag. = "CON"; CONIC = "AEG"; analytic ephemeris generator
PVTAB	AWA	Real	0	180	30,3		Position/velocity phase table. Three vectors: (1) initial, (2) ignition, (3) burnout
MNCODE	AWA	6CH	0	6	2		Maneuver identification code and maneuver location code
SUMTAB	AWA	Free	0	128	8,16		Output summary table

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	0 = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- GPMP MANEUVER INPUT MATRIX

MANID	MNSTOP	STOPVL	THRT	MANOPT	APSOPT	MNSPEC
(6CH)	(6CH)	(R)	(3R)	(6CH)	(6CH)	(3R) (Definition of input required)
DELV	CUR	b̸	b̸	LVLH DVYP	b̸ b̸	DV_x, DV_y, DV_z DV, YAW, PITCH
	APS	R	R,R,R			
	TIM	b̸	R,R,R			
	ALT	R	R,R,R			
	LAT	R	R,R,R			
	LON	R	R,R,R			
	RAD	R	R,R,R			
	APO	R	R,R,R			
	PER	R	R,R,R			
	ARG	R	R,R,R			
PLCH	same	same	same	b̸	b̸	Wedge angle
NOSH	same	same	same	b̸	b̸	Delta node
	OPT	b̸	R,R,R	b̸	b̸	Delta node
CIRC	same	same	same	b̸	b̸	None req.
CRPC	same	same	same	b̸	b̸	Wedge angle
CRNS	same	same	same	b̸	b̸	Delta node
	OPT	b̸	R,R,R	b̸	b̸	Delta node
HITE	same	same	same	H	No Yes	Altitude
				DH	No Yes	Delta altitude

TABLE 4-II.- Concluded

MANID	MNSTOP	STOPVL	THRT	MANOPT	APSOPT	MNSPEC
(6CH)	(6CH)	(R)	(3R)	(6CH)	(6CH)	(3R)
HTPC	same	same	same	H	No / Yes	Altitude and wedge angle
				DH	No / Yes	Delta altitude and wedge angle
HTNS	same	same	same	H	No / Yes	Altitude and delta node
				DH	No / Yes	Delta altitude and delta node
	OPT	ƀ	R,R,R	same	same	same
SHFT	same	same	same	ƀ	ƀ	Delta true anomaly
SHKP	set internally	set internally	R,R,R	ƀ	PER / APO	Delta true anomaly
HAHP	same	same	same	ƀ	ƀ	Height of apogee and height of perigee

```
\MANID   = :  "DELV"
\MNSTOP  = :  "CUR"
\STOPVL  = :  0.
\THRT    = :  0., 0., 0.
\MANOPT  = :  "LVLH"
\MPSOPT  = :  "ƀ"
\MNSPEC  = :  100., 0., 10.
\SVPROP  = :  "AEG"
```

Figure 4-1.- Example response.

TABLE 4-III.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR GPMP

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array, master data base element; see table 7.2-III of JSC IN 78-FM-60, volume 1 for definition of contents.
SESCON	(1) . . . (90)	!SESCN	Input/output session constants array; see table 7.2-II of JSC IN 78-FM-60, volume I for definition of contents.
PROCON	(1)	0	Output unit on which to write all print; the default value is a flag to use the user terminal. A positive value identifies the output unit.
	(2)	0	Not used
	(3)	0	Internal print flag option = 0; no internal print = 1; internal print
	(4)	0	Not used
	(5)	20	Maximum iterations limit
	(6)	0	Not used
	(7)	-25.0	First guess change in value of independent variable. Used in subroutine ITERV.
	(9)	2.0E-4	Transfer angle tolerance
	(11)	6.0E-4	Eccentricity tolerance
	(13)	2.0E-4	Convergence tolerance for travel angle
	(15)	0	Independent variable guess switch = 0; no independent variable input = 1; a guess for independent variable is input
	(16)	0	Not used
	(17)	0.	An initial guess for the independent variable
	(19)	250.	Radius tolerance
BVEC	(1) . . . (15)		Position/velocity input state vector. See figure 7.3-2 of JSC IN 78-FM-60, volume I for definition of contents

TABLE 4-III.- Continued

PROCESSOR GPMP

Array name	Index location	Default value	Definition
MNSPEC			Maneuver specifications
			For MANID = "DELV" and MANOPT = "LVLH":
	(1)	0.	DV_x - the LVLH delta velocity component in the X direction
	(2)	0.	DV_y - the LVLH delta velocity component in the Y direction
	(3)	0.	DV_z - the LVLH delta velocity component in the Z direction
			For MANID = "DELV" and MANOPT = "DVYP":
	(1)	0.	DV - the total delta velocity magnitude
	(2)	0.	Yaw - the LVLH yaw angle
	(3)	0.	Pitch - the LVLH pitch angle
			For MANID = "PLCH" or "CRPC":
	(1)	0.	WEDANG - wedge angle for plane change
	(2) - (3)	0.	Not used
			For MANID = "NOSH" or "CRNS":
	(1)	0.	DELNOD - node shift delta node angle
	(2) - (3)	0.	Not used
			For MANID = "CIRC":
	(1) - (3)	0.	Not used
			For MANID = "HITE" and MANOPT = "H":
	(1)	0.	H - altitude desired 180 degrees from the maneuver point
	(2) - (3)	0.	Not used
			For MANID = "HITE" and MANOPT = "DH":
	(1)	0.	DELH - height difference desired 180 degrees from the maneuver point
	(2) - (3)	0.	Not used
			For MANID = "HTPC" and MANOPT = "H":
	(1)	0.	H - altitude desired 180 degrees from the maneuver point
	(2)	0.	WEDANG - plane change wedge angle
	(3)	0.	Not used
			For MANID = "HTPC" and MANOPT = "DH":
	(1)	0.	DH - height difference desired 180 degrees from maneuver point
	(2)	0.	WEDANG - plane change wedge angle
	(3)	0.	Not used

TABLE 4-III.- Continued

PROCESSOR GPMP

Array name	Index location	Default value	Definition
MNSPEC (cont'd)			For MANID = "HTNS" and MANOPT = "H":
	(1)	0.	H - altitude desired 180 degrees from the maneuver point
	(2)	0.	DELNOD - delta node shift angle
	(3)	0.	Not used
			For MANID = "HTNS" and MANOPT = "DH":
	(1)	0.	DH - height difference desired 180 degrees from maneuver point
	(2)	0.	DELNOD - delta node shift angle
	(3)	0.	Not used
			For MANID = "SHFT" or "SHKP"
	(1)	0.	DELF - desired amount of delta true anomaly apsidal shift
	(2) - (3)	0.	Not used
			For MANID = "HAHP"
	(1)	0.	HA - height of apogee
	(2)	0.	HP - height of perigee
	(3)	0.	Not used
THRT			Threshold ground elapsed time to begin search for MNSTOP; or maneuver time
	(1)	0.	Hours
	(2)	0.	Minutes
	(3)	0.	Seconds
PVTAB			Position/velocity phase table; see figure 7.3-13 of JSC IN 78-FM-60, volume I for definition of contents. Position/velocity state and propagation data at:
	(1,1) . . . (30,1)		1. Initial time
	(1,2) . . . (30,2)		2. Ignition time

TABLE 4-III.- Concluded

PROCESSOR GPMP

Array name	Index location	Default value	Definition
PVTAB (cont'd)	(1,3)		3. Burnout time
	.		
	.		
	.		
	(30,3)		
SUMTAB	(1,1)		ERROR = Reserved error flag for summary table output processing
	(1,2)		GMTI = Greenwich mean time of ignition
	(1,3)		GETI = Ground elapsed time of ignition
	(1,4)		DVMAN = Total delta velocity magnitude
	(1,5)		DTBMAN = Total burn duration, in sec
	(1,6)		PITMAN = LVLH pitch angle of maneuver
	(1,7)		YAWMAN = LVLH yaw angle of maneuver
	(1,8)		DVX = LVLH delta velocity component in X direction
	(1,9)		DVY = LVLH delta velocity component in Y direction
	(1,10)		DVZ = LVLH delta velocity component in Z direction
	(1,11)		I = Resultant inclination
	(1,12)		HA = Resultant height of apogee
	(1,13)		HP = Resultant height of perigee
	(1,14)		HMAN = Altitude at burn initiation
	(1,15)		WHTI = Initial gross weight
	(1,16)		WHTF = Final gross weight

TABLE 4-IV.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR GPMP

Prompt	Meaning	Valid responses
"GPMP DISPLAY:"	Execution of GPMP is completed; ready to show display	= ƀƀ; Continue; no display shown. = DS; General purpose maneuver table display.

TABLE 4-V.- PROCESSOR DISPLAY TABLE

PROCESSOR GPMP

```
                   GENERAL PURPOSE MANEUVER TABLE
    MANID*         *       *
                                      WEIGHT I              XXXXXXX.X
    HA          XXXXXXXXX.XX          DELTA WHT             XXXXXXX.X
    GET A        XXX XX XX.X          WEIGHT F              XXXXXXX.X
    LONG A        +XXX.XXXXX
    DECL A         +XX.XXXXX          GMTI                  XXXXXXX.X
                                      GETI                  XXXXXXX.X
    HP          XXXXXXXXX.XX
    GET P        XXX XX XX.X          DEL V MAN               XXXXX.X
    LONG P        +XXX.XXXXX          DEL TB MAN              XXXXX.X
    DECL P         +XX.XXXXX          PIT MAN                   +XX.X
                                      YAW MAN                  +XXX.X
    V             XXXXXX.XXX          H MAN             XXXXXXXXX.XXX
    FPA           +X.XXXXXXX
    AZ            +XXX.XXXXX          A                 XXXXXXXXX.XX
    R            XXXXXXXX.XX          E                    .XXXXXXXX
    LONG          +XXX.XXXXX          I                     XXX.XXX
    DECL           +XX.XXXXX          TRUE A             +XXX.XXXXX
    GETV       .XXXXXXXXE+XX
                                      DEL I                 +XXX.XX
    DVX           +XXXXX.XXX          DEL NODE              +XXX.XX
    DVY           +XXXXX.XXX          DEL G                 +XXX.XX
    DVZ           +XXXXX.XXX          PERIOD            XXXXXX.XXXX
```

TABLE 4-VI.- DISPLAY PARAMETER DEFINITIONS TABLE

PROCESSOR GPMP

GENERAL PURPOSE MANEUVER TABLE

Display parameter label	Parameter definition
MANID	Maneuver identification and type of stop condition code
HA	Resulting height of apogee
GET A	Ground elapsed time of arrival at resultant apogee, in hr., min., sec.
LONG A	Longitude of resultant apogee
DECL A	Declination of resultant apogee
HP	Resulting height of perigee
GET P	Ground elapsed time to arrival at resultant perigee, in hr., min., sec.
LONG P	Longitude of resultant perigee
DECL P	Declination of resultant perigee
V	Resultant velocity
FPA	Resultant flightpath angle
AZ	Resultant azimuth
R	Resultant radius
LONG	Resultant geographic longitude
DECL	Resultant declination
GETV	Ground elapsed time of resultant vector
DVX	LVLH delta velocity component in X direction
DVY	LVLH delta velocity component in Y direction
DVZ	LVLH delta velocity component in Z direction
WEIGHT I	Initial vehicle weight
DELTA WHT	Vehicle weight loss during burn
WEIGHT F	Final vehicle weight
GMTI	Greenwich mean time of burn initiation, in hr., min., sec.
GETI	Ground elapsed time of burn initiation, in hr., min., sec.
DEL V MAN	Total delta velocity magnitude of maneuver
DEL TB MAN	Total burn duration, in seconds
PIT MAN	LVLH pitch angle of maneuver
YAW MAN	LVLH yaw angle of maneuver
H MAN	Altitude of vehicle at burn initiation

TABLE 4-VI.- Concluded

PROCESSOR GPMP

GENERAL PURPOSE MANEUVER TABLE

Display parameter label	Parameter definition
A	Resultant semimajor axis
E	Resultant eccentricity
I	Resultant inclination
TRUE A	Resultant true anomaly
DEL I	Resultant change in inclination
DEL NODE	Resultant change in ascending node
DEL G	Resultant change in argument of perigee
PERIOD	Resultant orbital period; in minutes

TABLE 4-VII.- PROCESSOR MESSAGE TABLE

PROCESSOR <u>GPMP</u>

MSG no.	Message ID block	Message text block and explanation
1	*GPMP*	*GPMIN* INPUT VECTOR IS NOT TEG/CARTESIAN Meaning: The user has input a vector that is not in the true Equator and Greenwich meridian of epoch Cartesian coordinate system. Severity: The processor will terminate, and control will be returned to the FDS Executive. Action required by user: Transform the vector to TEG/Cartesian; then resume execution.
2	*GPMP*	*GPMIN* THE FOLLOWING CODE IS INVALID INPUT: "XXXXXX". Meaning: The user has supplied an incorrect maneuver code. Severity: The processor will terminate, and control will be returned to the FDS Executive. Action required by user: Use the interface table editor to correct the erroneous code; then resume execution.
3	*GPMP*	*HIGHT* SFAIL = 1. ITERV TERMINATED. Meaning: ITERV did not converge on the velocity required for the height adjustment maneuver. Severity: This is a warning only. The processor will continue. Action required by user: None.
4	*GPMP*	*SHIFT* MAXIMUM ITERATIONS. SFAIL = XX. Meaning: ITERV did not converge on the minimum velocity required for the line of apsides maneuver. Severity: This is a warning only; the processor will continue. Action required by user: None.
5		ALARM FLAG SET TO 8 IN UPDATE ROUTINE Meaning: UDATV did not converge on angle in 10 iterations. Severity: This is a warning only. The processor will continue. Action required by user: None.

TABLE 4-VIII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR GPMP

Processor name	Processor abstract prompt (maximum 256 characters)
GPMP	The GPMP computes a single vehicle impulsive maneuver that is initiated at a specified maneuver point in the vehicle orbit and is targeted to a specified final orbit condition.
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
GLOCON	Global constants array; master data base element normally defaulted to !!GLCN.
SESCON	Session constants array; normally defaulted to !SESCN.
PROCON	GPMP constants and tolerances
BVEC	Position/velocity input state vector
THRUST	Effective thrust of vehicle engine
ISP	Specific impulse of vehicle engine, in seconds
MANID	Maneuver identification code. The codes are "DELV", "PLCH", "NOSH", "CIRC", "CRPC", "CRNS", "HITE", "HTPC", "HTNS", "SHFT", "SHKP", and "HAHP".
MNSTOP	Maneuver location code. The codes are: "CUR", "APS", "TIM", "ALT", "LAT", "LON", "RAD", "APO", "PER", "OPT", and "ARG".

TABLE 4-VIII.- Continued

PROCESSOR GPMP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
STOPVL	Value of maneuver location stop condition. When MANSTOP = : "CUR"; = N/A "APS"; = N^{th} apsidal crossing "TIM"; = N/A "ALT"; = Altitude "LAT"; = Latitude "LON"; = Longitude "RAD"; = Radius "APO"; = N^{th} apogee crossing "PER"; = N^{th} perigee crossing "OPT"; = N/A "ARG"; = Argument of latitude
THRT	Threshold time to begin search for MNSTOP (or maneuver time)
MANOPT	Maneuver Option "LVLH"; = Input DVX,DVY,DVZ "DVYP"; = Input DV,YAW,PITCH "H"; = Input altitude "DH" = Input delta altitude

TABLE 4-VIII.- Continued

PROCESSOR GPMP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
APSOPT	Line of apsides option = "NO" -- Do not create apsis at maneuver point = "YES" -- Create apsis and maneuver = "PER" -- Perform maneuver near perigee = "APO" -- Perform maneuver near apogee
MNSPEC	Maneuver specifications. The combinations are as follows: 1. DVX,DVY,DVZ 2. DV,YAW,PITCH 3. Wedge angle 4. Delta node 5. Altitude 6. Delta altitude 7. Altitude, wedge angle 8. Altitude, delta node 9. Delta altitude, wedge angle 10. Delta altitude, delta node 11. Desired amount of delta true anomaly 12. Apogee, perigee
SVPROP	State vector propagation flag = "CON"; Conic = "AEG"; Analytical Emphemeris Generator

TABLE 4-VIII.- Concluded

PROCESSOR GPMP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
PVTAB	Position/velocity phase table. Three vectors are generated per call to GPMP: (1,1) Initial (1,2) Ignition (1,3) Burn-out
SUMTAB	Output Summary Table

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - MAIN PROGRAM GPMP

5.1.1 Purpose

The General Purpose Maneuver Program (GPMP) is the main routine of the General Purpose Maneuver Processor. GPMP is designed to be the driver for each of the segments residing in the processor.

5.1.2 Functional Description

The General Purpose Maneuver Processor is too large to reside entirely in core at one time. Consequently, the function of GPMP is to control the computational flow of the processor, and to load into core each segment of the processor as it is needed. It also sets up the initialization and finalization functions required of each processor.

5.1.3 Assumptions and Limitations

None.

5.1.4 Method

None.

5.1.5 Routine Input/Output Variables

The input/output variables are given in table 5.1-I.

5.1.6 Functional Logic Flow

The functional logic flow is given in figure 5.1-1.

5.1.7 Diagnostics and Debug

None.

5.1.8 Special Comments

None.

5.1.9 <u>References</u>

None.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine GPMP

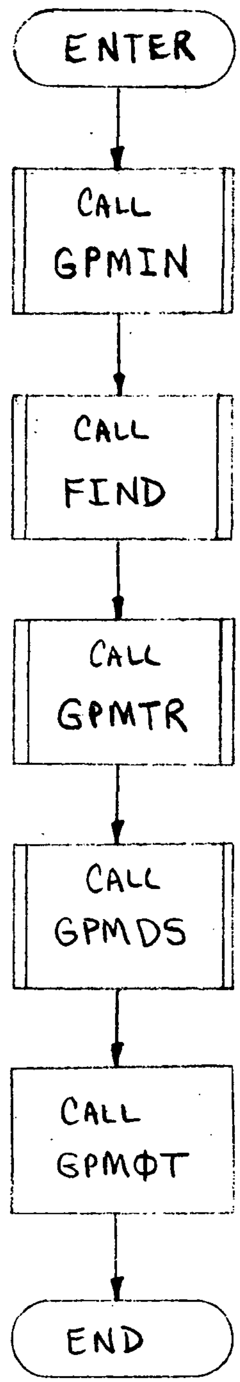
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BVEC	—	Real	I	—	C	BVEC	Base vector
CONST	--	Real	I	—	C	--	Earth constants array
ERROR	--	Intg	I	—	C	--	Processor error return flag
GPMINP	--	Real	I	—	C	--	Parameters array
INTBUF	--	Intg	I/O	—	C	—	Interface table header
IPRAM	--	Intg	I/O	—	A	--	System parameters
LU	—	Intg	O	—	A	IPRAM(1)	User logical unit number
RP	$\bar{R}_P$	Real	I/O	ft	C	--	Position vector
SESCON	--	Free	I	—	C	!SESCN	Session constants
T	T	Real	I/O	sec	C	--	Time of vector
VP	$\bar{V}_P$	Real	I/O	fps	C	—	Velocity vector
VPF	$\bar{V}_{PF}$	Real	O	fps	C	--	Final velocity vector

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.1-1.- GPMP functional logic flow.

5.2 ROUTINE NAME - GPMIN

5.2.1 Purpose

GPMIN is the General Purpose Maneuver Processor data input routine.

5.2.2 Functional Description

GPMIN calls the system routine XPGET to obtain the user supplied input quantities necessary to run GPMP. Then, the input data is interpreted, converted to internal units, and assigned to the appropriate common locations for execution. If the input state vector is found not to be defined in the True-Equatorial-Greenwich reference axis system (TEG), GPMIN sets the error return code to -32768 prior to exiting the routine.

5.2.3 Assumptions and Limitations

None.

5.2.4 Method

None.

5.2.5 Routine Input/Output Variables

The input/output variables are given in table 5.2-I.

5.2.6 Functional Logic Flow

The functional logic flow is given in figure 5.2-1.

5.2.7 Diagnostics and Debug

When an inappropriate parameter code for the state vector coordinate system is encountered, GPMIN sends a message to the user, sets the appropriate error return code, and exits the routine.

If the processor debug flag is set, GPMIN writes the maneuver type flag, the input state vector, the search option flag, the maneuver location parameter value, and the threshold time.

5.2.8 Special Comments

None.

5.2.9 References

None.

TABLE 5.2-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine GPMIN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LU	—	Intg	I	—	C	IPRAM(1)	Logical unit number
INTERFACE TABLE VARIABLES	—	—	I	—	IT	—	See table 4-I of this processor for definition of contents
AREA	—	Real	O	ft^2	C	—	Vehicle reference area
BVEC	—	Real	O	—	C	—	Base vector
CD	C_D	Real	O	—	C	—	Vehicle coefficient of drag
DELF	ΔF	Real	O	rad	C	—	Delta true anomaly
DELH	Δh	Real	O	ft	C	—	Delta height
DNODE	ΔN	Real	O	rad	C	—	Delta node shift angle
DPC	ΔPc	Real	O	rad	C	—	Delta plane change angle
DV	ΔV	Real	O	fps	C	—	Delta velocity magnitude
DVOPT	—	Intg	O	—	C	—	Delta velocity option

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.2-I.- Continued

Routine GPMIN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
DVX	ΔV_X	Real	O	fps	C	--	Delta velocity component - X
DVY	ΔV_Y	Real	O	fps	C	--	Delta velocity component - Y
DVZ	ΔV_Z	Real	O	fps	C	--	Delta velocity component - Z
ERROR	--	Intg	O	—	C	--	Input error return code
H	h	Real	O	ft	C	—	Height
HA	ha	Real	O	ft	C	--	Height of apogee
HOPT	—	Intg	O	—	C	—	Height option
HP	hp	Real	O	ft	C	--	Height of perigee
IAPS	--	Intg	O	—	C	—	Apsis option flag
MN	--	Intg	O	—	C	--	Maneuver option number
NPRINT	--	Intg	O	—	C	—	Internal print flag
PARAM	--	Real	O	—	C	--	Maneuver location parameter

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.2-I.- Concluded

Routine GPMIN

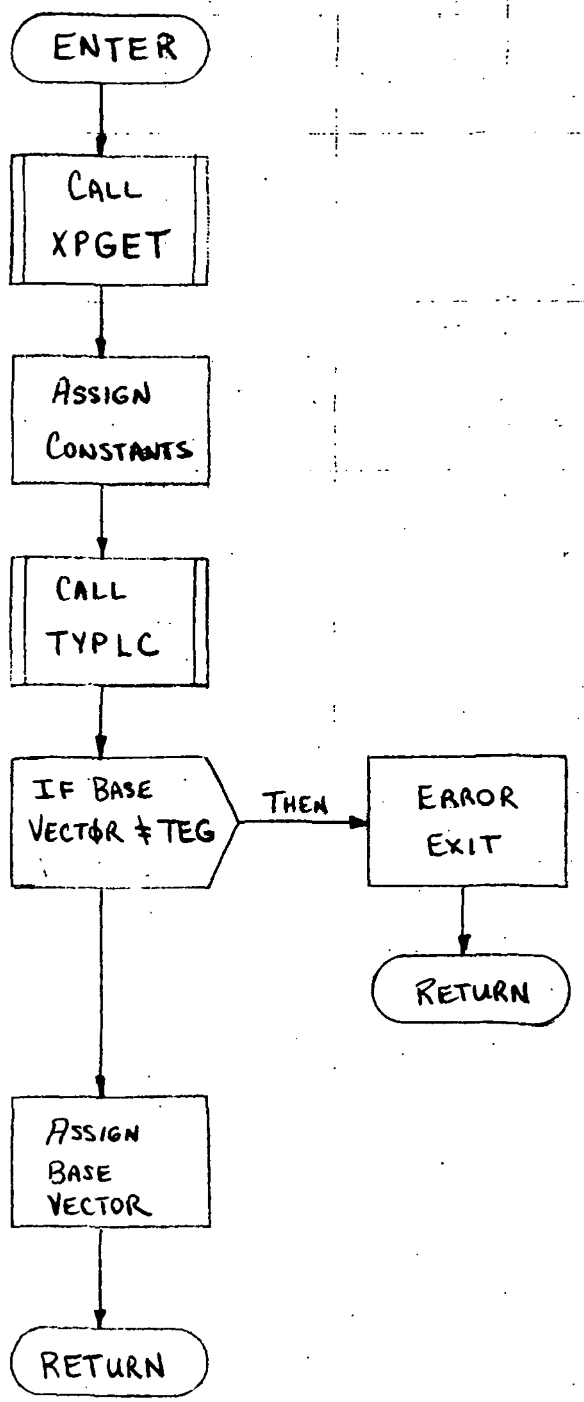
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
PITCH	P	Real	O	rad	C	--	LVLH pitch angle
RP	$\bar{R}_p$	Real	O	ft	C	--	Vehicle position vector
SARCH	—	Intg	O	—	C	--	SIRCH maneuver location flag
SPERT	—	Intg	O	—	C	--	State vector propagation flag
T	t	Real	O	sec	C	—	Time of vehicle state vector
TI	t_i	Real	O	sec	C	--	Maneuver threshold time
VP	$\bar{V}_p$	Real	O	fps	C	—	Vehicle velocity vector
WHT	W_i	Real	O	lb	C	--	Vehicle gross mass
YAW	Y	Real	O	rad	C	—	LVLH yaw angle

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.2-1.- GPMIN functional logic flow.

5.3 ROUTINE NAME - TYPLC

5.3.1 Purpose

The Maneuver Type and Location (TYPLC) routine interprets and assigns a numerical value to each parameter that requires ASCII input by the user.

5.3.2 Functional Description

There are six parameters in the GPMP interface table that require ASCII input by the user. TYPLC is called sequentially by the routine GPMIN, using a routing flag, to have each parameter interpreted. When the input is recognized, a numerical value is assigned the parameter, which subsequently will be used by GPMP for proper program routing.

5.3.3 Assumptions and Limitations

None.

5.3.4 Method

The correct user responses for each parameter are stored in data statements within TYPLC. Each user response is compared, bit-by-bit, using the system routine XRCPR. When a match occurs, an appropriate numerical value is assigned to the parameter. If no match occurs, the user is informed of an input error, the error return code flag is set, and the processor GPMP is terminated.

5.3.5 Routine Input/Output Variables

The TYPLC input/output variables are presented in table 5.3-1.

5.3.6 Functional Logic Flow

The functional logic flow is presented in figure 5.3-1.

5.3.7 Diagnostics and Debug

None.

5.3.8 Special Comments

None.

5.3.9 References

None.

TABLE 5.3-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine TYPLC

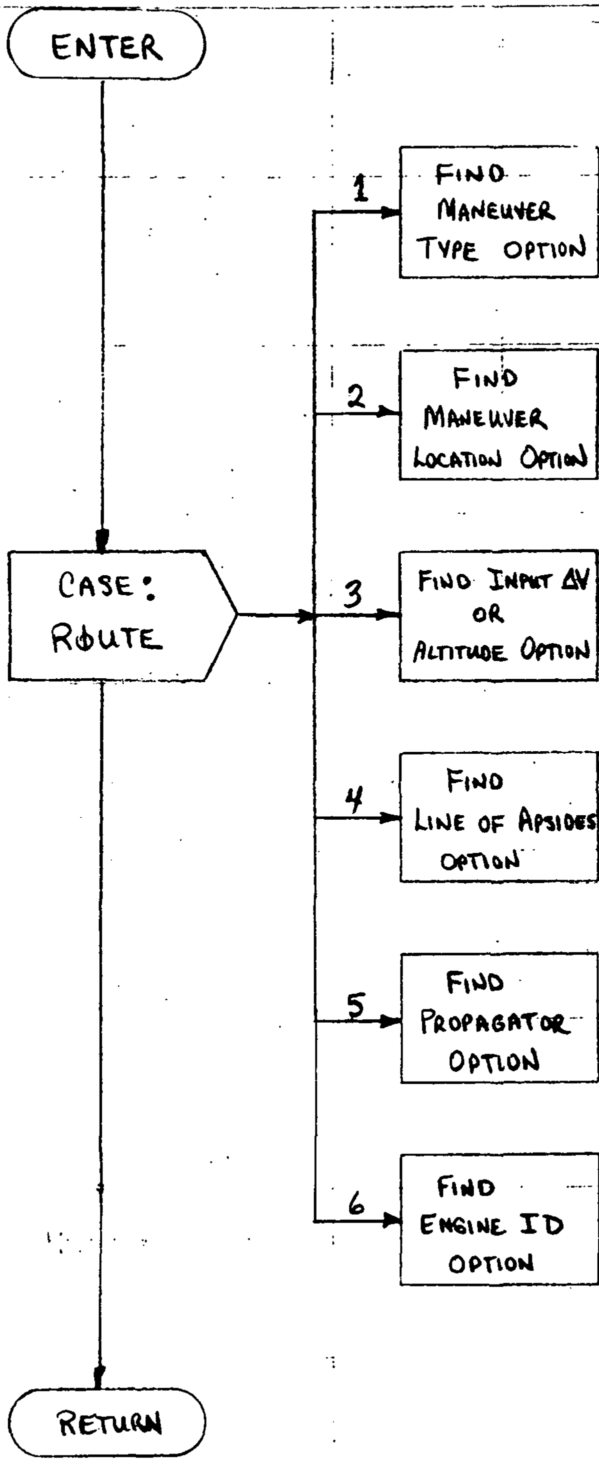
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IBUG	--	Intg	I	--	A	--	Logical unit number for internal print
ROUTE	--	Intg	I	--	A		Routing flag
ID	--	6CH	I	--	A	--	Three-word array containing maneuver code
NUMBER	--	Intg	O	--	A	--	Numerical value for maneuver code
ERROR	--	Intg	O	--	A	--	Error return code

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.3-1.- TYPLC functional logic flow.

5.4 ROUTINE NAME - FIND

5.4.1 Purpose

FIND is the driver routine for SIRCH, the program that computes the time of arrival of the maneuvering vehicle at a specified condition.

5.4.2 Functional Description

FIND takes the input vehicle state vector, maneuver threshold time, and user specified stop condition and calls the SIRCH program, which propagates the state vector to the time of the stop condition. The updated state vector and time is then returned by FIND to the main routine, GPMP. Since SIRCH is a program itself, FIND is the interface between GPMP and SIRCH.

5.4.3 Assumptions and Limitations

None.

5.4.4 Method

None.

5.4.5 Routine Input/Output Variables

The input/output variables are given in table 5.4-I.

5.4.6 Functional Logic Flow

None.

5.4.7 Diagnostics and Debug

None.

5.4.8 Special Comments

None.

5.4.9 References

None.

TABLE 5.4-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine FIND

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
AREA	--	Real	I	ft^2	C	--	Vehicle reference area
CD	C_D	Real	I	--	C	--	Coefficient of drag
PARAM	--	Real	I	--	C	--	Value of specified search condition
RP	$\bar{R}_P$	Real	I/O	ft	C	--	Vehicle position vector
SARCH	--	Intg	I	--	C	--	Search selection flag
SPERT	--	Intg	I	--	C	--	Vector propagation flag
T	t	Real	I/O	sec	C	--	Vector time
VP	$\bar{V}_P$	Real	I/O	fps	C	--	Vehicle velocity vector
WHT	W_i	Real	I	lb	C	--	Vehicle weight
TI	t_i	Real	I/O	sec	C	--	Threshold or maneuver time
ERROR	--	Intg	O	--	C	--	Error return code

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.5 ROUTINE NAME - GPMTR

5.5.1 Purpose

GPMTR is the main routine of the targeting portion of the General Purpose Maneuver Processor. GPMTR is designed to be the driver for each of the maneuver types within the processor.

5.5.2 Functional Description

When the targeting segment of the GPMP is called, GPMTR acts as the driver for the segment. Dependent upon the user supplied maneuver type flag, GPMTR calls the appropriate subroutine to compute the maneuver.

5.5.3 Assumptions and Limitations

None.

5.5.4 Method

None.

5.5.5 Routine Input/Output Variables

The maneuver number flag, MN, is the only input to the routine. It is the routing flag to call the appropriate maneuver computation routine.
There are no output parameters from the routine.

5.5.6 Functional Logic Flow

The functional logic flow is presented in figure 5.5-1.

5.5.7 Diagnostics and Debug

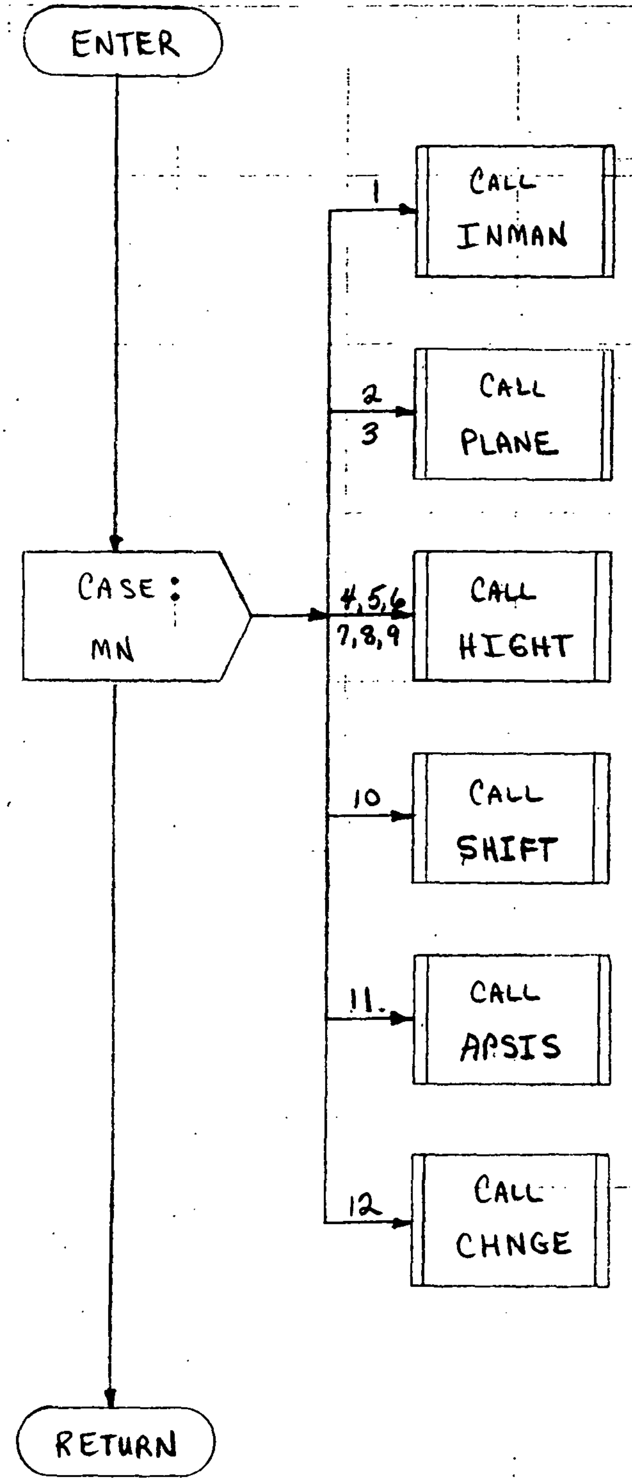
None.

5.5.8 Special Comments

None.

5.5.9 References

None.



Page 1 of 1

Figure 5.5-1.- GPMTR functional logic flow.

5.6 ROUTINE NAME - INMAN

5.6.1 Purpose

INMAN is the Input Delta Velocity Maneuver routine.

5.6.2 Functional Description

Given the vehicle state vector at the maneuver time, and either the local vertical/ local horizontal delta velocity components or the delta velocity magnitude, and the yaw and pitch angles, INMAN will compute the final velocity vector that is appropriate for those conditions.

5.6.3 Assumptions and Limitations

The final velocity vector is determined by using a yaw-pitch angle sequence.

5.6.4 Method

Using the delta-V magnitude, yaw, and pitch to define them, the LVLH delta velocity vector components are defined in the LVLH coordinate system. Thus,

$$a = \Delta V \sin P$$

$$b = \Delta V \cos P$$

$$c = b \sin Y$$

$$d = b \cos Y,$$

and

$$\Delta Vx = d = \Delta V \cos P \cos Y$$

$$\Delta Vy = c = \Delta V \cos P \sin Y$$

$$\Delta Vz = -a = -\Delta V \sin P ,$$

where P = pitch

Y = yaw

The inertial delta velocity vector is then defined to be

$$\Delta\overline{V}_I = \Delta V_x \overline{X}_{LV} + \Delta V_y \overline{Y}_{LV} + \Delta V_z \overline{Z}_{LV},$$

where $\overline{X}_{LV}$, $\overline{Y}_{LV}$, and $\overline{Z}_{LV}$ are the unit vectors which define the LVLH coordinate system.

The post-burn velocity vector, $\overline{V}_{PF}$, is then given by $\overline{V}_{PF} = \overline{V}_P + \Delta\overline{V}_I$.

The delta velocity vector for the maneuver in the LVLH coordinate system is depicted in Figure 5.6-1(a). The same vector in the inertial coordinate system is depicted in Figure 5.6-1(b).

5.6.5 Routine Input/Output Variables

The input/output variables are given in Table 5.6-I.

5.6.6 Functional Logic Flow

The functional logic flow is given in Figure 5.6-2.

5.6.7 Diagnostics and Debug

None.

5.6.8 Special Comments

None.

5.6.9 References

None.

TABLE 5.6-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine INMAN

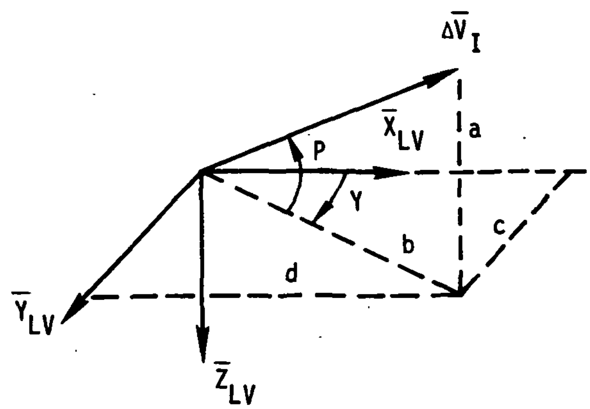
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
VP	$\bar{V}_P$	Real	I	fps	C	--	Vehicle velocity vector
DV	ΔV	Real	I/O	fps	C	--	Delta velocity magnitude
DVOPT	--	Intg	I	--	C	--	Delta-V input option flag
DVX	ΔV_X	Real	I/O	fps	C	--	LVLH ΔV component - X
DVY	ΔV_Y	Real	I/O	fps	C	--	LVLH ΔV component - Y
DVZ	ΔV_Z	Real	I/O	fps	C	--	LVLH ΔV component - Z
PITCH	P	Real	I	rad	C	--	LVLH pitch angle
YAW	Y	Real	I	rad	C	--	LVLH yaw angle

NOTES:

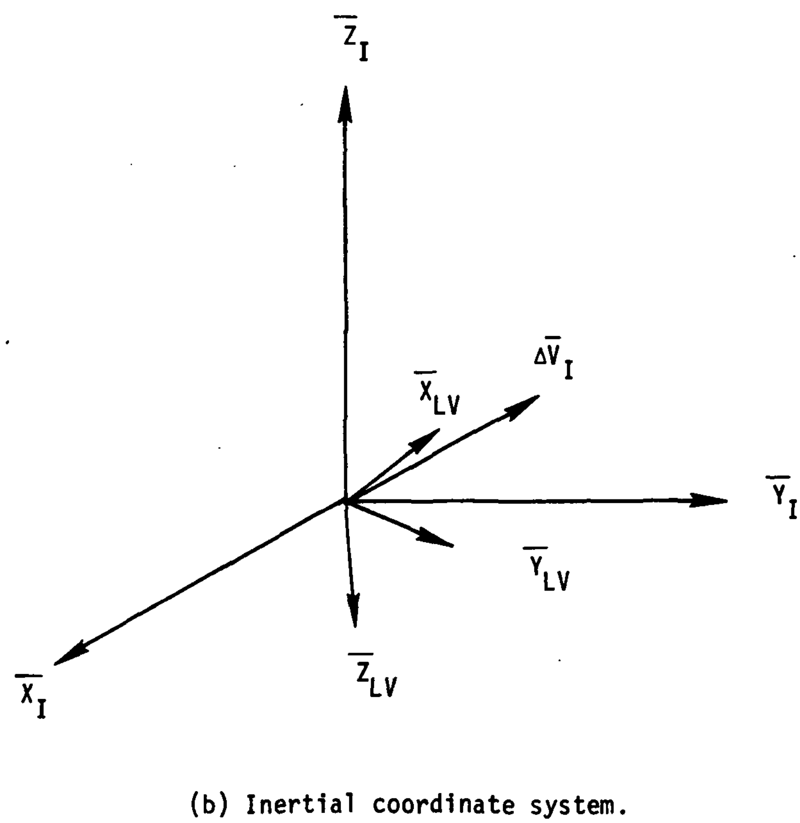
TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



(a) LVLH coordinate system.



(b) Inertial coordinate system.

Figure 5.6-1.- Coordinate systems for INMAN subroutine.

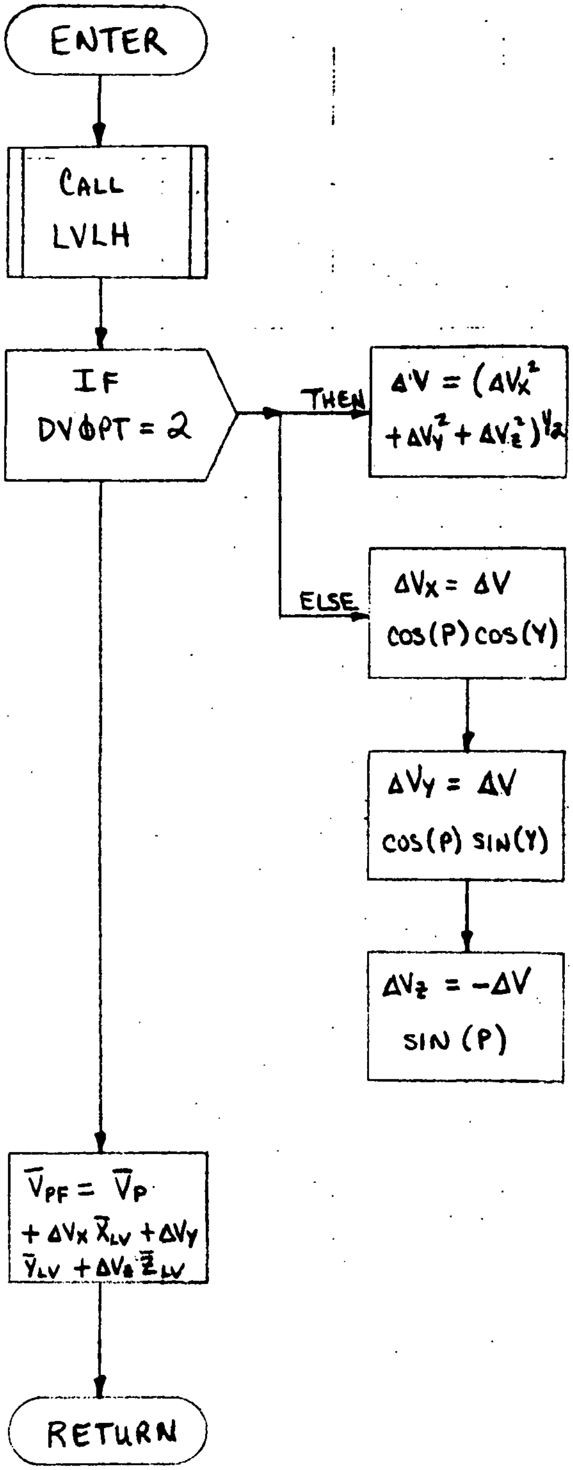


Figure 5.6-2.- INMAN functional logic flow.

5.7 ROUTINE NAME - LVLH

5.7.1 Purpose

LVLH defines the local vertical/local horizontal coordinate system.

5.7.2 Functional Description

Given the pre-burn position and velocity vectors, LVLH computes the local vertical/local horizontal coordinate system.

5.7.3 Assumptions and Limitations

None.

5.7.4 Method

The LVLH coordinate system is defined by the following equation (see Figure 5.7-1):

$$\bar{Z}_{LV} = \frac{-\bar{R}_P}{|R_P|}$$

$$\bar{Y}_{LV} = \frac{\bar{V}_P \times \bar{R}_P}{|\bar{V}_P \times R_P|}$$

$$\bar{X}_{LV} = \bar{Y}_{LV} \times \bar{Z}_{LV}.$$

5.7.5 Routine Input/Output Variables

The input/output variables are given in table 5.7-I.

5.7.6 Functional Logic Flow

None.

5.7.7 Diagnostics and Debug

None.

5.7.8 Special Comments

None.

5.7.9 References

None.

TABLE 5.7-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LVLH

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Velocity vector
XLV	$\bar{X}_{LV}$	Real	O	—	A	--	Unit vector of $(\bar{Y}_{LV} \times \bar{Z}_{LV})$
YLV	$\bar{Y}_{LV}$	Real	O	—	A	--	Unit vector of $(\bar{V}_P \times \bar{R}_P)$
ZLV	$\bar{Z}_{LV}$	Real	O	—	A	--	Unit vector of $(-\bar{R}_P)$

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

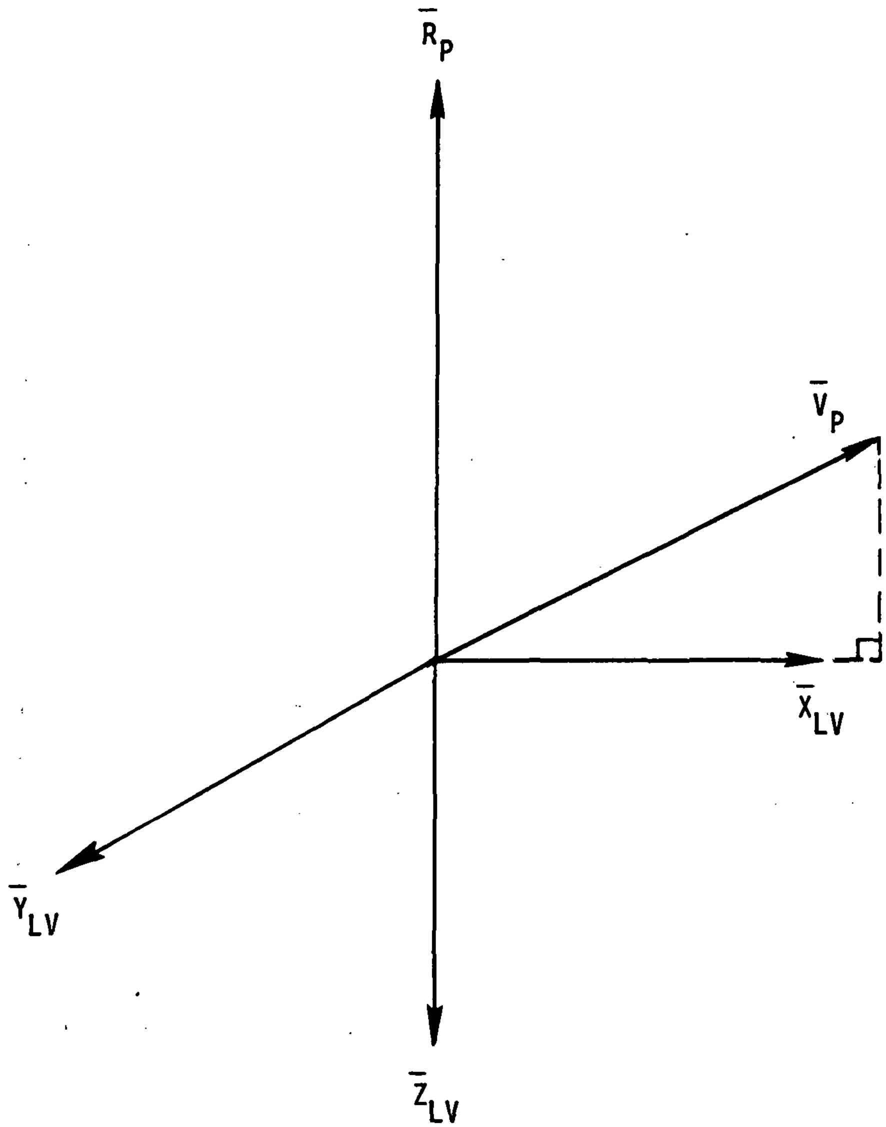


Figure 5.7-1.- Definition of LVLH coordinate system.

5.8 ROUTINE NAME - PLANE

5.8.1 Purpose

PLANE is the plane change maneuver and the node shift maneuver targeting routine.

5.8.2 Functional Description

Given the vehicle state vector at the maneuver time and the amount of plane change requested by the user, PLANE computes the post-burn velocity vector that will effect the requested plane change. Or, given the vehicle state vector and the amount of node shift, PLANE will compute the corresponding plane change angle and then the post-burn velocity vector that will effect the requested node shift.

5.8.3 Assumptions and Limitations

None.

5.8.4 Method

a. When a plane change maneuver is requested, PLANE calls the routine FINAL, which computes a final velocity vector based on the requested plane change angle, DPC. When a node shift is requested, the wedge angle, δ, which is the angle between the pre-burn and the post-burn orbital planes, must be computed first. Then FINAL is called.

Define ν to be the angular distance along the equator between the ascending node and the longitude of the maneuver (fig. 5.8-1). Then,

$$\nu_b = \lambda - \Omega_b$$

and $$\nu_a = \nu_b - \Delta N \ ,$$

where λ = longitude of maneuver
Ω = right ascension of ascending node
ΔN = node shift angle.
b = pre-burn orbit
a = post-burn orbit

Define U to be the argument of latitude of the maneuver point. Then,

$$U_b = \cos^{-1} [\cos \phi_c \cos\nu_b]$$

and $$U_a = \cos^{-1} [\cos \phi_c \cos\nu_a],$$

where ϕ_c = geocentric latitude of maneuver.

Using the spherical triangle ABC in Figure 5.8-1,

$$\cos \Delta N = \cos U_a \cos U_b + \sin U_a \sin U_b \cos \delta.$$

Solving for δ , the wedge angle between the two orbit planes

$$\delta = \cos^{-1} \left[\frac{\cos \Delta N - \cos U_a \cos U_b}{\sin U_a \sin U_b} \right]$$

b. For Plane Change.-

The relationship between the sign of the desired plane change angle (DPC) and the sign of the wedge angle (δ) is,

DPC	δ	U_b
+	-	$-\pi/2 \leq U \leq \pi/2$
-	+	$-\pi/2 \leq U \leq \pi/2$
+	+	$\lvert U \rvert > \pi/2$
-	-	$\lvert U \rvert > \pi/2$

c. For Node Shift.-

The relationship between the sign of the node shift angle (ΔN) and the sign of the wedge angle (δ) is,

ΔN	δ	U_b
+	-	+
-	+	+
+	+	-
-	-	-

5.8.5 Routine Input/Output Variables

The input/output variables are given in Table 5.8-I.

5.8.6 Functional Logic Flow

The functional logic flow is given in Figure 5.8-2.

5.8.7 Diagnostics and Debug

If internal print is requested, various pre-burn and post-burn parameters are printed out so that a check of the logic flow and of the equations used may be made.

5.8.8 Special Comments

None.

5.8.9 References

None.

TABLE 5.8-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine PLANE

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Vehicle position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Vehicle velocity vector
DNODE	ΔN	Real	I	rad	C	--	Node shift angle
DPC	--	Real	I	rad	C	--	Plane change angle
MN	--	Intg	I	--	C	--	Maneuver number
VPF	$\bar{V}_{PF}$	Real	O	fps	C	--	Postburn velocity vector

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

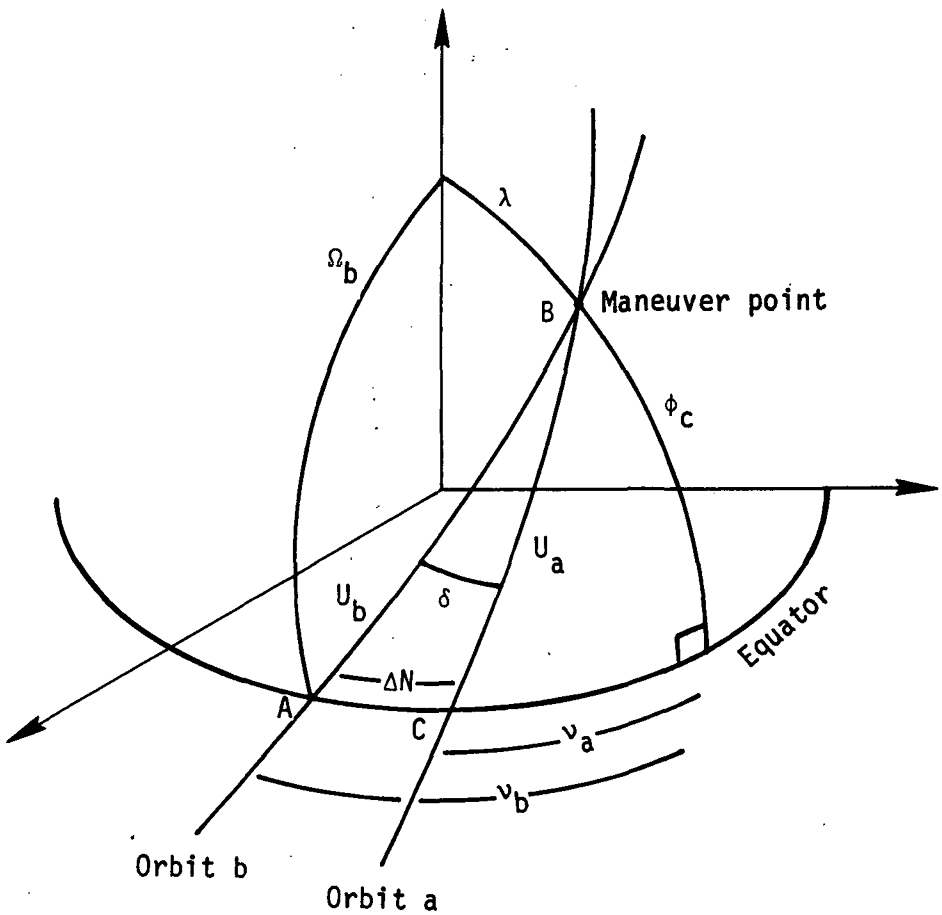
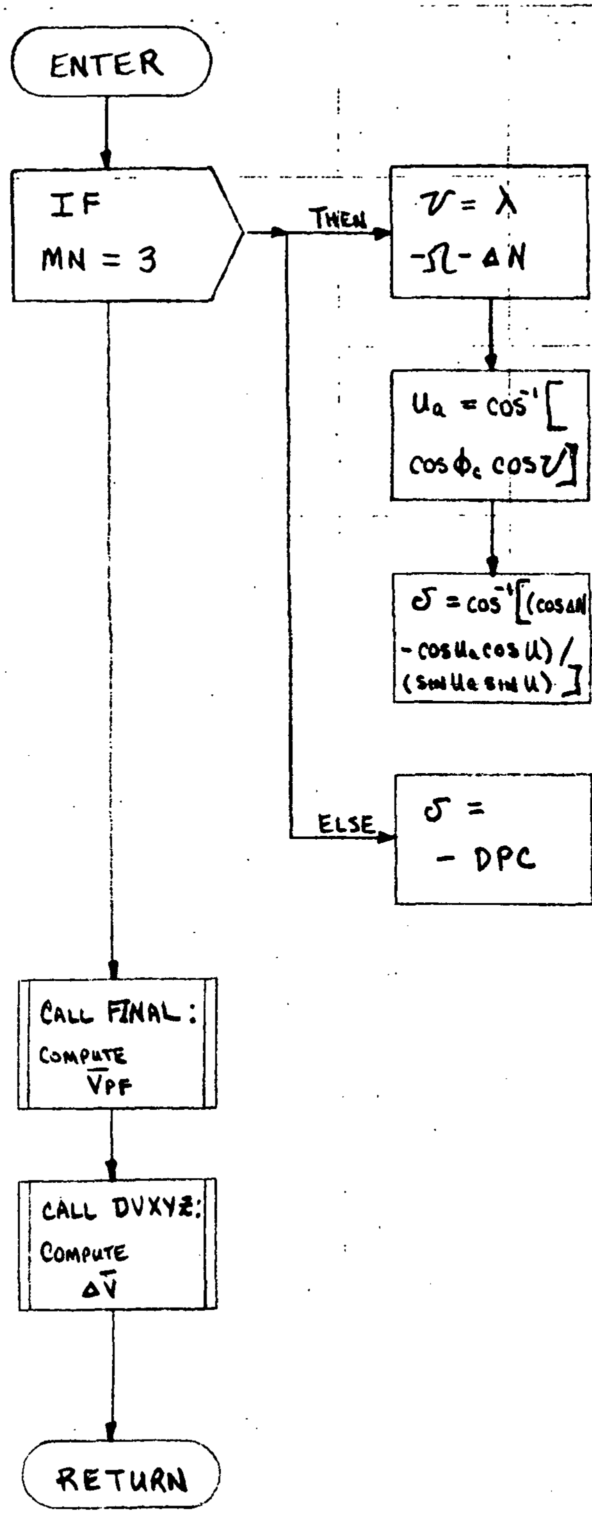


Figure 5.8-1.- Node shift geometry.



Page 1 of 1

Figure 5.8-2.- PLANE functional logic flow.

5.9 ROUTINE NAME - APIE

5.9.1 Purpose

APIE converts a vector from cartesian to geocentric spherical coordinates.

5.9.2 Functional Description

Given the position and velocity vectors in cartesian coordinates, APIE computes the corresponding elements in the geocentric spherical coordinate system.

5.9.3 Assumptions and Limitations

None.

5.9.4 Method

The conversion from cartesian to spherical is accomplished with the following equations (fig. 5.9-1):

a. $R = |\bar{R}_P|$,

where $\bar{R}_P$ is the radius vector.

b. $V = |\bar{V}_P|$,

where $\bar{V}_P$ is the velocity vector.

c. $\gamma = \sin^{-1} [(\bar{R}_P \cdot \bar{V}_P)/(RV)]$,

where γ is the flight path angle.

d. Define the unit vectors $\bar{I}$, $\bar{J}$, $\bar{K}$ to lie along the inertial X, Y, Z axes, respectively. Then, the angular momentum vector is defined to be

$$\bar{H} = H_1\bar{I} + H_2\bar{J} + H_3\bar{K}$$

$$= \bar{R}_P \times \bar{V}_P,$$

and $$H = |\bar{H}|.$$

e. Define the vector, $\bar{N}$, to point along the line of nodes of the orbital and equatorial planes in the direction of the ascending node. Thus,

$$\bar{N} = \bar{K} \times \bar{H}$$

$$= -H_2\bar{I} + H_1\bar{J},$$

and $$N = |\bar{N}|.$$

f. Since the inclination, i, is the angle between

$\bar{K}$ and $\bar{H}$,

$$\cos i = \frac{H_3}{H}$$

and $$\sin i = \frac{N}{H}.$$

g. Since the argument of latitude, U, is the angle between $\bar{N}$ and $\bar{R}_P$,

$$\cos U = \frac{\bar{N} \cdot \bar{R}_P}{NR}$$

h. The geocentric latitude, ϕ_c, is

$$\phi_c = \tan^{-1} [R_3/(R_1^2 + R_2^2)^{1/2}],$$

where R_1, R_2, R_3 are the vector components of $\bar{R}_P$. This is depicted in Figure 5.9-2.

i. The inertial longitude, λ, is

$$\lambda = \tan^{-1} (R_2/R_1).$$

j. The azimuth, ψ, is derived as follows.

$$\begin{aligned}
\tan \psi &= \sin \psi/\cos \psi \\
&= (\cos i/\cos \phi_c)/[(\sin i \cos U)/\cos \phi_c] \\
&= \cos i/(\sin i \cos U) \\
&= \left(\frac{H_3}{H}\right) / \left[\left(\frac{N}{H}\right)\left(\bar{N} \cdot \bar{R}_p\right)/(NR)\right] \\
&= (RH_3)/(\bar{N} \cdot \bar{R}_p)
\end{aligned}$$

thus, $\Psi = \tan^{-1} [(RH_3)/(\bar{N} \cdot \bar{R}_p)]$.

5.9.5 Routine Input/Output Variables

The routine input/output variables are given in Table 5.9-I.

5.9.6 Functional Logic Flow

None.

5.9.7 Diagnostics and Debug

None.

5.9.8 Special Comments

None.

5.9.9 References

None.

TABLE 5.9-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine APIE

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
X	$\bar{X}$	Real	I	ft	A	RP	Position vector
XD	$\dot{\bar{X}}$	Real	I	fps	A	VP	Velocity vector
SPHER	--	Real	O	--	A	--	Geocentric spherical element set (velocity, flightpath angle, azimuth, radius, longitude, latitude)

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

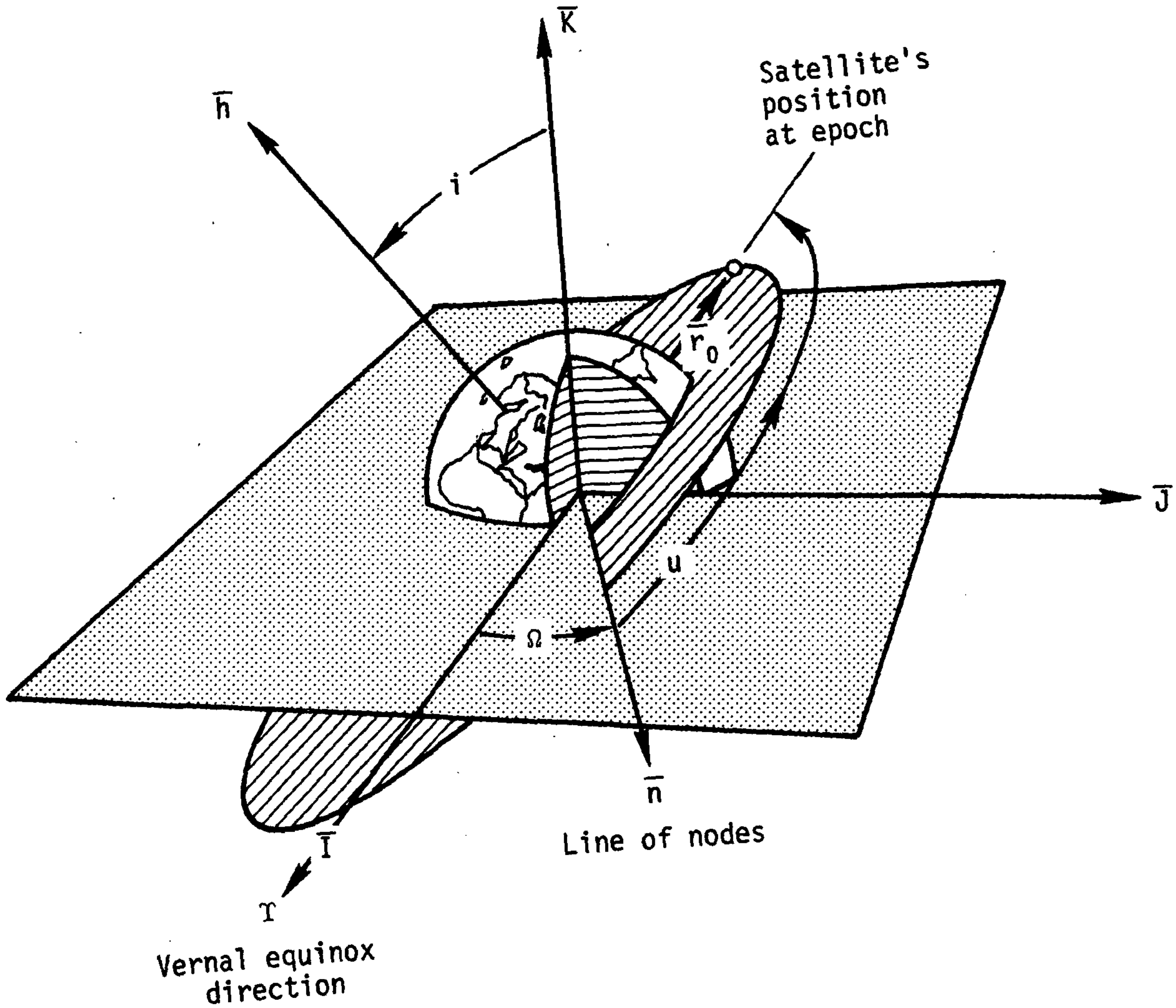


Figure 5.9-1.- Orbital elements.

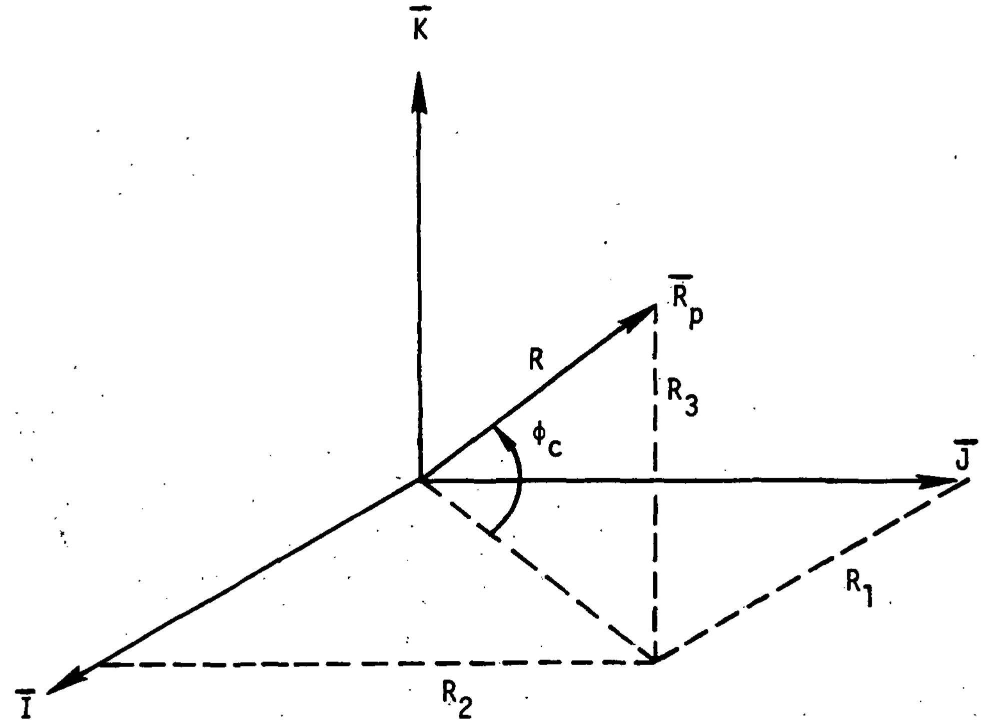


Figure 5.9-2.- Computation of geocentric latitude.

5.10 ROUTINE NAME - FINAL

5.10.1 Purpose

FINAL computes the post-burn velocity vector in the inertial coordinate system.

5.10.2 Functional Description

Given the pre-burn position and velocity vectors, the post-burn velocity magnitude and flight path angle, and the wedge angle between the pre-burn and post-burn planes, FINAL computes the post-burn velocity vector in the local vertical/local horizontal (LVLH) coordinate system. It then transforms that vector to the inertial coordinate system.

5.10.3 Assumptions and Limitations

None.

5.10.4 Method

In order to determine the post-burn velocity vector defined in the inertial coordinate system, FINAL first calculates the vector in the LVLH coordinate system. The equations used are the following (fig. 5.10-1).

$$a = V \sin \gamma$$
$$b = V \cos \gamma$$
$$c = b \sin \delta$$
$$d = b \cos \delta,$$

where v = post-burn velocity magnitude
γ = flight path angle
δ = wedge angle between pre-burn and post-burn planes.

Then, the post-burn velocity vector components in the LVLH system are defined to be:

$$V_{LVLH}(1) = V \cos \gamma \cos \delta$$
$$V_{LVLH}(2) = V \cos \gamma \sin \delta$$
$$V_{LVLH}(3) = -V \sin \delta.$$

The post-burn vector in the inertial system, $\bar{V}_{PF}$, is then determined by a vector transformation matrix.

5.10.5 Routine Input/Output Variables

The input/output variables are given in Table 5.10-I.

5.10.6 Functional Logic Flow

None.

5.10.7 Diagnostics and Debug

If internal print is desired, the $\bar{V}_{LVLH}$ and the $\bar{V}_{PF}$ vectors are displayed for reference.

5.10.8 Special Comments

None.

5.10.9 References

None.

TABLE 5.10-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine FINAL

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	A	—	Position vector
VP	$\bar{V}_P$	Real	I	fps	A	--	Preburn velocity vector
V	--	Real	I	fps	A	--	Postburn velocity magnitude
GAM	γ	Real	I	rad	A	--	Postburn flightpath angle
DELTA	δ	Real	I	rad	A	--	Wedge angle between planes
VPF	$\bar{V}_{PF}$	Real	O	fps	A	--	Postburn velocity vector

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

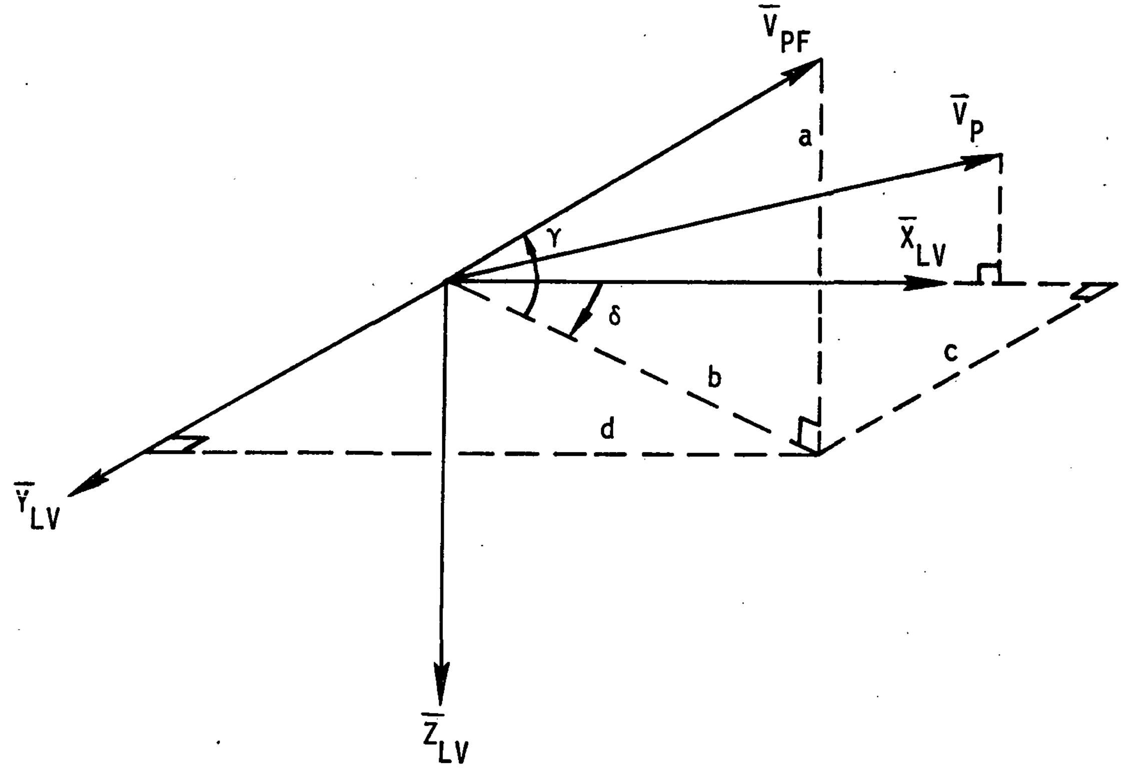


Figure 5.10-1.- Post-burn velocity vector in LVLH coordinate system.

5.11 ROUTINE NAME - DVXYZ

5.11.1 Purpose

DVXYZ calculates the delta velocity magnitude, inertial vector, and local vertical/local horizontal vector.

5.11.2 Functional Description

Given the position vector, and the pre-burn and post-burn velocity vectors, DVXYZ computes the inertial delta velocity vector, the delta velocity magnitude, and the LVLH delta velocity vector.

5.11.3 Assumptions and Limitations

None.

5.11.4 Method

The inertial delta velocity vector is defined to be

$$\Delta\bar{V}_I = \bar{V}_{PF} - \bar{V}_P.$$

The delta velocity magnitude is

$$\Delta V = |\Delta\bar{V}_I|.$$

The LVLH delta velocity vector is determined by a vector transformation of

$$\Delta\bar{V}_I.$$

5.11.5 Routine Input/Output Variables

The routine input/output variables are given in table 5.11-I.

5.11.6 Functional Logic Flow

None.

5.11.7 Diagnostics and Debug

If the internal print flag is set, ΔV_I, ΔV, and ΔV_{LV}, are displayed for reference.

5.11.8 Special Comments

None.

5.11.9 References

None.

TABLE 5.11-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine DVXYZ

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Preburn velocity vector
VPF	$\bar{V}_{PF}$	Real	I	fps	C	--	Postburn velocity vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Preburn velocity vector
DVX	ΔV_X	Real	O	fps	C	--	LVLH ΔV component - X
DVY	ΔV_Y	Real	O	fps	C	--	LVLH ΔV component - Y
DVZ	ΔV_Z	Real	O	fps	C	--	LVLH ΔV component - Z
DV	ΔV	Real	O	fps	C	--	ΔV magnitude

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.12 ROUTINE NAME - HIGHT

5.12.1 Purpose

HIGHT is the height maneuver and the circularization maneuver targeting routine.

5.12.2 Functional Description

Given the vehicle pre-burn state vector at the maneuver time, HIGHT iterates on the velocity required to achieve an altitude Π radians from the maneuver point. Depending upon the input option, this results in either a circularization maneuver or a height maneuver. When the iteration tolerance is met, HIGHT then computes the post-burn velocity vector achieved by the burn.

5.12.3 Assumptions and Limitations

None.

5.12.4 Method

For the circularization and height maneuver, a first guess at the solution is obtained. Then, an iteration is performed to achieve the desired result. For the circularization maneuver, the first guess is from the circular velocity equation,

$$\bar{V}_{PF} = \left(\frac{\mu}{R_p}\right)^{1/2} \bar{X}_{LV},$$

where $\bar{V}_{PF}$ = final velocity vector
μ = Earth gravitational constant
R_P = magnitude of the position vector
X_{LV} = LVLH unit vector in X-direction.

For the height maneuver, the first guess at the velocity is

$$\bar{V}_{PF} = \left[\frac{2\mu}{R_P\left(1 + \frac{R_P}{RD}\right)}\right]^{1/2} \bar{X}_{LV},$$

where RD = magnitude of the desired position vector.

After the final velocity vector is computed, the state vector is propagated radians, and the radius magnitude is compared to the desired radius magnitude. If the miss tolerance is met, the iteration is complete. If the miss tolerance is not met, then the post-burn velocity magnitude is incremented and the iteration continues.

5.12.5 Routine Input/Output Variables

The input/output variables are given in Table 5.12-I.

5.12.6 Functional Logic Flow

The functional logic flow is given in Figure 5.12.1.

5.12.7 Diagnostics and Debug

If internal print is requested, the radius missed value and the velocity magnitude are listed for each iteration.

5.12.8 Special Comments

None.

5.12.9 References

Software Development Branch. Preliminary Detailed Requirements for the Orbital Maneuver Targeting Module for the Orbiter On-Board Computer. JSC IN 75-FM-4, rev. 1, July 11, 1975.

TABLE 5.12-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine HIGHT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Velocity vector
CD	C_D	Real	I	—	C	--	Coefficient of drag
AREA	--	Real	I	ft^2	C	--	Reference area
WHT	—	Real	I	lb	C	—	Vehicle mass
SPERT	--	Intg	I	—	C	--	Vector propagation flag
CMAX	—	Intg	I	—	C	--	Maximum iteration tolerance
DELH	Δh	Real	I	ft	C	--	Delta altitude desired
DX1	—	Real	I	fps	C	—	First guess for velocity stepsize
H	h	Real	I	ft	C	--	Desired altitude
HOPT	—	Intg	I	—	C	—	Height option flag
IAPS	--	Intg	I	—	C	--	Apsis option flag

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.12-I.- Concluded

Routine HIGHT

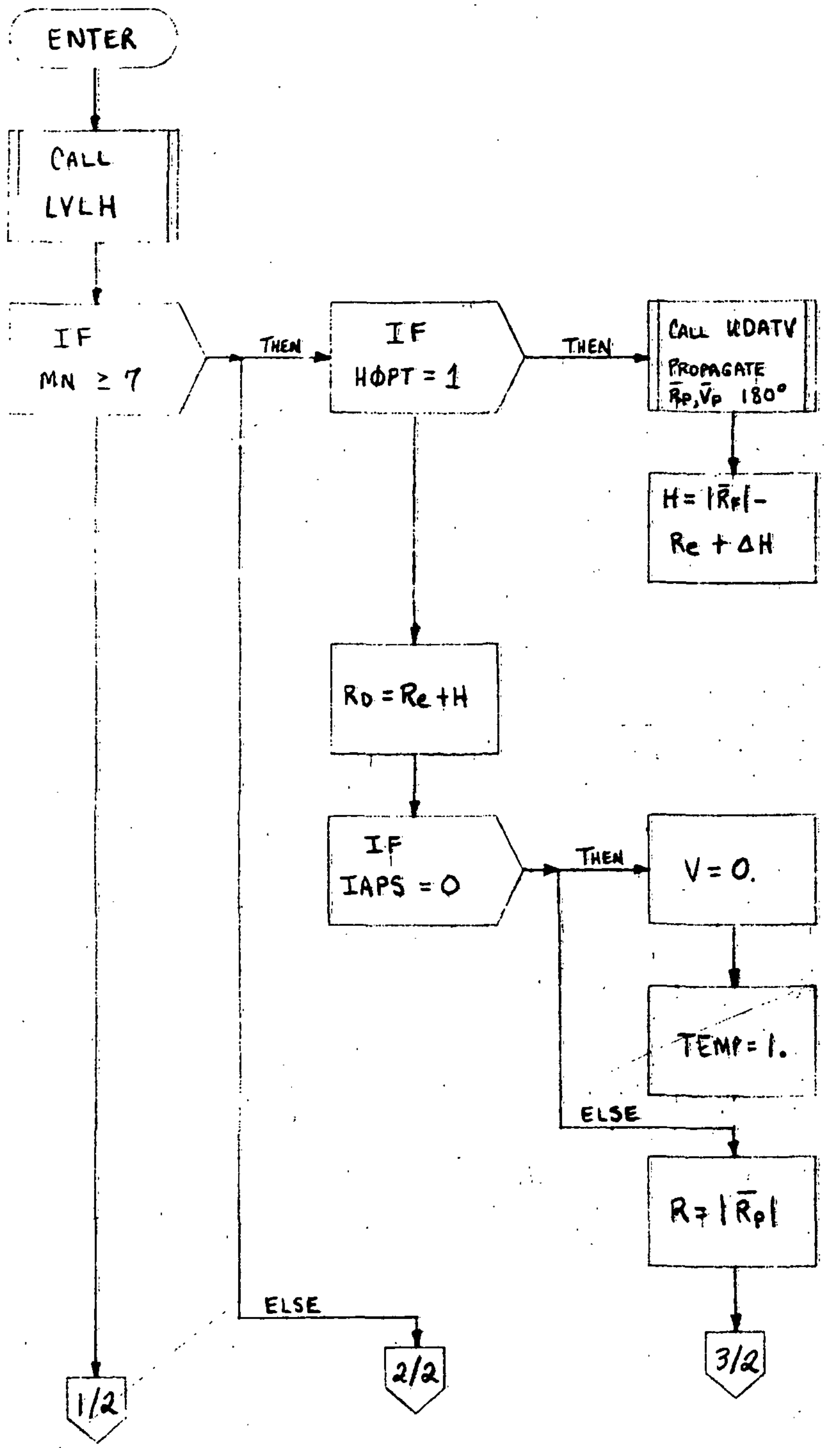
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
MN	--	Intg	I	—	C	—	Maneuver number flag
RTOL	--	Real	I	ft	C	--	Radius tolerance
VPF	$\bar{V}_{PF}$	Real	O	fps	C	—	Final velocity vector
ERROR	--	Intg	O	--	C	--	Error return code

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

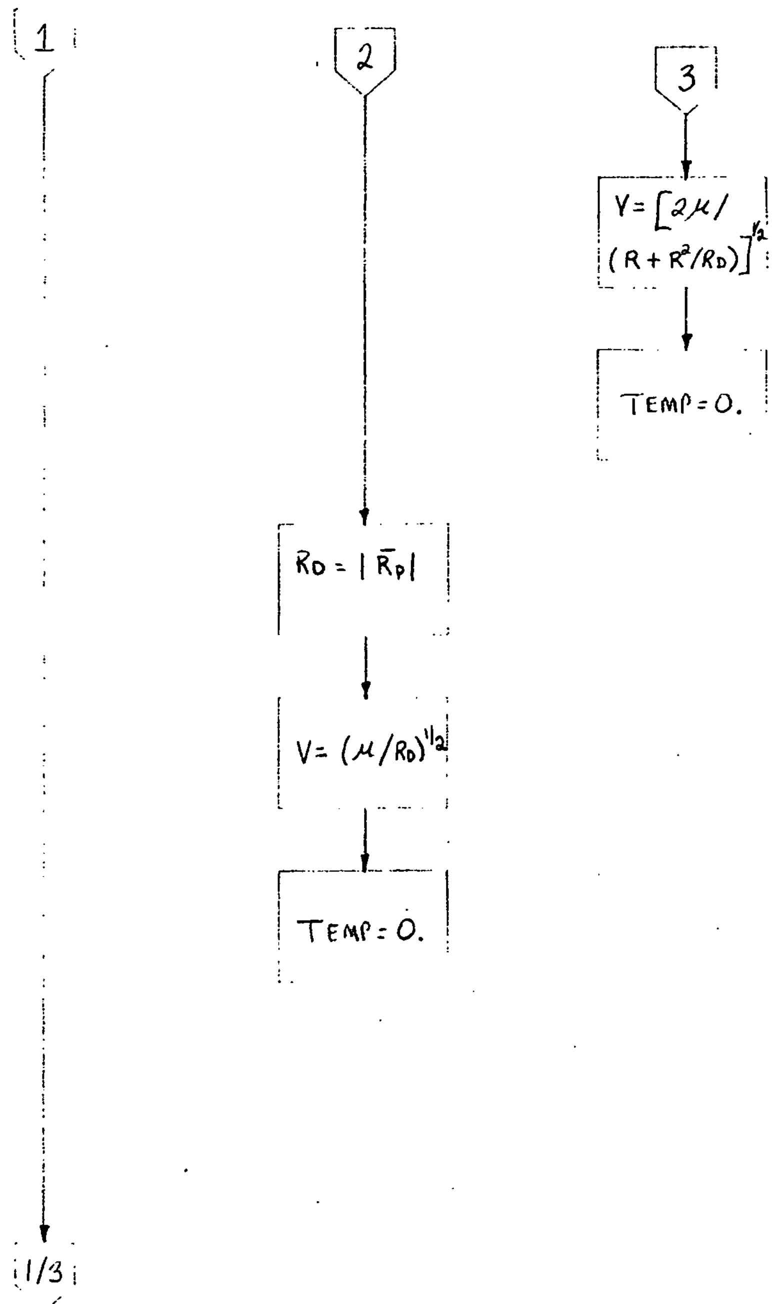
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



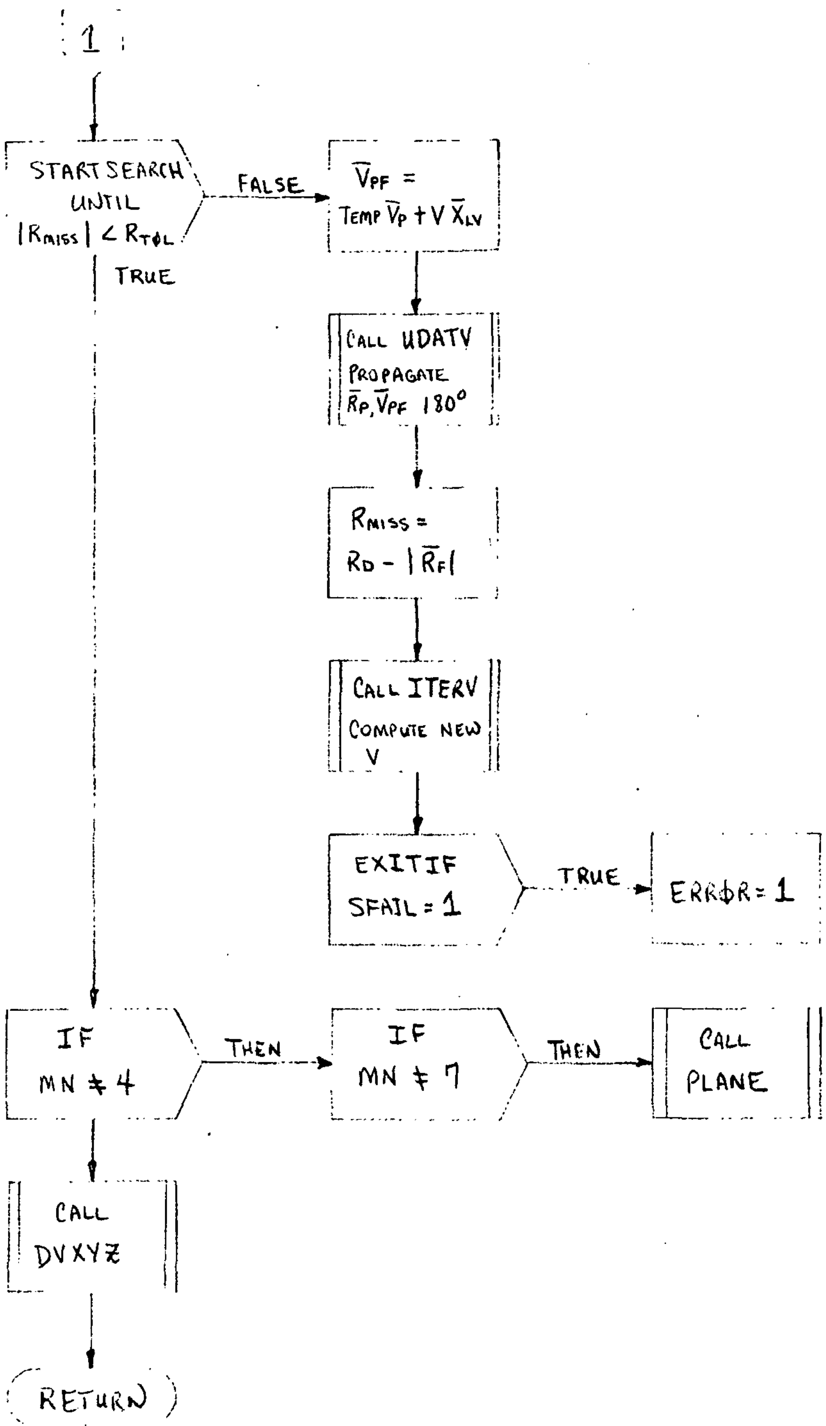
Page 1 of 3

Figure 5.12-1.- HIGHT functional logic flow.



Page 2 of 3

Figure 5.12-1.- Continued.



Page 3 of 3

Figure 5.12-1.- Concluded.

5.13 ROUTINE NAME - SHIFT

5.13.1 Purpose

SHIFT is the shift line of apsides maneuver targeting routine.

5.13.2 Functional Description

Given the vehicle pre-burn state vector at the maneuver time, SHIFT iterates on the minimum delta velocity required to accomplish the specified line of apsides shift.

5.13.3 Assumptions and Limitations

None.

5.13.4 Method

Since the required input to SHIFT does not define a unique solution (the post-burn element g, the argument of perigee, is the only element specified; the other five are free), SHIFT iterates on the minimum delta velocity magnitude (ΔV) solution. The computational sequence is as follows:

a. Assume $a_A = a_B$

$V_A = |\bar{V}_P|$,

where $\bar{V}_P$ = pre-burn velocity vector
a = semimajor axis
A = post-burn
B = pre-burn

compute $R = |\bar{R}_P|$
$f_A = f_B - \Delta f$
$a_{min} = (1/2)(R)(1 \pm \sin f_A)$,

where $\bar{R}_P$ = pre-burn position vector
f_A = true anomaly
Δf = input change in true anomaly.

b. Compute eccentricity, e.

c. Check for negative radicand in solution for e.

(1) if Radicand ≥ 0, continue to step d.

(2) if Radicand < 0, set $a_A = a_{min}$,

compute $$V = \left[\frac{\mu(2a_A - R)}{a_A R}\right]^{1/2},$$

and go to step b.

d. Check for negative e.

(1) if $e \geq 0$, go to step e.

(2) if $e < 0$, set

$a_A = R \pm \Delta R$,

recompute V, and go to step b.

e. Compute flight path angle, γ.

f. Compute final velocity vector, $\bar{V}_{PF}$.

g. Compute the ΔV.

h. Iterate on minimum ΔV.

(1) if ΔV minimum, final solution found.

(2) if ΔV not minimum, set
$a_A = a_A + \Delta a$

recompute V, and go to step b.

The mathematical sequence is as follows:

a. Compute the desired argument of perigee:

$$\begin{aligned} f &= f_B - \Delta f \\ g &= U - f \\ &= U_B - f_B + \Delta f \\ &= g_B + \Delta f \end{aligned}$$

b. Compute the eccentricity:

$$\cos E = \left(\frac{a - R}{ae}\right) = \left(\frac{e + \cos f}{1 + e \cos f}\right)$$

or $ae^2 + (R \cos f)e + (R - a) = 0$.

Since $a \neq 0$, the solutions are

$$e = \{-R \cos f \pm [(R \cos f)^2 - (4a)(R-a)]^{1/2}\}/(2a).$$

When $\cos f > 0$, use the positive radical value.

When $\cos f < 0$, use the negative radical value.

c. Test for a negative radicand.

The minimum semimajor axis, a, required for the radicand to be non-negative is found when

$$(R \cos f)^2 - (4a)(R - a) = 0.$$

Solving for a,

$$a = (1/2)(R)(1 \pm \sin f)$$

Since $R \geq r_p$, using the equation $a = \dfrac{r_a + r_p}{2}$, where r_a is the radius of apogee and r_p is the radius of perigee, the minimum value that a can be is $a = 1/2R + 1/2r_a$. Thus, choose $(1 + \sin f)$ when $\sin f \geq 0$. Then, $1/2R \leq a \leq R$. Also, choose $(1 - \sin f)$ when $\sin f \leq 0$. Then, $1/2R \leq a \leq R$.

d. Test for a negative eccentricity.

If e is negative, the minimum value for the semimajor axis, a, that is required for e to be positive is found when

$$-R \cos f \pm [(R \cos f)^2 - (4a)(R-a)]^{1/2} > 0, \ a \neq 0.$$

Two solutions exist:

(1) When $\cos f > 0$,

$$-R \cos f + [(R \cos f)^2 - (4a)(R-a)]^{1/2} > 0,$$

or $a > R$.

(2) When cos f < 0,

$$- R \cos f - \left[(R \cos f)^2 - (4a)(R-a) \right]^{1/2} > 0,$$

or a < R.

5.13.5 <u>Routine Input/Output Variables</u>

The input/output variables are given in Table 5.13-I.

5.13.6 <u>Functional Logic Flow</u>

The functional logic flow is given in Figure 5.13.1.

5.13.7 <u>Diagnostics and Debug</u>

If the internal print flag is set, SHIFT will print out various parameters at key steps in the code so that the logic flow may be followed.

5.13.8 <u>Special Comments</u>

None.

5.13.9 <u>References</u>

None.

TABLE 5.13-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine SHIFT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Preburn velocity vector
CMAX	—	Intg	I	—	C	—	Maximum iteration tolerance
DELF	Δf	Real	I	rad	C	--	Delta true anomaly
DV	ΔV	Real	I	fps	C	—	Delta velocity magnitude
VPF	$\bar{V}_{PF}$	Real	O	fps	C	--	Postburn velocity vector
ERROR	—	Intg	O	—	C	—	Error return code

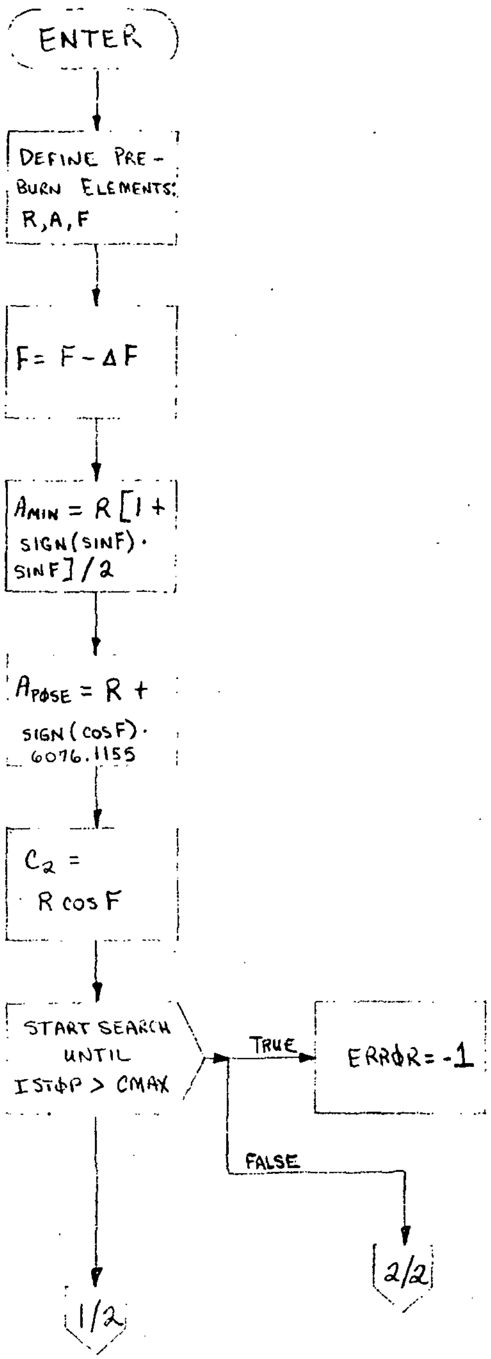
NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

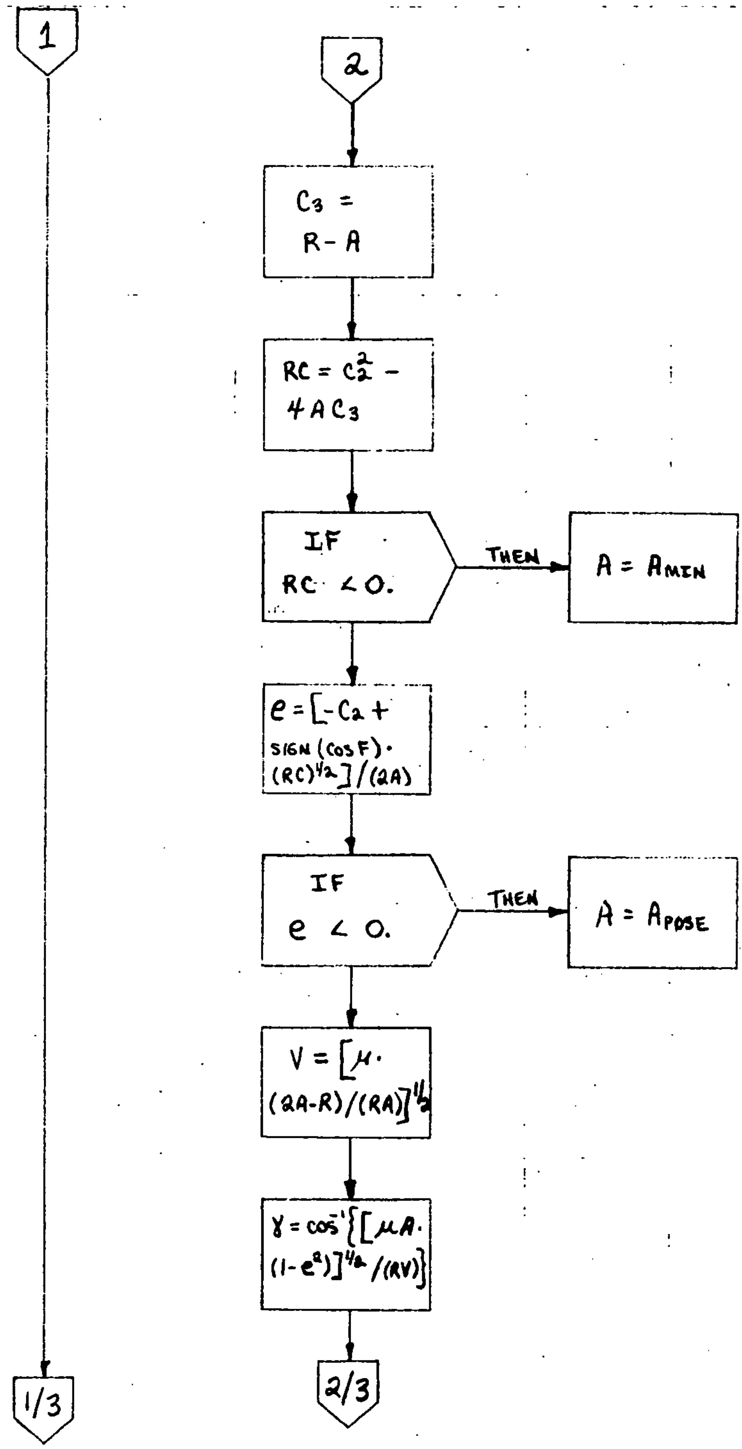
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



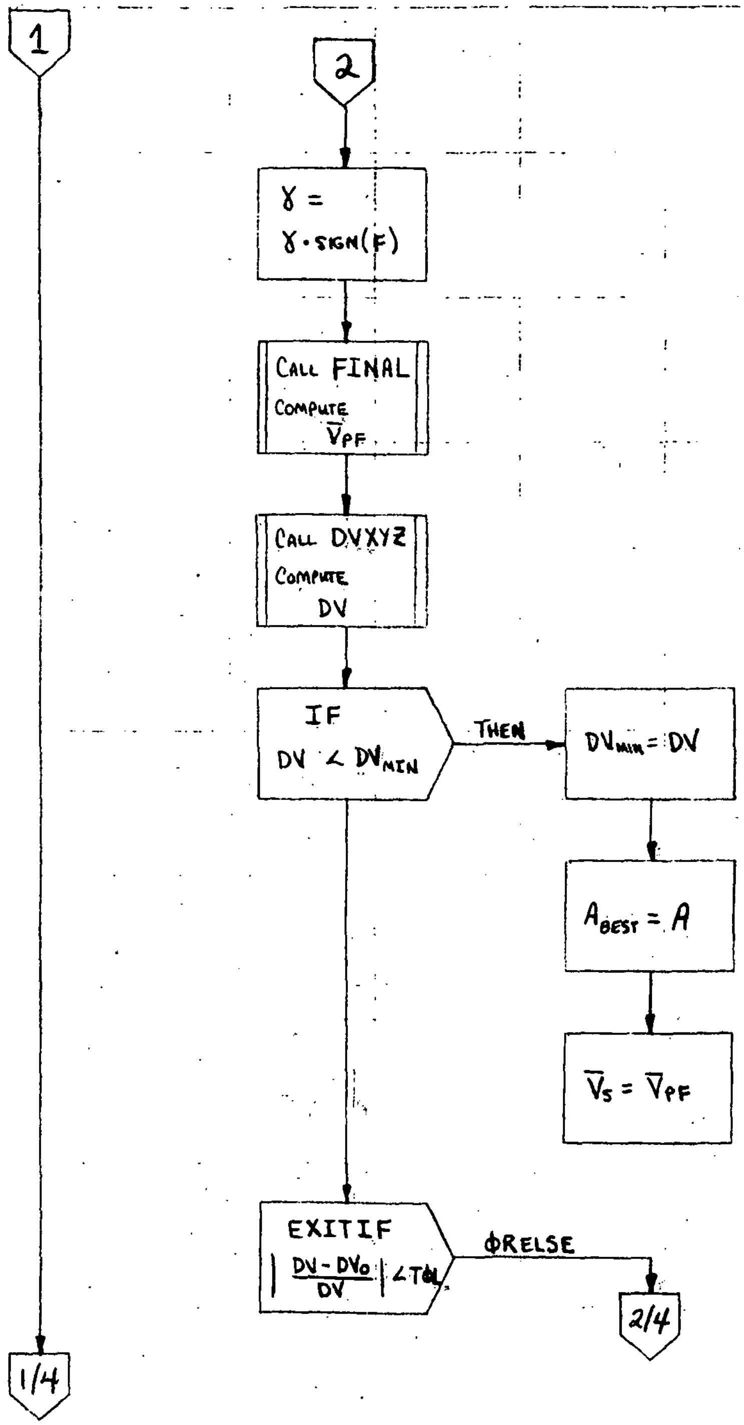
Page 1 of 4

Figure 5.13-1.- SHIFT functional logic flow.



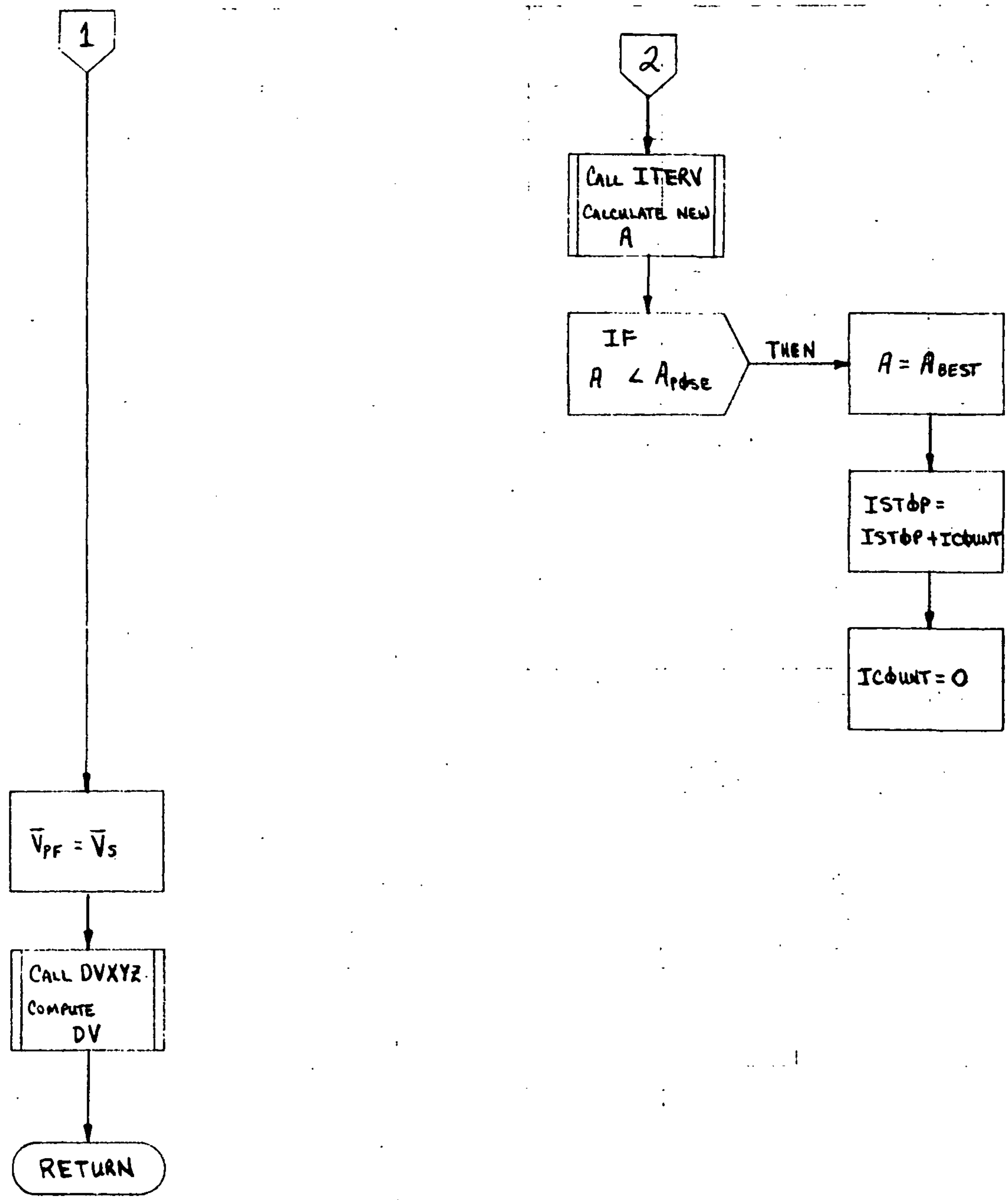
Page 2 of 4

Figure 5.13-1.- Continued.



Page 3 of 4

Figure 5.13-1.- Continued.



Page 4 of 4

Figure 5.13-1.- Concluded.

5.14 ROUTINE NAME - APSIS

5.14.1 Purpose

APSIS is the shift line of apsides and maintain apogee and perigee maneuver targeting routine.

5.14.2 Functional Description

Given the vehicle pre-burn state vector at the maneuver time, APSIS computes the new flight path angle required to accomplish the specified line of apsides shift. The resultant post-burn velocity vector is then determined.

5.14.3 Assumptions and Limitations

A positive shift in the line of apsides increases the argument of perigee.

5.14.4 Method

Since it is desired to maintain the pre-burn apogee and perigee, there are only two points in an orbit at which the line of apsides may be shifted the specified amount. The true anomaly, f, before the maneuver is

$$f_b + n\pi + 1/2\ \Delta f, \text{ for } n = 0 \text{ or } 1,$$

where Δf is the change in true anomaly.

Following the maneuver, the value of the true anomaly is

$$f_a = n\pi - 1/2\ \Delta f, \text{ for } n = 0 \text{ or } 1.$$

Thus, the unique solution for the apsidal shift is determined by the conditions

$$V_a = V_b$$
$$\gamma a = -\gamma b$$
$$\psi a = \psi b$$
$$R_a = R_b$$
$$\lambda_a = \lambda_b$$
$$\phi_a = \phi_b$$

where
V = velocity magnitude
γ = flight path angle
ψ = azimuth
R = radius magnitude
λ = longitude of maneuver
ϕ = geocentric latitude of maneuver
a = pre-burn orbit
b = post-burn orbit

This maneuver type is depicted in Figure 5.14-1.

5.14.5 Routine Input/Output Variables

The input/output variables are given in Table 5.14-I.

5.14.6 Functional Logic Flow

The functional logic flow is given in Figure 5.14-2.

5.14.7 Diagnostics and Debug

If internal print is desired, the semimajor axis, radius and velocity magnitudes, the true anomaly, and the flight path angle are displayed.

5.14.8 Special Comments

None.

5.14.9 References

None.

TABLE 5.14-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine APSIS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	—	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	—	Preburn velocity vector
DELF	Δf	Real	I	rad	C	—	Delta true anomaly
VPF	$\bar{V}_{PF}$	Real	O	fps	C	—	Postburn velocity vector

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

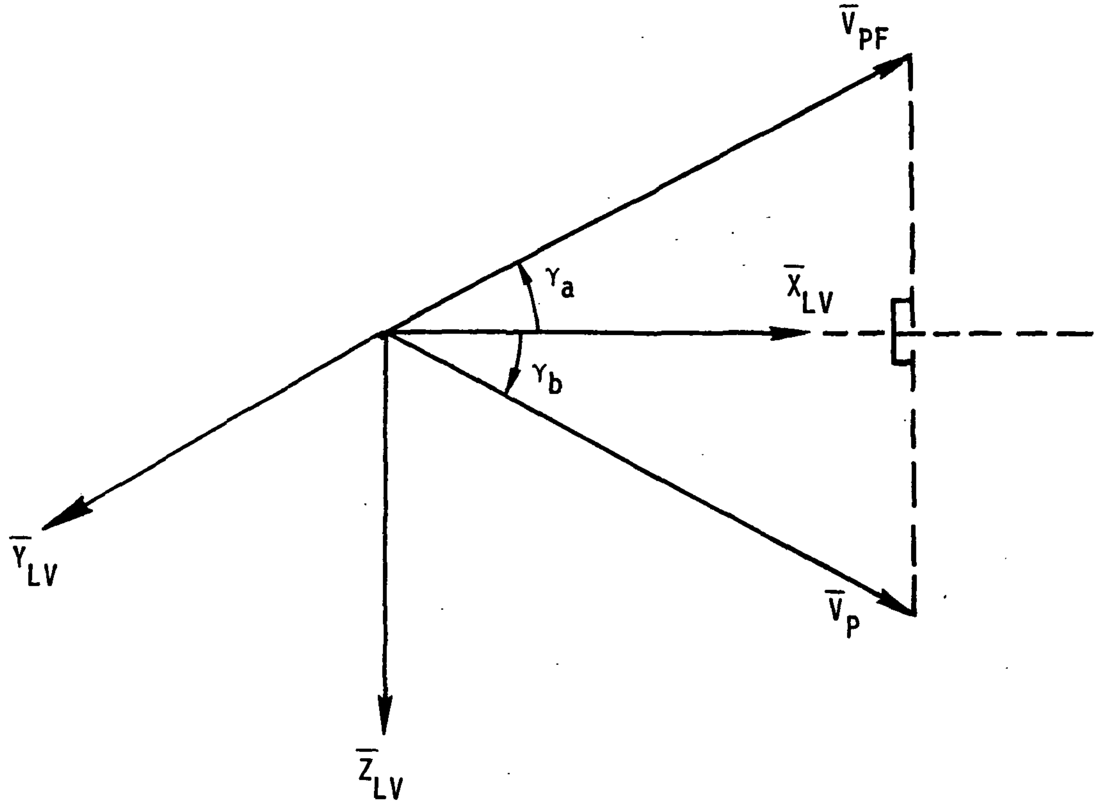
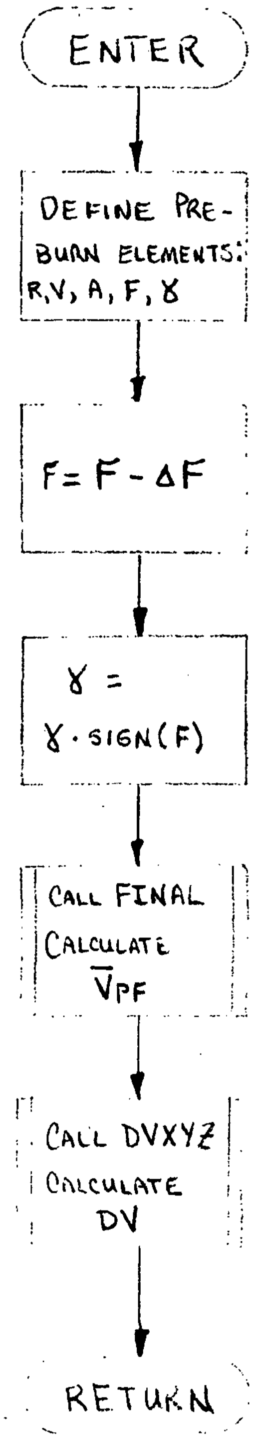


Figure 5.14-1.- Shift line of apsides, maintain apogee and perigee.



Page 1 of 1

Figure 5.14-2.- APSIS functional logic flow.

5.15 ROUTINE NAME - CHNGE

5.15.1 Purpose

CHNGE is the change apogee and perigee maneuver targeting routine.

5.15.2 Functional Description

Given the vehicle pre-burn state vector at the maneuver time, CHNGE computes the new semimajor axis, eccentricity, velocity magnitude, and flight path angle to determine the post-burn velocity vector which will achieve the specified apogee and perigee values.

5.15.3 Assumptions and Limitations

None.

5.15.4 Method

In order to find the post-burn velocity vector, the following quantities must be determined.

a.

$$a = (1/2)(r_a + r_p)$$
$$e = (R_a/a) - 1$$
$$v = [\mu(2/r - 1/a)]^{1/2},$$

where
- a = semimajor axis
- r_a = radius of apogee
- r_p = radius of perigee
- e = eccentricity
- V = velocity magnitude
- r = radius magnitude
- μ = Earth gravitational constant

b. And, $\gamma = \cos^{-1} \{[1/(Rv)][(\mu)(a)(1 - e^2)]^{1/2}\}$,

where γ = flight path angle. Gamma should have the same sign as the pre-burn true anomaly, which is measured

$$-\pi \lesssim f \lesssim \pi.$$

5.15.5 Routine Input/Output Variables

The input/output variables are given in Table 5.15-I.

5.15.6 Functional Logic Flow

The functional logic flow is given in Figure 5.15.1.

5.15.7 Diagnostics and Debug

If internal print is desired, the radius magnitude, the semimajor axis, the eccentricity, the velocity magnitude, the true anomaly, and the flight path angle are displayed for reference.

5.15.8 Special Comments

None.

5.15.9 References

None.

TABLE 5.15-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine CHNGE

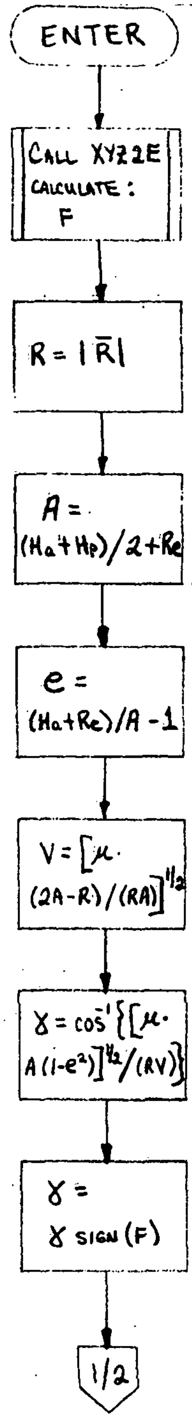
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Preburn velocity vector
HA	ha	Real	I	ft	C	--	Height of apogee
HP	hp	Real	I	ft	C	--	Height of perigee
VPF	$\bar{V}_{PF}$	Real	O	fps	C	--	Postburn velocity vector

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 2

Figure 5.15-1.- CHNGE functional logic flow.

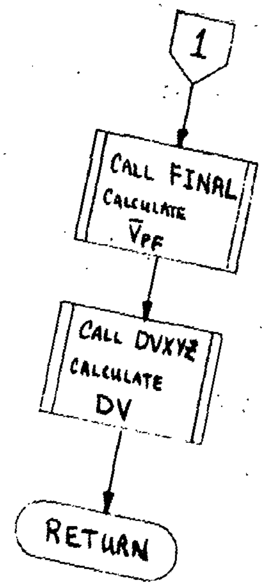


Figure 5.15-1.- Concluded.

Page 2 of 2

5.16 ROUTINE NAME - GPMDS

5.16.1 GPMDS is the General Purpose Maneuver Processor Display routine.

5.16.2 Functional Description

GPMDS writes the GPMP display to the user specified logical output unit. See table 4-VI of this processor for format and contents of the display.

5.16.3 Assumptions and Limitations

None.

5.16.4 Method

None.

5.16.5 Routine Input/Output Variables

The GPMDS input/output variables are presented in table 5.16-I.

5.16.6 Functional Logic Flow

None.

5.16.7 Diagnostics and Debug

None.

5.16.8 Special Comments

None.

5.16.9 References

None.

TABLE 5.16-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine GPMDS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	--	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Preburn velocity vector
T	t	Real	I	sec	C	--	Vector time
VPF	$\bar{V}_{PF}$	Real	I	fps	C	--	Postburn velocity vector
SESCON	--	Intg	I	--	C	!SESCN	Session constants
AREA	--	Real	I	ft^2	C	--	Vehicle reference area
CD	C_D	Real	I	--	C	--	Coefficient of drag
DV	ΔV	Real	I	fps	C	--	Delta velocity magnitude
DVX	ΔV_X	Real	I	fps	C	--	LVLH delta-V component - X direction
DVY	ΔV_Y	Real	I	fps	C	--	LVLH delta-V component - Y direction
DVZ	ΔVZ	Real	I	fps	C	--	LVLH delta-V component - Z direction
MN	--	Intg	I	--	C	--	Maneuver-type number
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.16-I.- Concluded

Routine GPMDS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
SARCH	—	Intg	I	—	C	—	Maneuver location selection flag
SPERT	--	Intg	I	—	C	--	Vector propagation code
TLO	—	Real	I	sec	C	--	GMT of lift-off
WHT	--	Real	I	lb	C	--	Vehicle preburn weight
DTB	Δt_B	Real	O	sec	C	—	Burn duration
HA	h_a	Real	O	ft	C	--	Height of apogee
HMAN	h_{MAN}	Real	O	ft	C	—	Height of maneuver
HP	hp	Real	O	ft	C	--	Height of perigee
ID	—	Real	O	—	C	—	Postburn inclination (in external units)
WHTF	--	Real	O	lb	C	—	Postburn weight
GPMP DISPLAY	--	Mix	O	—	T	--	See table 4-VI of this processor for definition of contents.

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.17 ROUTINE NAME - EXDV

5.17.1 Purpose

EXDV computes the external delta velocity burn information.

5.17.2 Functional Description

Given the vehicle pre-burn weight, the time of the impulsive burn, and the impulsive maneuver delta velocity magnitude, EXDV computes vehicle post-burn weight and the estimated burn duration.

5.17.3 Assumptions and Limitations

None.

5.17.4 Method

The weight flow, $\dot{W}$, is defined to be

$$\dot{W} = \text{Thrust}/I_{sp},$$

where Thrust = engine thrust value

and I_{sp} = engine specific impulse, in seconds

Then, the weight after the maneuver, W_f, is found by

$$W_f = W_i \, e^{-\Delta V/(I_{sp} \, G)},$$

where W_i = initial weight
ΔV = impulsive maneuver delta velocity magnitude
G = acceleration at sea level due to gravity

Finally, the estimated burn time, t_{burn}, is calculated by

$$t_{burn} = |W_f - W_i|/\dot{W}.$$

5.17.5 Routine Input/Output Variables

The EXDV input/output variables are presented in Table 5.17-I.

5.17.6 Functional Logic Flow

None.

5.17.7 Diagnostics and Debug

None.

5.17.8 Special Comments

None.

5.17.9 References

None.

TABLE 5.17-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine EXDV

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GSL	--	Real	I	ft/sec^2	C	--	Acceleration due to gravity at sea level
ISP	I_{SP}	Real	I	sec	C	--	Specific impulse
THRUST	--	Real	I	lb	C	--	Engine thrust
WHT	W_I	Real	I	lb	A	--	Vehicle preburn weight
TI	t_I	Real	I	sec	A	--	Time of impulsive burn
VGMAG	V_G	Real	I	fps	A	--	Impulsive maneuver delta-V magnitude
WHTF	W_F	Real	O	lb	A	--	Vehicle postburn weight
TIG	t_{IG}	Real	O	sec	A	--	Finite burn ignition time
TBURN	t_{BURN}	Real	O	sec	A	--	Estimated burn duration

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.18 ROUTINE NAME - GPMOT

5.18.1 Purpose

GPMOT is the General Purpose Maneuver Processor Output Tables routine.

5.18.2 Functional Description

GPMOT assigns and writes the appropriate data required for the maneuvering vehicle phase table (ref. 1) and the GPMP output summary table.

5.18.3 Assumptions and Limitations

None.

5.18.4 Method

None.

5.18.5 Routine Input/Output Variables

The GPMOT input/output variables are given in table 5.18-I.

5.18.6 Functional Logic Flow

None.

5.18.7 Diagnostics and Debug

None.

5.18.8 Special Comments

None.

5.18.9 Reference

1. Flight Design System-1, System Design Document, Standards. Vol. VI, JSC IN 77-FM-18, rev. 1, January 1978.

TABLE 5.18-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine GPMOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RP	$\bar{R}_P$	Real	I	ft	C	—	Position vector
VP	$\bar{V}_P$	Real	I	fps	C	—	Preburn velocity vector
T	t	Real	I	sec	C	—	Vector time
VPF	$\bar{V}_{PF}$	Real	I	fps	C	—	Postburn velocity vector
SESCON	—	Intg	I	—	C	—	Session constants
BVEC	—	Real	I	—	C	—	Base vector
DTB	—	Real	I	sec	C	—	Burn duration
ERROR	—	Intg	I/O	—	C	—	Error return code
HA	ha	Real	I	ft	C	—	Height of apogee
HMAN	h_{MAN}	Real	I	ft	C	—	Height of maneuver
HP	hp	Real	I	ft	C	—	Height of perigee
ID	I_D	Real	I	—	C	—	Inclination (in user external units)

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.18-I.- Continued

Routine GPMOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
THRUST	—	Real	I	lb	C	—	Engine thrust value
ISP	I_{SP}	Real	I	sec	C	—	Engine specific impulse
MN	—	Intg	I	—	C	—	Maneuver-type code
DVX	ΔV_X	Real	I	fps	C	—	LVLH delta-V component - X direction
DVY	ΔV_Y	Real	I	fps	C	—	LVLH delta-V component - Y direction
DVZ	ΔV_Z	Real	I	fps	C	—	LVLH delta-V component - Z direction
DV	ΔV	Real	I	fps	C	—	Delta-V magnitude
YAW	Y	Real	I	rad	C	—	Yaw angle (LVLH)
PITCH	P	Real	I	rad	C	—	Pitch angle (LVLH)
TLO	t_{LO}	Real	I	sec	C	—	GMT lift-off
SARCH	—	Intg	I	—	C	—	Maneuver location selection flag
SPERT	—	Intg	I	—	C	—	Vector propagation flag

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.18-I.- Concluded

Routine GPMOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
CD	C_D	Real	I	—	C	--	Coefficient of drag
AREA	--	Real	I	ft^2	C	--	Reference area
WHT	W_I	Real	I	lb	C	--	Preburn weight
ENGID	--	Intg	I	--	C	--	Engine identification code
WHTF	W_F	Real	I	lb	C	—	Postburn weight
PVTAB	--	Mix	O	—	F	--	Position/velocity phase table. See SSD volume VI, section 1.2.3.4 for definition of contents.
MNCODE	--	6CH	O	—	F	—	Maneuver ID and location codes
ISUM	--	Mix	O	—	F	--	Output summary table. See table 4-III of this processor for definition of contents.

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

GROUNDTRACK PROCESSOR (GTRAK)

1.0 PURPOSE

The groundtrack processor (GTRAK) produces a vehicle ephemeris in geodetic latitude versus longitude form for either a tabular or graphic display. If the graphic display is selected, the output appears as an orbital groundtrack superimposed on a map of the world.

2.0 FUNCTIONAL

The processor accepts interface table inputs that define the display mode, vehicle ephemeris generation control parameters, and display control parameters.

The two display modes operate independently. When in tabular display mode, the user provides the start/stop option (time or orbit count), the starting and stopping limits, and the time step. The time, orbit count, latitude, and longitude are output at each time step. Upon completion of a page of the display, the user is prompted with a "CONTINUE?" prompt.

When in graphic mode, the user provides the same inputs as in tabular display mode, but the time step specifies the spacing of the time tics on the groundtrack. The actual time step at which the latitude and longitude computations are performed is computed internally such that it is evenly divisible into the specified time tic interval, and the intervals between the data points on the groundtrack are approximately four degrees. In addition, the user specifies the latitude and longitude limits of the map to be generated, and the groundtrack display is automatically limited to that area.

The processor will either compute the size and location of the map on the screen such that its size is maximized, or allow the user to specify the size and location. If the latter option is selected, the user utilizes the graphics input crosshairs to specify the locations of the lower left and upper right corners of the map.

With either option the user must establish the correlation between time and orbit number. This is accomplished by inputting a time relative to the user-established base date, and the orbit number corresponding to that time.

The latitude and longitude at each time point is computed by applying the angular rates in the phase table to the nearest phase table state vector to obtain the classical elements at that time. The state vector is converted first to Cartesian form, and then to geodetic polar form to obtain the desired latitude and longitude.

The specified start time must not precede the time of the first state in the phase table. The specified stop time may be later than the last state. No output will be produced outside of the range defined by the start and stop limits. However, if the time at which the time/orbit number correlation is defined precedes the start time, the computation will begin at the earlier time, with output suppressed until the start time is reached.

The latitude/longitude start/stop capability is available only in graphic mode, and is provided through the user control on the map limits. Only the portions of the orbit groundtrack that appear within the map limits are plotted. The time or orbit count limits are also active in this case, giving the user the capability to display a certain portion of a specified range of orbits.

If the specified start/stop limits enclose a number of orbits greater than the specified maximum per plot, the processor pauses upon completion of a plot and prompts the user with a "continue" prompt. This places paging and hardcopying under direct user control. The user may also terminate execution at the end of any plot.

The orbit numbers are placed at both the ascending and descending nodes. If a small map segment is requested such that the display does not contain any nodal crossings, the orbit numbers appear at the grid boundaries.

3.0 ASSUMPTIONS AND LIMITATIONS

a. When graphic display is selected, the margin control switch on the Tektronix must not be set on 1.

b. A vehicle ephemeris in the form of a position/velocity phase table must have been produced using INVAR.

c. The user must provide the initial time/orbit count relationship.

d. Only shorelines are plotted. No political boundaries are available.

e. Mercator map projection is utilized.

f. The maximum northerly latitude grid limit must be greater than the maximum southerly latitude grid limit (i.e., LATMAX > LATMIN). This restriction does not apply to the longitude limits.

g. Latitude grid limits must be limited to $\pm 84^{\circ}$.

h. The graphic display is only generated at a Tektronix terminal.

i. The specified start time must not precede the first state vector time in the phase table.

j. The time at which the time/orbit count relationship is defined must not precede the first state vector time in the phase table, but may precede the start time.

k. The grid latitude and longitude limits are entered in degrees.

l. The printed display is output in degrees.

m. The graphics display latitude and longitude grid lines are labeled in degrees.

4.0 PROCESSOR INPUT/OUTPUT

The user/processor communication is both through the interface table and solicited prompts. The primary communication is via the interface table.

a. Processor interface table - All user control parameters are passed to the processor via the interface table with the single exception noted under 4-d. The constants required for the geodetic latitude computation are obtained from !!GLCN. Those program control parameters that may require user access are contained in PROCON.

   The definitions of the GTRAK processor interface table parameters are provided in table 4-I.

b. Interface table data array definitions - The definitions of the input/output data arrays appearing in the GTRAK interface table are provided in table 4-II.

c. Interface table data file definitions - The contents of the PVTAB phase table are shown in figure 1.2-13 in the FDS-1 Standards document. The world map data file is described in table 4-III.

d. Interface table extended prompts - The interface table extended prompts are provided in table 4-IV.

e. Processor solicited (prompted) inputs - The processor solicited prompts are shown in table 4-V. In addition to the prompts, the size and location of the graphic display on the screen can be defined by using the graphics input crosshairs. The processor prompts the user for the location of the lower left corner, and then activates the crosshairs. The user positions the crosshairs to the desired location and enters "space, carriage return." The process is repeated for the upper right corner.

f. Processor displays and display parameter definition table - The tabular display format is shown in table 4-VI, and its parameter definitions are shown in table 4-VII(a). The graphic display format is shown in table 4-VII(b), and its parameter definitions are shown in table 4-VII(c).

   Note: When using the graphic display, the margin control switch on the Tektronix terminal must not be set on 1. If it is set on 1, the page number will not appear on the display, and no groundtrack is produced until the switch is changed.

g. Processor message table - The processor messages are shown in table 4-VIII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR GTRAK

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Value stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants
GRDLNS	AWA	2CH	I	1	1	"YE"	Lat/long grid line option
GRDOPT	AWA	2CH	I	1	1	"YE"	Automatic grid boundary
LATMAX	AWA	Real	I	1	1	70.	North latitude limit
LATMIN	AWA	Real	I	1	1	-60.	South latitude limit
LONMAX	AWA	Real	I	1	1	180.	Right longitude limit (between ± 180°)
LONMIN	AWA	Real	I	1	1	-180.	Left longitude limit (between ± 180°)
LUO	AWA	Intg	I	1	1	0	Print output unit
MAPNAM	AWA	6CH	I	3	1	$ERTHG	World map data file
ORBCNT	AWA	Intg	I	1	1		Orbit count at I = TIMORB
ORBPPG	AWA	Intg	I	1	1	8	Number of orbits per plot
OSTART	AWA	Intg	I	1	1		Starting orbit number

NOTES

CLASS	TYPE			USE
AWA	Free	2CH	72CH	I = Input
Disk	Intg	6CH	Mix	O = Output
	Real	18CH	Symb	I/O = Input/Output
	Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR GTRAK

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Value stored in default interface table	Definition
OSTOP	AWA	Intg	I	1	1		Final orbit number
OUTOPT	AWA	6CH	I	3	1	"GRAPH"	Display mode option
PROCON	AWA	Intg	I	6	6	31,20,20,10,50,6	Program constants
PVTAB	Disk	Real	I	—	—		INVAR phase table
STOPTN	AWA	6CH	I	3	1	"TIME"	Start/stop option (time or orbit)
TSTEP	AWA	Real	I	3	3	0,5,0	Time step (hr, min, sec)
TIMORB	AWA	Real	I	3	3		GET corresponding to ORBCNT
TITLE	AWA	36CH	I	18	1		Display title
TSTART	AWA	Real	I	3	3		Starting time
TSTOP	AWA	Real	I	3	3		Final time

NOTES	CLASS	TYPE	USE
	AWA Disk	Free 2CH 72CH Intg 6CH Mix Real 18CH Symb Dubl 36CH	I = Input O = Output I/O = Input/Output

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR GTRAK

Array name	Index location	Default value	Definition
GLOCON	1-180	!!GLCN	Global constants array; master data base element
SESCON	1-90	!!SESCN	Input/output session constants array
PROCON	1	31	Logical unit number of graphics terminal
	2	20	Logical unit number of world map data file
	3	20	Logical unit number of the PVTAB phase table (DRDE)
	4	10	Minimum number of raster units between successive plotted points on the world map
	5	50	Maximum number of lines per page of tabular output display
	6	0	Debug print flag. Set = 1 for debug output.

TABLE 4-III.- INTERFACE TABLE DATA FILE DEFINITIONS

PROCESSOR GTRAK

DATA FILE MAPNAM

Record number	Integer word allocations	Content and definition
1-17	1-500	Latitudes of 500 shoreline data points, expressed in mercator map coordinates.
	501-1000	Longitudes of 500 shoreline data points, expressed in mercator map coordinates.
		All records are in the same format. All values are integer. Map discontinuities are indicated by the latitude and longitude, both having a value of 1. The end of data is indicated by the latitude and longitude, both having a value of 0. File contains 17 records.

TABLE 4-IV.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR GTRAK

Processor name	Processor abstract prompt (maximum 256 characters)
GTRAK	GTRAK produces either a tabular listing or a graphic plot of an orbit groundtrack. The graphic plot is overlayed onto a plot of the world map for the specified area.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
GLOCON	Name of FDS global constants array.
GRDLNS	Lat/long grid line option. "YE" requests grid lines, "NO" suppresses grid lines. Default value is "YE".
GRDOPT	Automatic grid boundary generation. If "YE", processor computes grid size and location. If "NO", user defines size using the crosshairs.
LATMAX	Latitude of top grid boundary (must be $\leq 84^{\circ}$).
LATMIN	Latitude of bottom grid boundary (must be $\geq -84^{\circ}$).
LONMAX	Longitude of right grid boundary (must be within $\pm$ 180°).
LONMIN	Longitude of left grid boundary (must be within $\pm$ 180°).
MAPNAM	Name of data file containing world shoreline data.
ORBCNT	Provides the correlation between orbit count and GET. The correlation is established by the orbit count in ORBCNT and the corresponding GET in TIMORB.
ORBPPG	Maximum number of orbits to be displayed per plot.

TABLE 4-IV.- Continued

PROCESSOR GTRAK

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
OSTART	Orbit number at which processing is to begin if orbit start/stop option is selected.
OSTOP	Orbit number at which processing is to terminate if orbit start/stop option is selected.
OUTOPT	Display mode option. Valid inputs are "GRAPH" and "PRINT".
PROCON	Program constants array.
PVTAB	Name of INVAR phase table.
STOPTN	Start/stop option. Valid inputs are "TIME" and "ORBIT".
TSTEP	Time step. If tabular display is selected, TSTEP is the frequency in time of the display line entries. If in GRAPH mode, TSTEP specifies the spacing of the time tics. Format is real (hr, min, sec).
TIMORB	The GET that corresponds to the orbit number in ORBCNT. Used for correlating orbit count to GET.
TITLE	Display title to appear at the top of each display. Maximum of 36 characters.
TSTART	GET at which processing is to begin if in TIME start/stop mode.
TSTOP	GET at which processing is to terminate if in TIME start/stop mode.

TABLE 4-IV.- Concluded

PROCESSOR GTRAK

Processor name	Processor abstract prompt (maximum 256 characters)

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
LUO	Printed output logical unit number. A value of zero causes the output to be sent to the user terminal.

TABLE 4-V.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR <u>GTRAK</u>

Prompt	Meaning	Valid responses
CONTINUE?:	Proceed to next display, or terminate processor.	"YE", "b̸" A response of "space-carriage return" may be substituted for the "YE" response. "b̸" terminates the processor.
ENTER LOWER LEFT CORNER:	Used when user is defining size and location of graphic display on the screen. User positions crosshairs at point where he wants the lower left corner, and enters "space, carriage return".	"Space-carriage return". GTRAK will respond with a vertical and a horizontal line through that point and extending to the screen boundaries. If the two lines do not appear, leave the crosshairs at the same position and repeat the "space-carriage return".
ENTER UPPER RIGHT CORNER:	Same as above, except for upper right corner.	See above.

TABLE 4-VI.- PROCESSOR DISPLAY TABLE

PROCESSOR <u>GTRAK</u>

```
   1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75
 1                            VEHICLE  EPHEMERIS  DISPLAY

                      αααααααααααααααααααααααααααααααααααα

 5
       TIME     ORBIT    LAT        LONG                 TIME     ORBIT    LAT        LONG

    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
10  XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
15  XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
20  XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
    XXX:XX:XX    XXX   ±XX.XX     ±XXX.XX              XXX:XX:XX   XXX   ±XX.XX     ±XXX.XX
24  XXX:XX:XX    XXX   +XX.XX     +XXX.XX              XXX:XX:XX   XXX   +XX.XX     +XXX.XX
       5    10   15   20   25   30   35   40   45   50   55   60   65   70   75
```

TABLE 4-VII.- DISPLAY PARAMETER DEFINITION TABLE

PROCESSOR GTRAK

(a) Vehicle ephemeris display

Vehicle Ephemeris Display	
Display parameter label	Parameter definition
TITLE	User-provided run identification (maximum of 36 characters) in parameter keyword TITLE.
TIME	GET in hours, minutes, and seconds.
ORBIT	Orbit number.
LAT	Geodetic latitude.
LONG	Longitude.

TABLE 4-VII.- Continued

PROCESSOR GTRAK

(b) Example of groundtrack

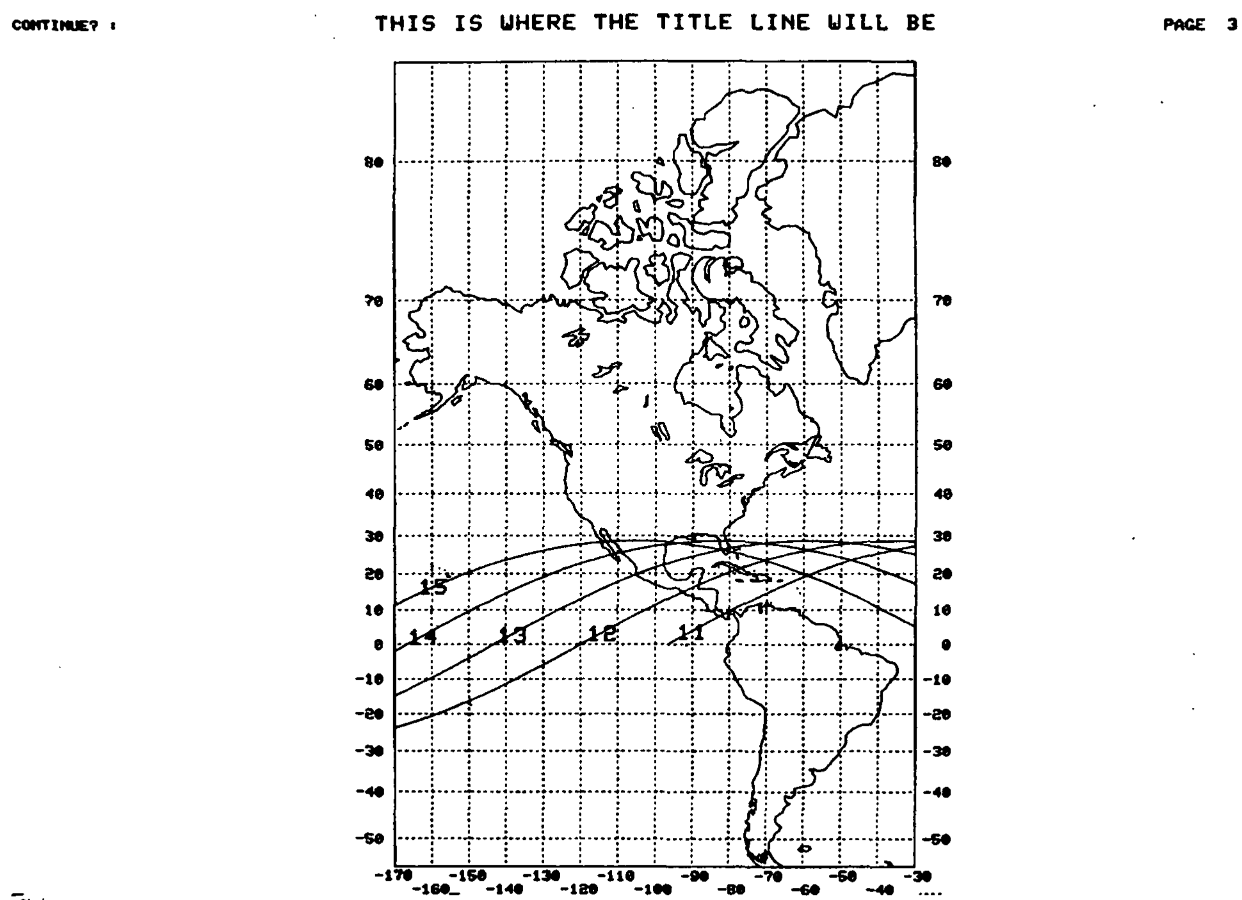


TABLE 4-VII.- Concluded

PROCESSOR GTRAK

(c) Vehicle groundtrack display

Vehicle Groundtrack Display	
Display parameter label	Parameter Definition
	The line of information across the top of the display is the user-supplied display title.
	The numbers along the left and right boundaries of the map are the latitudes of the latitude grid lines.
	The numbers across the bottom boundary are the longitudes of the longitude grid lines.
	The larger numbers adjacent to the groundtracks are the orbit numbers.
	The tic marks on the groundtracks are placed at each user-specified time step. Those tic marks which fall on the hour are twice as long as the others.

TABLE 4-VIII.- PROCESSOR MESSAGE TABLE

PROCESSOR GTRAK

MSG no.	Message ID block	Message text block and explanation
1	*GTRAK*	"LONGITUDE LIMITS MUST BE WITHIN THE RANGE $\pm$ 180°" Meaning: One or both limits invalid. Severity: Processor terminated. Action required by user: Correct and restart.
2	*GTRAK*	"LATITUDE LIMITS INVALID" Meaning: Either LATMIN $\geq$ LATMAX, or one is outside the range $\pm$ 84°. Severity: Processor terminated. Action required by user: Correct and restart.
3	*GTRAK*	"GRAPHIC OUTPUT IS REQUESTED FROM A NONGRAPHIC TERMINAL" Meaning: Self-explanatory. Severity: Processor terminated. Action required by user: Either change output option or change terminal.
4	*GTRAK*	"STOP LIMIT IS NOT GREATER THAN START LIMIT" Meaning: Self-explanatory. Severity: Processor terminated. Action required by user: Correct and restart.
5	*GTRAK*	"TABULAR DISPLAY IS REQUESTED BUT TSTEP $\leq$ 0" Meaning: Self-explanatory. Severity: Processor terminated. Action required by user: Correct and restart.
6	*GTRAK*	"UNABLE TO OPEN XXXXXX" Meaning: Named file cannot be opened. Severity: Processor terminated. Action required by user: Check correctness of name request. If correct, contact FM6.

TABLE 4-VIII.- Concluded

PROCESSOR GTRAK

MSG no.	Message ID block	Message text block and explanation
7	*GTRAK*	"ERROR READING XXXXXX" Meaning: Error encountered while attempting to read file XXXXXX. Severity: Processor terminated. Action required by user: Same as above.
8	*GTRAK*	"ORBCNT IS GREATER THAN OSTART" ORBCNT = XXXX OSTART = XXXX

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

CONDITIONAL EXECUTION PROCESSORS (IF/ELSE/ENDIF)

1.0 PURPOSE

The IF/ELSE/ENDIF utility processors, used as an ordered group, provide the FDS user with the capability to perform conditional execution of other processors within a sequence table.

2.0 FUNCTIONAL DESCRIPTION

The IF processor evaluates the relationship between two single precision real quantities and establishes a true or false condition based on a specified relationship between the quantities. If the condition is true, sequence execution continues with the sequential execution of subsequent processors within the sequence table. If an ELSE is then encountered, control is passed to the processor following the matching ENDIF in the sequence table.

When the evaluated condition is false, execution skips to the processor immediately following the matching ELSE, or the ENDIF in the event the ELSE is not found. Note that the ELSE need only be used if processors are to be executed under a false condition.

Example sequence table 1

```
  .
  .
  .
 IF  )
  .  |
  .  }   executed only on true condition
  .  )
 ELSE
  .  )
  .  }   executed only on a false condition
  .  )
 ENDIF
  .
  .
  .
```

Example sequence table 2

```
     .
     .
     .
    IF
     .)
     .} executed only on true condition
     .)
    ENDIF
     .
     .
     .
```

3.0 ASSUMPTIONS AND LIMITATIONS

Listed below are the assumptions and limitations associated with the IF/ELSE/ENDIF processors.

a. IF and ENDIF must be used as a pair to ensure correct conditional execution results. When execution of processors under a false condition is also desired, the ELSE processor must be used in a position between the IF and ENDIF in the sequence table. Omission of the ENDIF will result in an error message and sequence execution termination.

b. There are no interface tables for the ELSE and ENDIF processors.

c. IF groups are executed in a nested fashion, and they should appear as nests within the sequence tables. For example:

```
Correct nesting                                          Incorrect nesting

      IF                                                       IF
      .                                                        .
      .                                                        .
      .                                                        .
      IF                                                       IF
      .      }                                                 .
      .      }                                                 .
      .      }                                                 .
      ELSE   }  inner                                          ELSE
      .      }  group                                          .
      .      }                       } outer                   .
      .      }                       } group                   .
      ENDIF  }                                                 ELSE
      .                                                        .
      .                                                        .
      .                                                        .
      ELSE                                                     ENDIF
      .                                                        .
      .                                                        .
      .                                                        .
      ENDIF                                                    ENDIF
```

d. When executing IF in the semiautomatic mode, the user should be aware of the following:

   When an IF processor is overriden by the user with an IF and the results of the override are true, sequential execution will continue with the original IF sequence table entry. The user should skip this second IF when only a single conditional evaluation is desired. That is, only one level of IF nesting is to be performed.

   <u>Example of an override of an IF</u>:

```
%:  SEMI, TABLEΔ
$ 100 = IF, ITABLE:  IF, JTABLEΔ
$ 100 = IF, ITABLE:  200Δ
$200 = . . .
```

   The second opportunity to execute should be taken (via a carriage return) only if nesting of IF's is desired. Otherwise, a sequence number override should be entered.

e. Nesting errors involving IF, DO, and SCAN are not explicitly caught, but may be detected because of unmatched IFs, ELSE, DOs, or ENDDOs.

As an example:

```
#100 = IF, ITABLE
#200 = DO, DTABLE
#300 = ENDIF
#400 = ENDDO
```

An error will occur when the ENDDO is executed without having executed the DO <u>or</u> when the ENDIF is executed more than once.

f. Execution of IF, ELSE, or ENDIF in the MANUAL mode is not meaningful.

g. Comparison of the two real quantities using the equality (=) or inequality (#) relationships may not always be meaningful. Equality of two real values requires that they be identical in all 32 bits. This is seldom the case if either operand results from any form of arithmetical computation.

4.0 PROCESSOR INPUT/OUTPUT

Listed below are the inputs required for the IF processor.

a. Processor interface table - The interface table for IF is defined in table 4-I and contains

   - Two single precision real values to be compared
   - A relationship to be applied to the two values

b. Interface table data array definitions - None.

c. Interface table data file definition - None.

d. Processor solicited (prompted) inputs - None.

e. Processor displays and display parameter definitions - None.

f. Processor message table - The IF/ELSE/ENDIF error messages are provided in table 4-II.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-III.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR IF/ELSE/ENDIF

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
OPRND1	AWA	Real	I	1	1	--	First quantity to be used in relation evaluation; symbol for relation condition from the set:
RELATN	AWA	2CH	I	1	1	--	# not equal < less than <= less than or equal =< less than or equal = equal >= greater than or equal => greater than or equal > greater than
OPRND2	AWA	Real	I	1	1	--	Second quantity to be used in relation evaluation

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- PROCESSOR MESSAGE TABLE

PROCESSOR IF/ELSE/ENDIF

MSG no.	Message ID block	Message text block and explanation
1	*IF01*	"IF CANNOT BE EXECUTED WITHOUT A MATCHING ENDIF" Meaning: Execution of IF was attempted without subsequent appearance of matching ENDIF in the sequence table. This condition is detected on a false condition if ELSE is not used or a true condition if ELSE is used. Severity: Fatal; sequence execution is aborted Action required by user: Use the sequence table editor to correct the sequence and supply the missing ENDIF. *Also, check for improper nesting structures.*
2	*IF02*	"IF AND ELSE MAY NOT BE EXECUTED IN THE MANUAL MODE" Meaning: Since it is only meaningful to perform conditional execution in a sequence of processors, IF and ELSE may only be used in the SEMIAUTOMATIC and AUTOMATIC modes of execution. Severity: Fatal; processor execution is aborted. Action required by user: Use the sequence table editor to build the desired processor sequence and execute the sequence.
3	*IF03*	"A VALID RELATIONAL OPERATOR MUST BE USED IN IF: #, <=, <, =, >=, >" Meaning: An invalid operator was input to IF in the RELATN argument. Severity: Fatal; sequence execution is aborted. Action required by user: Reedit the interface table for IF, correcting the error. One of #, >, >=, =>, =, <=, =<, and > must be used.
4	*IF04*	"SYSTEM ERROR IN IF OR ELSE. INVALID &SEQTAB" Meaning: The IF or ELSE processor is unable to retrieve an executive table from the FDS manager. This error is indicative of a software or hardware problem; this is not a user error. Severity: Fatal, sequence execution is aborted. Action required by user: Notify FDS support management

TABLE 4-III.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR IF

Processor name	Processor abstract prompt (maximum 256 characters)
IF	IF, when used with ELSE and ENDIF, defines a logical structure within a sequence table. Processors following the IF, and preceding the optional ELSE, are executed on a true condition. Processors following the ELSE are executed on a false condition.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
OPRND1	A single real value to be compared under the specified relation to OPRND2 for conditional execution control.
RELATN	A character string specifying the type of relationship to use in determining the truth value of the condition. Valid relations are #, <, <=, =<, =, >=, =>, and >.
OPRND2	A single real value to be compared under the specified relation to OPRND1 for conditional execution control.

TABLE 4-III.- Continued

PROCESSOR ELSE

Processor name	Processor abstract prompt (maximum 256 characters)
ELSE	ELSE is a processor for use in sequence tables as part of IF groups. Its use is optional. When present it indicates a partitioning between the true condition and false condition processors of the IF group.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)

TABLE 4-III.- Concluded

PROCESSOR ENDIF

Processor name	Processor abstract prompt (maximum 256 characters)
ENDIF	ENDIF is a processor that must be paired with an IF. It marks the end of an IF group in a sequence table.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing. Additional material may be found in JSC IN 77-FM-18, vol. IV, rev. 2 dated April 1978, and in JSC IN 77-FM-18, Vol. IX (to be published).

INVARIANT ELEMENT EPHEMERIS PROCESSOR (INVAR)

1.0 PURPOSE

The INVAR processor generates a trajectory ephemeris in the form of sets of invariant elements for use by planning processors.

2.0 FUNCTIONAL DESCRIPTION

The base trajectory and maneuver information may be input either through a position/velocity phase table (INPHT) or by inputting the initial state (INVEC) and an array of maneuvers (INMAN). The processor then calls the AEG to obtain invariant elements at the start time, premaneuver states, postmaneuver states, final time, and at fixed-time intervals between these states that define nonthrusting areas.

If the trajectory and maneuver input is through a phase table, the processor uses the pre- and postmaneuver states in the phase tables as input to the AEG to obtain the invariant elements. If the maneuver input is made by an array of maneuvers, the base trajectory is updated to the time of the maneuver by the AEG. Then, an impulsive delta-V is applied to the state, and the AEG is called to initialize the postmaneuver state. These invariant elements in the form of a position/velocity phase table may optionally be output to a disk file specified by OUTPHT for use by planning processors.

3.0 ASSUMPTIONS AND LIMITATIONS

a. Only elliptical orbits may be propagated.

b. Maneuver simulations are done impulsively.

c. Maximum of five maneuvers.

d. When the initial vector and a maneuver array are input, there cannot be a maneuver between the vector time and the start time.

e. When the initial state and maneuver information is input through a phase table, the requested time frequency for outputting coasting vectors should be larger than the time between thrust-on and thrust-off events in the phase table.

f. The data element supplied for the parameter keyword, INPHT, must be an array sized (30,20).

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The interface table for the INVAR processor is contained in table 4-I. The IVEC flag determines whether the initial state, and the pre- and postburn maneuvers are input through a phase table; or the initial state and an array of maneuvers are input through the variables INVEC and INMAN. If the input is through a phase table, the STARV parameter identifies which phase table column entry is to be considered the starting vector, where the first column is identified as number "1".

b. Interface table data array definitions - The interface table data arrays are contained in table 4-II.

c. Interface table data file definitions - The format of the output data file is provided in table 4-III.

d. Processor solicited (prompted) input - None.

e. Processor displays and display parameter definition tables - None.

f. Processor message table - The processor error messages are contained in table 4-IV.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-V.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR INVAR

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
IVEC	AWA	2CH	I	1	1	"PT"	Initial vector and maneuver source flag: "IN" = input initial state and maneuver array
INVEC	AWA	Real	I	30	15		Input state vector; required when IVEC = "IN"
INMAN	AWA	Real	I	80	8		Input maneuvers required when IVEC = "IN"
INPHT	AWA	Real	I	1200	30,20		Input phase table required when IVEC = "PT"
STARV	AWA	Intg	I	1	1		Number of initial phase table vector; required when IVEC = "PT"
GETS	AWA	Real	I	6	3		Start time (hrs, min, sec) relative to base time
GETF	AWA	Real	I	6	3		Final time (hrs, min, sec) relative to base time
OUTPHT	Disk	Real	O				Name of the output phase table, disk file
GLOCON	AWA	Free	O	180	180	!!GLCN	Global constants array

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR INVAR

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
SESCN	AWA	Free	I	90	90	!SESCN	Session constants array
PROCON	AWA	Free	I	7	7		Processor constants array

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR INVAR

Array name	Index location	Default value	Definition
INVEC	(1 - 15)		Standard state vector format
INMAN	(1,1)		Time relative to base time for impulsive maneuver (hr)
	(2,1)		Time relative to base time for impulsive maneuver (min)
	(3,1)		Time relative to base time for impulsive maneuver (sec)
	(4,1)		Impulsive delta-V (LVLH X-component)
	(5,1)		Impulsive delta-V (LVLH Y-component)
	(6,1)		Impulsive delta-V (LVLH Z-component)
	(7,1)		Thrust for maneuver
	(8,1)		Specific impulse for maneuver
	(1,2)		Repeat of items 1 through 8 for second maneuver
	(1,3)		Repeat of items 1 through 8 for third maneuver
	(1,4)		Repeat of items 1 through 8 for fourth maneuver
	(1,5)		Repeat of items 1 through 8 for fifth maneuver
GETS	1		Start time relative to base time (hr)
	2		Start time relative to base time (min)
	3		Start time relative to base time (sec)
GETF	1		Final time relative to base time (hr)
	2		Final time relative to base time (min)
	3		Final time relative to base time (sec)
GLOCON	1 . . . 180	!!GLCN	Global constants array See table 7.2-III of JSC IN 78-FM-60, volume I for definition of contents
SESCN	1 . . . 90	!SESCN	Input/output session constants array. See table 7.2-II of JSC IN 78-FM-60, volume I for definition of contents.

TABLE 4-II.- Concluded

PROCESSOR INVAR

Array name	Index location	Default value	Definition
PROCON	1	0	Debug print flag = 0 No print = 1 Debug print
	2	0	Output destination flag = 0 Terminal = positive number LU of output device
	3	48	Maximum number of lines per page
	4	20	Cartridge reference number for data files
	5	43200.	Delta time between coasting states to be stored on the output file (seconds)
	7	1	Option flag for creating output disk file = 0 Do not create = 1 Create

TABLE 4-III.- INTERFACE TABLE DATA FILE DEFINITIONS

PROCESSOR INVAR

DRDE DATA FILE OUTPHT

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file
	4-6	Interface table variable creating file
	7-9	Processor last changing file
	10-12	Interface table variable last changing file
2-N	1-60	Position/velocity phase table column, see figure 7.3-13 in JSC IN 78-FM-60, vol. I for form and content. Each record contains one column of an invariant element state phase table.

TABLE 4-IV.- PROCESSOR MESSAGE TABLE

PROCESSOR INVAR

MSG no.	Message ID block	Message text block and explanation
1	*INVAR*	"NO DATA DOCUMENTATION ENTRY IN PHASE TABLE" Meaning: No data documentation entry was found in the phase table between the start vector and the last entry. Severity: Warning. Processor continues. Action required by user: Make sure phase table entries are valid.
2	*INVAR*	"FIRST PHASE TABLE ENTRY TIME IS GREATER THAN START TIME" Meaning: Time on the first vector in the phase table identified by STARV is greater than the start time. Processor will propagate the start vector backwards to the start time. Severity: Warning. Processor continues. Action required by user: Make sure phase table entries are valid.
3	*INVAR*	"MANEUVER BETWEEN START VECTOR TIME AND START TIME" Meaning: A maneuver, input by INMAN, is defined between the initial vector time and the start time. Severity: Processor terminates Action required by user: Change the start time to be before the maneuver time, or change the initial vector to be after the maneuver.
4	*INVAR*	"FILE CREAT ERROR=', IIII, "FILE NAME=", AAAAAA Meaning: Error occured when an attempt was made to create the output file. "IIII" is the RTE file manager error code. Severity: Processor terminates. Action required by user: Check the file name, etc., of the output file.

TABLE 4-V.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR INVAR

Processor name	Processor abstract prompt (maximum 256 characters)
INVAR	The INVAR processor generates a trajectory ephemeris in the form of sets of invariant elements for use by planning processors.
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
IVEC	Use definitions from processor interface table definitions (table 4-1).
INVEC	Use definitions from processor interface table definitions (table 4-I).
INMAN	Use definitions from processor interface table definitions (table 4-I).
INPHT	Use definitions from processor interface table definitions (table 4-I).
STARV	Use definitions from processor interface table definitions (table 4-I).
GETS	Use definitions from processor interface table definitions (table 4-I).
GETF	Use definitions from processor interface table definitions (table 4-I).
OUTPHT	Use definitions from processor interface table definitions (table 4-I).
GLOCON	Use definitions from processor interface table definitions (table 4-I).
SESCN	Use definitions from processor interface table definitions (table 4-I).
PROCON	See definition provided in table 4-Il.

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - MAIN PROGRAM INVAR

5.1.1 Purpose

INVAR is the main routine for the Invariant Elements Processor. INVAR uses the Analytic Ephemeris Generator (AEG) program to generate an Invariant Elements DRDE phase table.

5.1.2 Functional Description

All the inputs required for this routine are obtained through the interface table by calling the routine XPGET. The processor then determines if the input data is through a position/velocity phase table or through an input initial state vector and an array of maneuvers. In either case the input parameters are converted to internal units and placed in the proper routine variable locations. The AEG program is accessed through the UDATI routine to produce the sets of invariant elements. The output is produced through the OUTVC routine.

5.1.3 Assumptions and Limitations

a. Only elliptical orbits may be propagated.

b. Maneuver simulations are done impulsively.

c. A maximum of five maneuvers may be input.

d. When the initial vector and a maneuver array are input, there cannot be a maneuver between the vector time and the start time.

e. When the initial state and the maneuver information is input through a phase table, the requested time frequency for outputting the coasting vector should be larger than the time between thrust-on and thrustoff events in the phase table.

5.1.4 Method

None.

5.1.5 Routine Input/Output Variables

The input/output variables are presented in table 5.1-I.

5.1.6 Functional Logic Flow

The routine functional logic flow is presented in figure 5.1-1.

5.1.7 Diagnostics and Debug

None.

5.1.8 Special Comments

None.

5.1.9 References

None.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine INVAR

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
AHOUR	--	Real	I	sec	IT	GLOCON(50)	Number of seconds per hour
AMIN	--	Real	I	sec	IT	GLOCON(49)	Number of seconds per minute
CFD CFL CFM CFT CFV	-- -- -- -- --	Real Real Real Real Real	I I I I I	external external external external external	IT	SESCON(15) SESCON(20) SESCON(18) SESCON(16) SESCON(17)	External to internal units conversion factors for distance, length, mass, time, and velocity
DT	--	Real	I	sec	IT	PROCON(5)	Invariant elements output frequency
GETF	--	Real	I	hr,min, sec	IT	GETF	Final time relative to reference time
GETS	--	Real	I	hr,min, sec	IT	GETS	Start time relative to reference time
GMTR	--	Real	I	hr	IT	SESCON(6)	Reference time
ICR	--	Intg	I	--	IT	PROCON(4)	Cartridge reference for data files
INMAN	--	Real	I	external	IT	INMAN	Input maneuver data array
INPHT	--	Real	I	external	IT	INPHT	Input position/velocity phase table

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.1-I.- Concluded

Routine INVAR

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
INVEC	--	Real	I	external	IT	INVEC	Input position/velocity state vector
IPRNT	--	Intg	I	--	IT	PROCON(1)	Debug print flag
IVEC	--	2CH	I	--	IT	IVEC	Vector source flag
LUO	--	Intg	I	--	IT	PROCON(2)	Logical unit destination for printed output
STARV	--	Intg	I	--	IT	STARV	Number of initial phase table vectors
WTMASS	--	Real	I	ft/sec^2	IT	GLOCON(59)	Acceleration of gravity at sea level

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

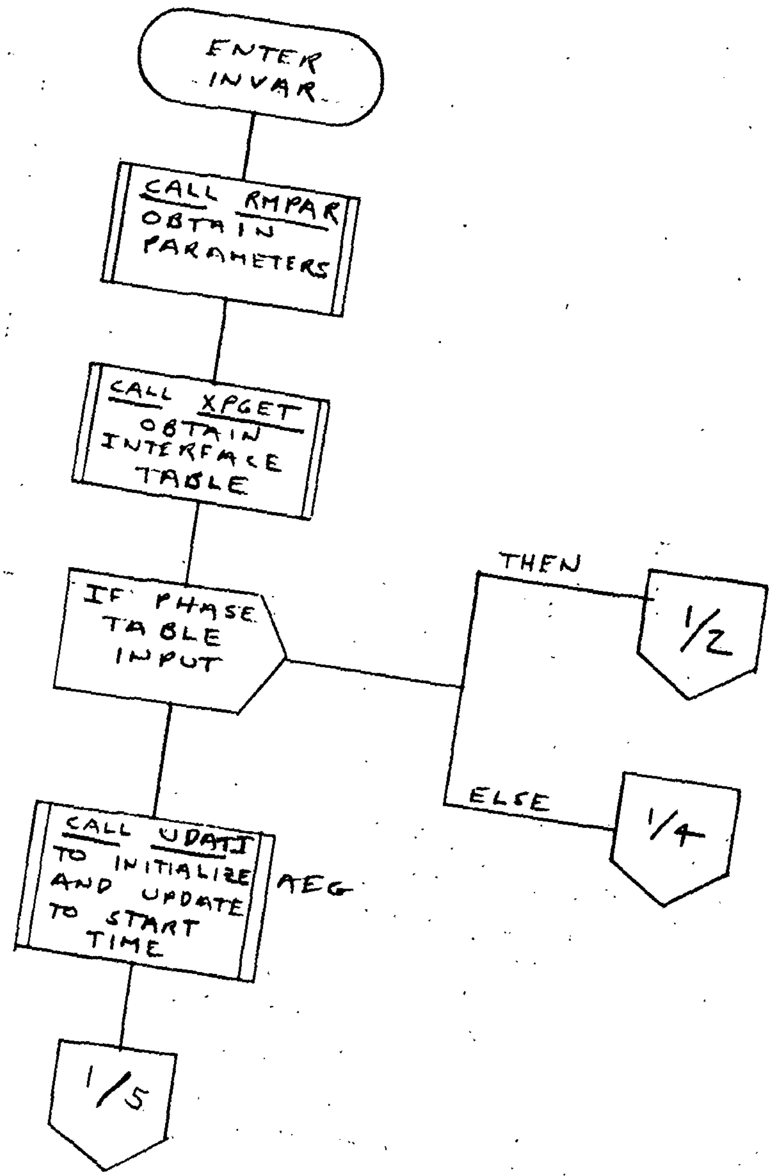


Figure 5.1-1.- INVAR functional logic flow.

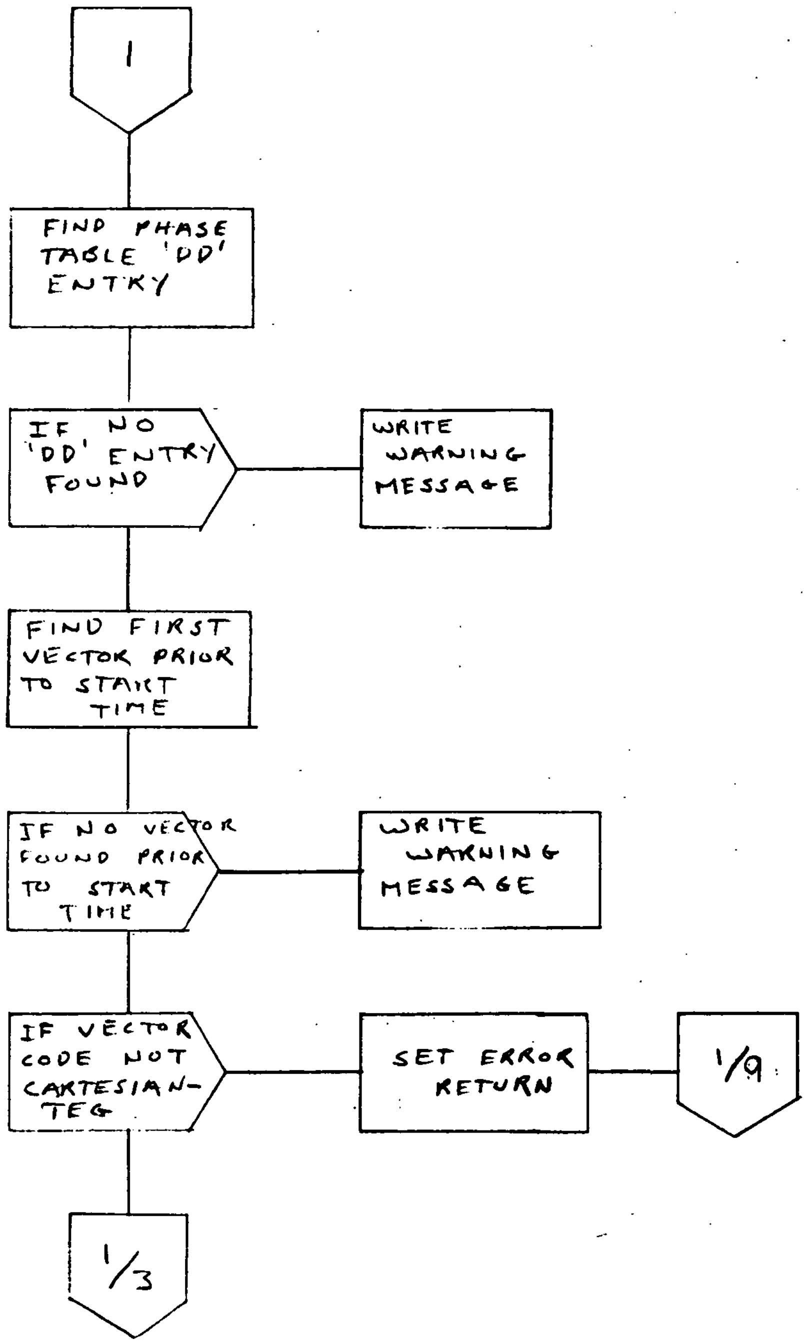


Figure 5.1-1.- Continued.

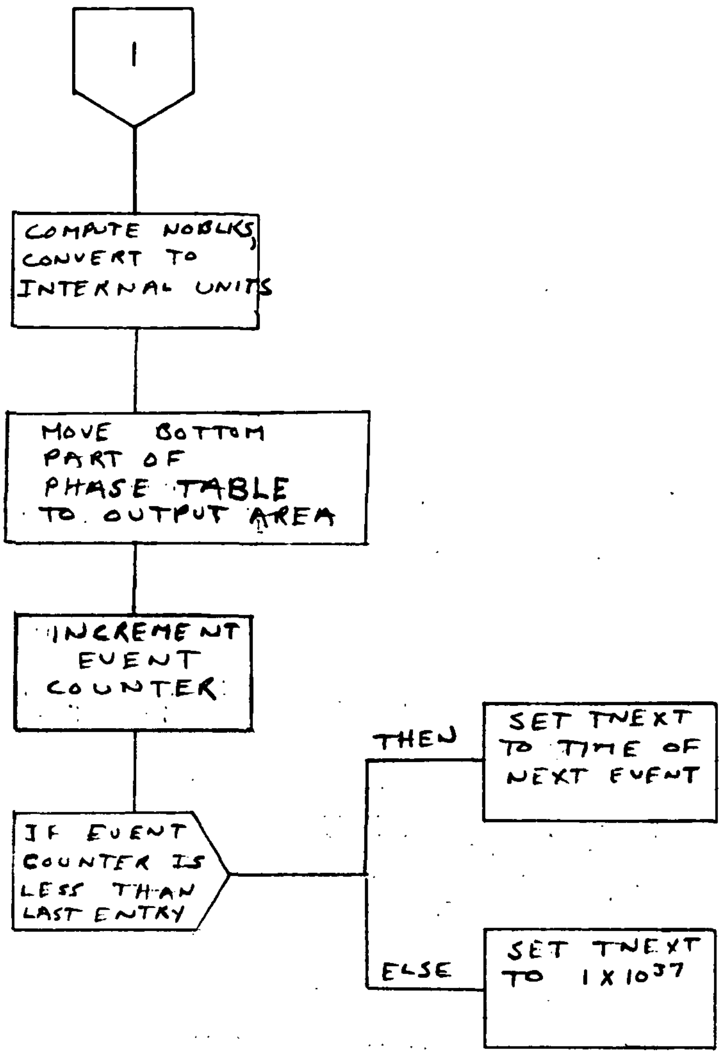
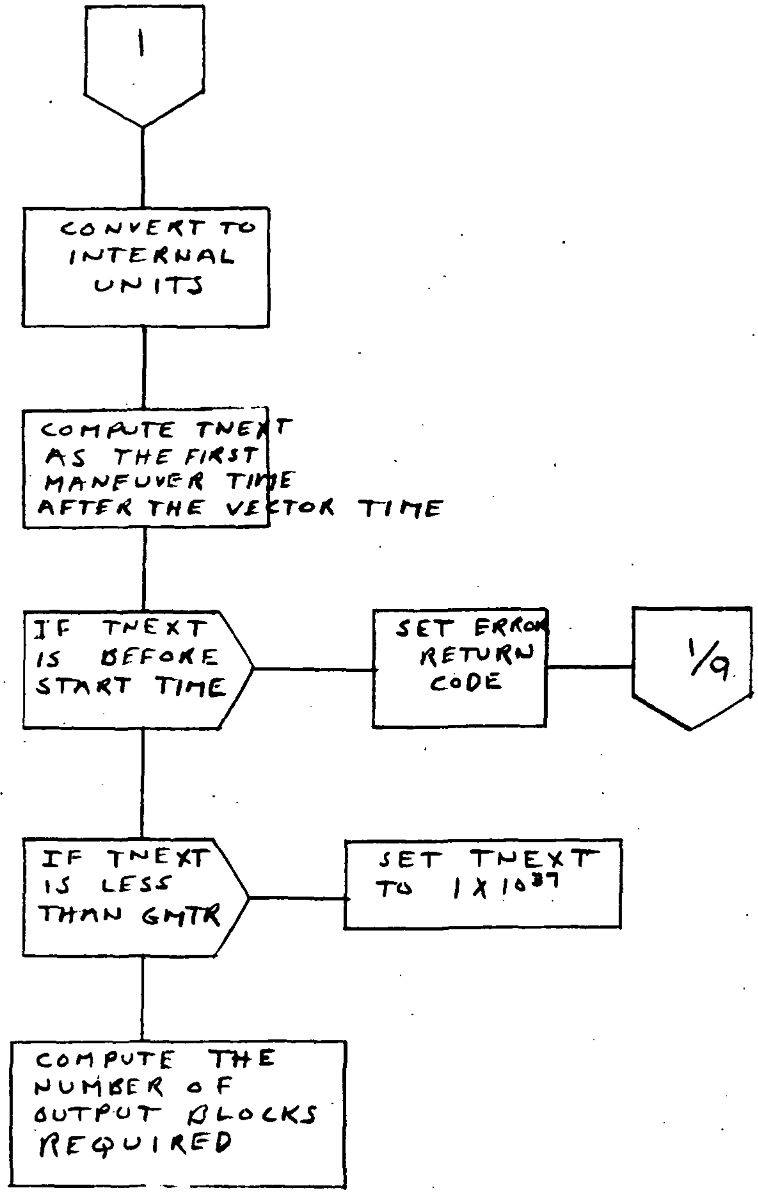
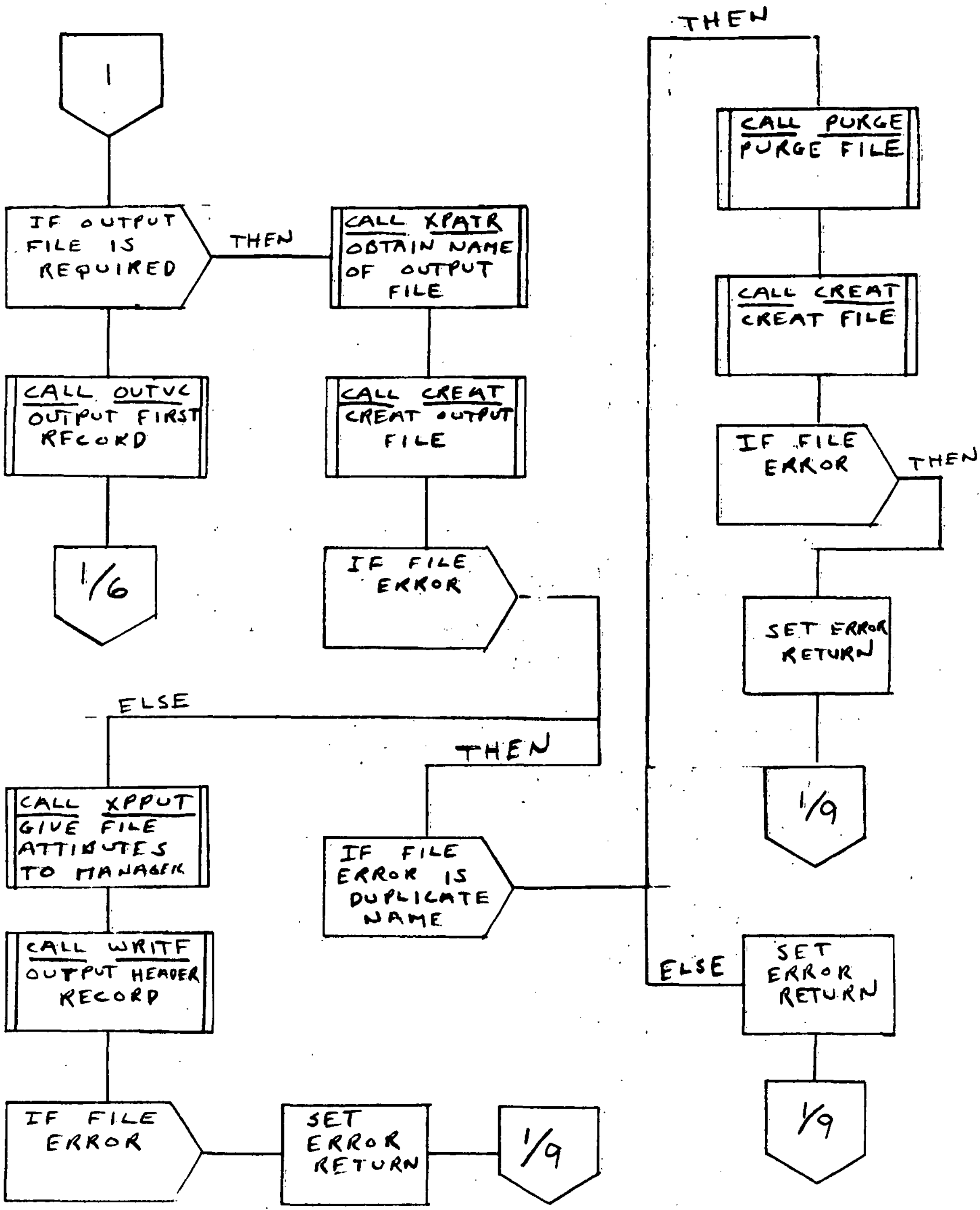


Figure 5.1-1.- Continued.



Page 4 of 9

Figure 5.1-1.- Continued.



Page 5 of 9

Figure 5.1-1.- Continued.

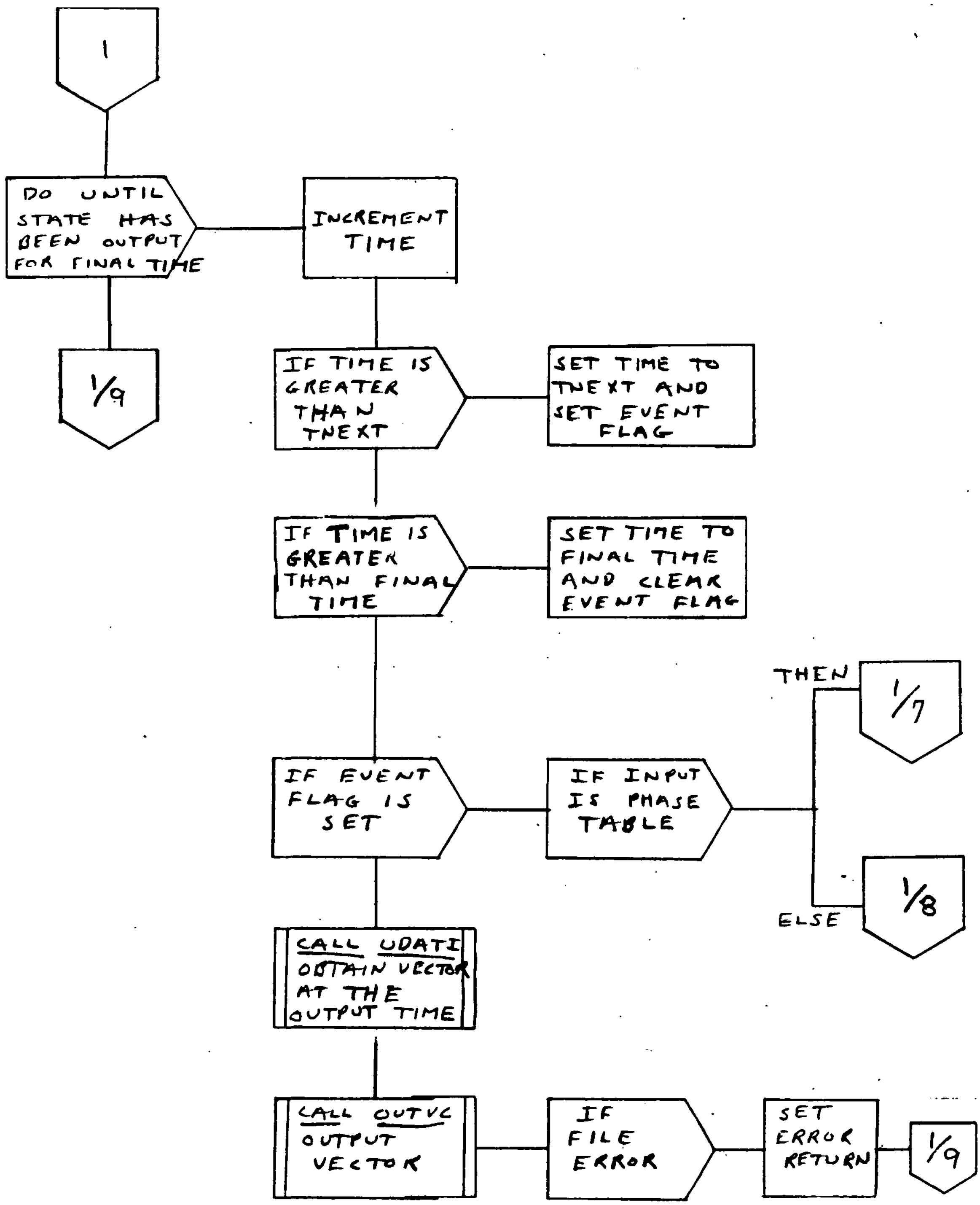
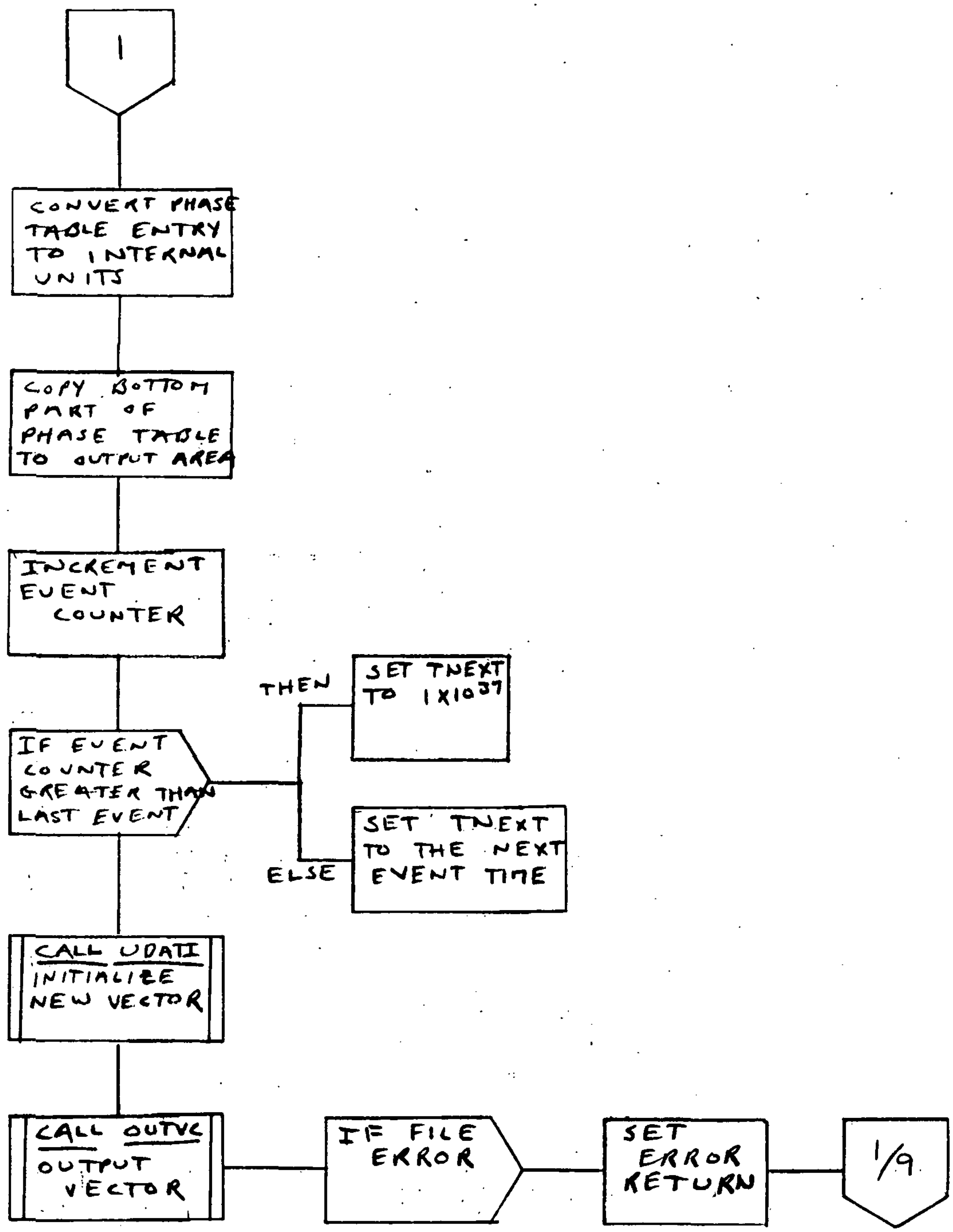
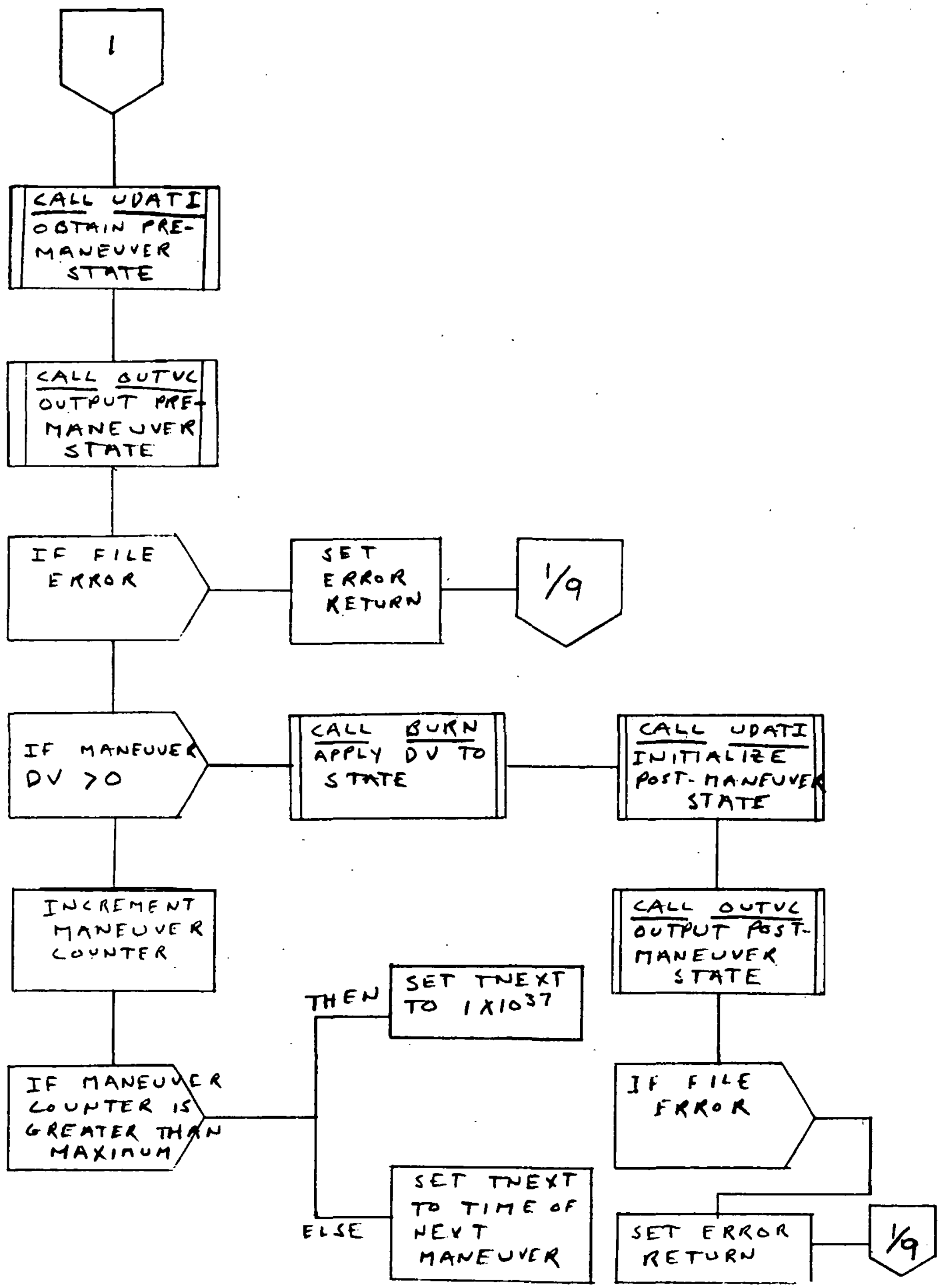


Figure 5.1-1.- Continued.



Page 7 of 9

Figure 5.1-1.- Continued.



Page 8 of 9

Figure 5.1-1.- Continued.

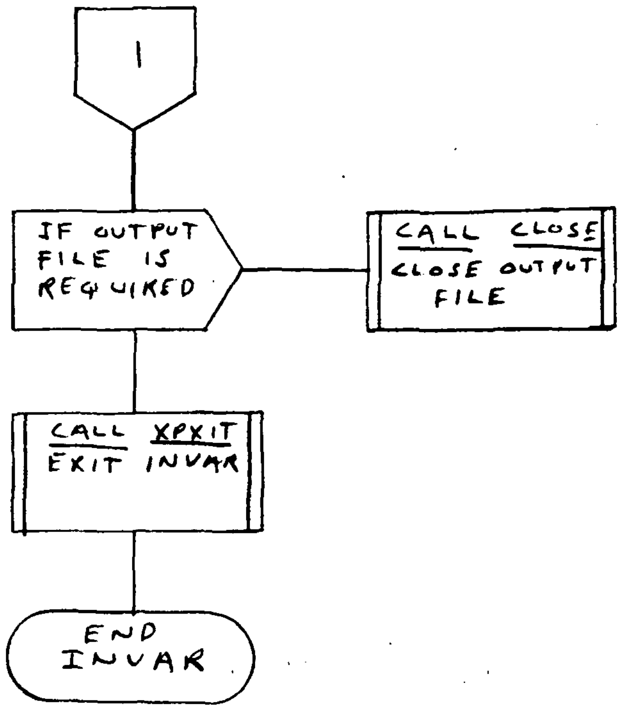


Figure 5.1-1.- Concluded.

5.2 ROUTINE NAME - BURN

5.2.1 Purpose

This routine applies a local-vertical, local-horizontal delta velocity to an input state vector and decrements the vehicle weight.

5.2.2 Functional Description

The routine computes the local-vertical, local-horizontal transformation matrix based on the input position and velocity vectors. The delta velocity vector is transformed to inertial coordinates and added to the velocity vector. The vehicle weight is also changed based on the magnitude of the velocity change, the specific impulse, and the acceleration at sea level.

5.2.3 Assumptions and Limitations

Only impulsive maneuvers are simulated.

5.2.4 Method

The local-vertical, local-horizontal transformation matrix is computed as

$$\hat{Y}_L = \text{unit}\ (\bar{V} * \bar{R})$$

$$\hat{Z}_L = -\text{unit}\ (\bar{R})$$

$$\hat{X}_L = \hat{Y}_L * \hat{Z}_L$$

where

$\bar{R}$ = position vector

$\bar{V}$ = velocity vector

The inertial velocity change is computed by

$$\Delta\overline{V}_I = \begin{bmatrix} \hat{X}_L \\ \hat{Y}_L \\ \hat{Z}_L \end{bmatrix}^T \Delta\overline{V}$$

where

$\Delta\overline{V}$ = input local vertical, local horizontal velocity change.

The new velocity is computed by

$$\overline{V} = \overline{V} + \Delta\overline{V}_I$$

The new weight is computed by

$$W_F = W_o * \exp(-VMAG/I_{SP} * g_o)$$

where

W_o = initial weight

VMAG = magnitude of velocity change

I_{SP} = specific impulse

g_o = acceleration at sea level

The following correlates the above symbols with the routine variables

Math symbol	Internal code symbol
g_o	WTMASS
I_{SP}	RSP
W_o	WHT

Math symbol	Internal code symbol
W_f	WHT
$\bar{R}$	R
$\bar{V}$	V
$\hat{X}_L$	XL
$\hat{Y}_L$	YL
$\hat{Z}_L$	ZL
$\Delta\bar{V}$	DV
$\Delta\bar{V}_I$	DVI

5.2.5 Routine Input/Output Variables

The FORTRAN calling sequence for this routine is

```
CALL BURN(R,V,DV,WHT,THRUST,ISP,GO)
```

The input/output variables are presented in table 5.2-I.

5.2.6 Functional Logic Flow

The functional logic flow is as follows.

```
1   BEGIN BURN
2      CONSTRUCT LOCAL-VERTICAL, LOCAL-HORIZONTAL
          COORDINATE SYSTEM
2      CONVERT DELTA VELOCITY TO INERTIAL COORDINATE
          SYSTEM
2      ADD DELTA VELOCITY TO VELOCITY VECTOR
2      DECREMENT VEHICLE WEIGHT
1   END BURN
```

5.2.7 Diagnostics and Debug

None.

5.2.8 Special Comments

None.

5.2.9 References

None.

TABLE 5.2-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine BURN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
R	$\bar{R}$	Real	I/O	feet	A		Position vector
V	$\bar{V}$	Real	I/O	ft/sec	A		Velocity vector
DV	$\bar{V}$	Real	I	ft/sec	A		Delta velocity vector
WHT	w_o/w_f	Real	I/O	lb	A		Vehicle weight
RSP	I_{sp}	Real	I	sec	A		Specific impulse
WTMASS	g_o	Real	I	ft/sec^2	A		Weight to mass conversion factor

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.3 ROUTINE NAME - OUTVC

5.3.1 Purpose

This routine is the output routine for the INVAR processor. Based on optional flag settings the routine may output to the CRT and/or to a DRDE disk file.

5.3.2 Functional Description

If the print flag is set the output will be made to the desired logical unit using the print state vector utility routine SPSV. If the output file flag is set, the output is made to a DRDE disk file using FMP routines.

5.3.3 Assumptions and Limitations

None.

5.3.4 Method

None.

5.3.5 Routine Input/Output Variables

The FORTRAN calling sequence for this routine is

```
CALL OUTVC(TU,CD,AREA,WHT,ISET,IERR)
```

The input/output variables are presented in table 5.3-I. If the ISET variable is less than zero, then a data documentation entry is output.

5.3.6 Functional Logic Flow

The functional logic flow is presented in figure 5.3-1.

5.3.7 Diagnostics and Debug

None.

5.3.8 Special Comments

None.

5.3.9 References

None.

TABLE 5.3-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine OUTVC

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TU		Real	I	sec	A		Output time
CD		Real	I	--	A		Coefficient of drag
AREA		Real	I	ft^2	A		Cross-sectional area
WHT		Real	I	lb	A		Vehicle weight
ISET		Intg	I	--	A		Flag used to control output to disk. Negative indicates last output.
LU		Intg	I	--	C		Input logical unit
ELEM		Real	I		C		Invariant elements array
IPRNT		Intg	I		C		Print flag
LUO		Intg	I		C		Output logical unit
SESCON		Free	I		C		Session constants array
OUTFLG		Intg	I		C		Output file flag
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.3-I.- Concluded

Routine QUTVC

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IBUFO		Intg	I/O		C		Output file buffer
IRECO		Intg	I/O		F		Output file record

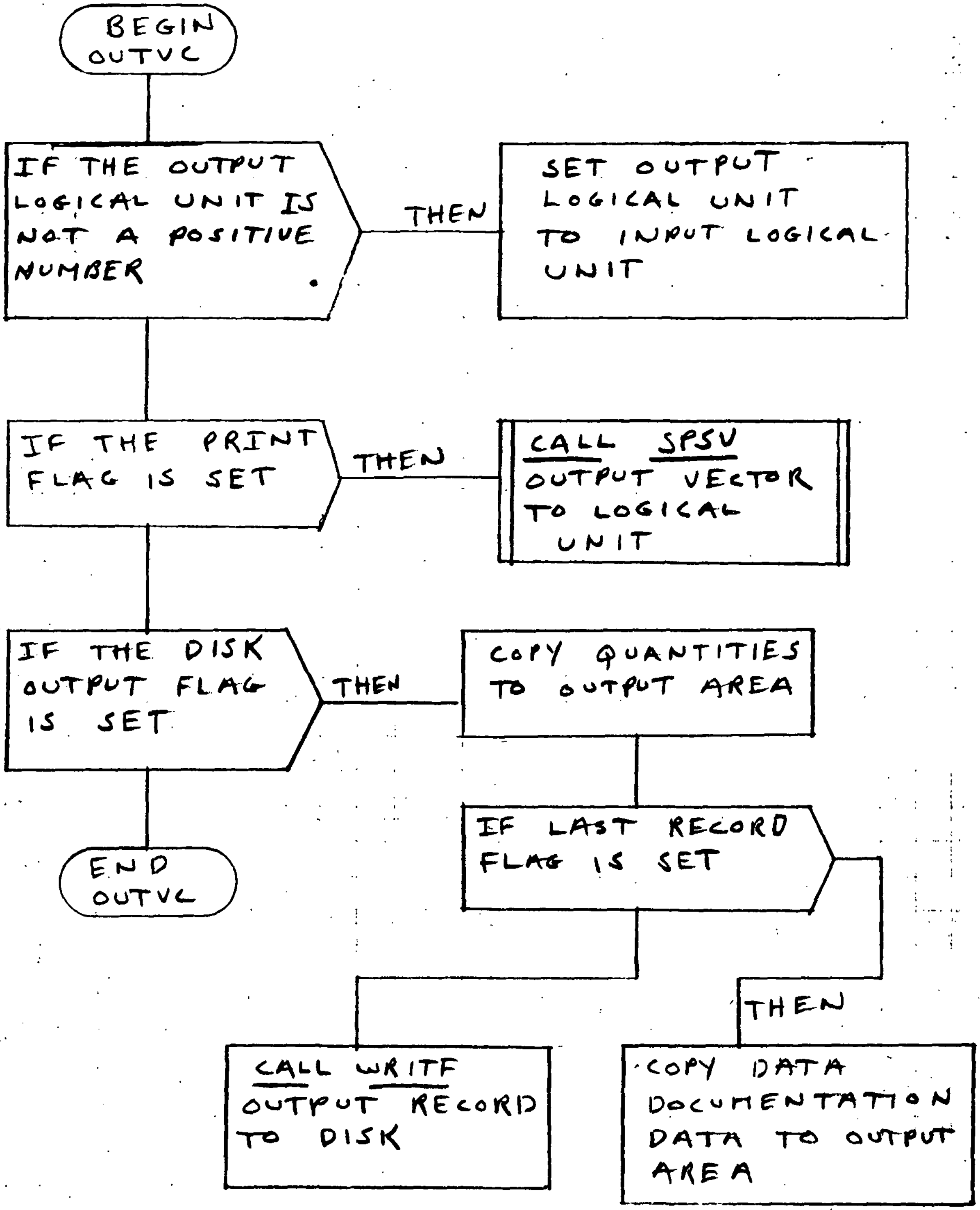
NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.3-1.- OUTVC functional logic flow.

5.4 ROUTINE NAME - UDATI

5.4.1 Purpose

This routine provides the interface to the Analytic Ephemeris Generator (AEG) for the purpose of obtaining invariant elements as output.

5.4.2 Functional Description

The routine obtains the vector quantities through the calling arguments and the constants quantities through common and places them in the proper location in the AEG data array. The STAEG routine is then used as the AEG interface to pass the AEG data array as input and to retrieve the data array as output. The position and velocity vectors, and the invariant elements are output at the update time.

5.4.3 Assumptions and Limitations

The AEG propagates only elliptical orbits.

5.4.4 Method

None.

5.4.5 Routine Input/Output Variables

The FORTRAN calling sequence is

```
      CALL UDATI(R,V,TV,TU,CD,AREA,WHT,KE,ELEM)
```

These parameters are defined in table 5.4-I.

5.4.6 Functional Logic Flow

The functional logic flow is presented as follows.

```
 1  BEGIN UDATI
 2      IF INITIALIZATION IS REQUIRED THEN
 3          INITIALIZE THE AEG DATA ARRAY
 2      CALL ROUTINE STAEG
```

2 OBTAIN POSITION AND VELOCITY VECTORS
FROM THE AEG DATA ARRAY

2 OBTAIN THE INVARIANT ELEMENTS
FROM THE AEG DATA ARRAY

1 END UDATI

5.4.7 Diagnostics and Debug

None.

5.4.8 Special Comments

None.

5.4.9 References

None.

TABLE 5.4-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine UDATI

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
R		Real	I/O	feet	A		Position vector
V		Real	I/O	ft/sec	A		Velocity vector
TV		Real	I/O	sec	A		Vector time
TU		Real	I	sec	A		Update time
CD		Real	I	--	A		Drag coefficient
AREA		Real	I	ft^2	A		Cross-sectional area
WHT		Real	I	lb	A		Vehicle weight
KE		Intg	I	--	A		Initialization flag
ELEM		Real	I/O		A		Invariant elements array
LU		Intg	I		C		Logical unit
CONST		Real	I		C		AEG constants array
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

CASKU AND QUIKU OUTPUT DISPLAY PROCESSOR (LKOUT)

1.0 PURPOSE

The LKOUT application processor provides the means to display the results of the nonpropulsive consumables analysis performed by either the CASKU or QUIKU applications processors. The processor produces both tabular listings or plots of nonpropulsive consumables versus time.

2.0 FUNCTIONAL DESCRIPTION

The LKOUT display processor has two phases of operation during its execution. These phases are:

a. Table presentation

b. Plot presentation

The processor will first compute and display to the user, on the terminal, a summary of the results produced by the CASKU or QUIKU processors, then will enter the next sequence of execution.

The processor will issue a prompt to the user to determine which of the two phases of execution is desired. A response of a question mark (enclosed in quotes) will provide the user with a list of the options available for display.

a. Table presentation - After the user has selected the desired table option in response to the solicited prompt, the processor will branch to the logic necessary to produce and display the requested table.

   During this portion of the execution the user, in response to the solicited prompt, may request that only a specified time interval of the result be displayed. When this option is selected the user will be prompted by the processor for the necessary information to determine the time interval. All tables from this point will only display the data for the specified time interval.

b. Plot presentation - When the plot option is selected, in response to the processor solicited prompt, a branch is made to the logic necessary to provide plot displays. The processor will prompt the user for the availability of the plotting terminal if it is not the one being used.

   The processor will prompt the user for the desired plot option. As with the table presentation phase, a question mark (enclosed in quotes) will provide the user with a list of the plot options.

   After selecting the desired plot option, the user will be prompted for the scale values to be used on the plots. The processor upon receiving the scale information will branch to the logic necessary to produce the desired plot.

Upon the completion of the desired plots, the user selects the table option in response to the solicited prompt and then exits the processor or reenters the table presentation phase.

3.0 ASSUMPTIONS AND LIMITATIONS

None.

4.0 PROCESSOR INPUT/OUTPUT

The user/processor communication is both through the interface table and solicited prompts. The primary communications during the execution is through the solicited prompts issued by the processor.

a. Processor interface table - The user, through the variable TIMREF, specifies the time reference base to be used for the displays. For ground elapsed time (GET) the user will enter GE and for Greenwich mean time (GMT) GM will be used.

   Initial quantities of oxygen, hydrogen, nitrogen, and ammonia (in that order), are specified by the user through the variable SQTY.

   Specific constants required by the processor will be maintained through the default values stored in the processor constants array PROCON. Specifically, these constants are the cartridge reference number for the DRDE file, the logical unit number of the plotting terminal, and the device print flag. The user may specify the line printer for the tables by entering a 6 in PROCON. The name of the DRDE file to be used is specified by entering the four character name in the variable DATAFL. The values for LHOUR, LYEAR, and LDAY are defaulted to the corresponding values in the !SESCON array for use with this processor; therefore, the BASTM processor must be executed to initialize the !SESCON array prior to execution of this processor.

   The definitions of the processor interface table variables are provided in table 4-I.

b. Interface table data array definitions - The definitions of the LKOUT processor constants and data arrays are provided in table 4-II.

c. Interface table data file definitions - The definition of the input DRDE file used by the LKOUT processor are contained in table 4-III.

d. Processor solicited (prompted) inputs - The processor solicited prompts are provided in table 4-IV.

e. Processor displays and display definition table - The processor displays and the display parameter definitions are provided in table 4-V(a) through table 4-V(xx).

f. Processor message table - The LKOUT processor error messages are provided in table 4-VI.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR LKOUT

Parameter keyword name	Class	Type	Use	Size	Array dimen-sion (I,J)	Values stored in default interface table	Definition
TIMREF	AWA	2CH	I	1	--		Time reference base "GE" ground elapsed time "GM" Greenwich mean time
SQTY	AWA	Real	I	8	4		Initial quantities of O2, H2, N2, NH3
DATAFL	Disk	Intg	I	--	--		Name of data file to be used for input
PROCON	AWA	Intg	I	3	3	20,31,0	Processor constants (1) DRDE logical unit (2) Plot terminal logical unit (3) Print device flag
LDAY	AWA	Intg	I	1	1	!SESCN(6)	Day of launch
LHOUR	AWA	Real	I	1	1	!SESCN(11)	Hour of launch (decimal)
LYEAR	AWA	Intg	I	1	1	!SESCN(3)	Year of launch

NOTES	CLASS	TYPE			USE
		Free	2CH	72CH	I = Input
	AWA	Intg	6CH	Mix	O = Output
	Disk	Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR LKOUT

Array name	Index location	Default value	Definition
SQTY	(1)	--	Initial quantity of oxygen
	(2)	--	Initial quantity of hydrogen
	(3)	--	Initial quantity of nitrogen
	(4)	--	Initial quantity of ammonia
PROCON	(1)	20	DRDE logical unit number
	(2)	31	Plotting terminal logical unit number
	(3)	0	Print device flag - (6 for line printer)

TABLE 4-III.- INTERFACE TABLE DATA FILE DEFINITIONS

PROCESSOR LKOUT

DRDE DATA FILE /XXXXC

Record number	Integer word allocations	Content and definition
1	1-3	"CASKU"/"QUIKU" - FDS processor name that created the file
	4-6	DATAFL - Interface table variable name through which the file was created
	7-9	"CASKU"/"QUIKU" - FDS processor name that last updated the file
	10-12	DATAPL - Interface table variable name through which the file was changed
2	1-5	IT (1-5) - Year, Julian calendar day, hour, minute, and second that the data file was generated
	6-8	TFILNM - Name of the users time-line file
	9-11	ORBACT - Name of the Orbiter activity data file used
	12-14	PAYACT - Name of the payload activity data file used
	15-17	SYSFIL - Name of the system characteristics data file used
	18-20	PROGNM - Name of the processor that created the file
3	1-2	TIME (1-2) - The time (in decimal) that a system constraint limit was exceeded
	3-5	KIND (1-3) - A variable containing the four-character mnemonic for the system and the integer value designating the system branch (node) that caused the constraint violation
	6-9	VAL (1-2) - The value (in decimal) that caused the violation and the system limit (in decimal) that was violated

TABLE 4-III.- Concluded

PROCESSOR LKOUT

DRDE DATA FILE /XXXXC

Record number	Integer word allocations	Content and definition
4	1-2	TIME (1-2) - The time (in decimal) that an activity occurred
	3-14	QTY (1-6) - Quantity values at time for O_2, H_2, N_2, NH_3, waste H_2O, and potable H_2O
	15-22	PWR (1-4) - Electrical power values at time for the connected load, busload, source power, and subsystems load
	23-34	SRCVLT (1-6) - Source voltages at time for fuel cells and batteries
	35-46	SRCCUR (1-6) - Source currents at time for fuel cells and batteries
	47-58	HEAT (1-6) - Heat values at time for the total heat load, heat rejected by the radiator, ammonia boiler, ascent evaporator, topping evaporator, and total heat rejected
	59-66	PRS (1-4) - Partial pressures of N_2 and O_2 at time for cabins 1 and 2
	67-76	TMP (1-5) - Temperatures at time for the cabins, heat exchanger, hydraulics inlet, radiator inlet, and cooling outlet
	77-78	PEAK - Statistical peak power at time
	79-80	PMIN - Statistical minimum power at time
	81-82	THMCAP - Total cooling capacity of the system at time
	83-84	TPMRGN - Cooling capacity remaining at time

TABLE 4-IV.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR LKOUT

Prompt	Meaning	Valid responses
TABLE?	Enter two character code of the desired table.	CO - Print the table containing the system constraints information. QT - Print the table containing the nonpropulsive consumables information for O_2, H_2, N_2NH_3, waste H_2O and potable H_2O. PW - Print the table containing the electrical power summary information. SR - Print the table containing the source power information. HT - Print the table containing the heat summary information. PT - Print the table containing the pressure and temperature information. PL - Diplay the list of available plots and input codes. ? - List the table showing the optional tables and input codes. EX or ƀƀ - Transfer to an exit routine. TI - Enter time interval of table when prompted.
IS TEKTRONIX TERMINAL FREE?	Enter yes or no.	YES - The terminal can be used for plots. NO or ƀƀ - Return to main program.
PLOT?	Enter two-character code for the desired plot.	PP - Plot power profile curve(s). PA - Plot power available curve(s). KW - Plot kilowatt hours consumed curve.

TABLE 4-IV.- Continued

PROCESSOR LKOUT

Prompt	Meaning	Valid responses
PLOT?		O2 - Plot oxygen required (remaining) curve. H2 - Plot hydrogen required (remaining) curve. N2 - Plot nitrogen required (remaining) curve. NH - Plot ammonia required (remaining) curve. WW - Plot waste water produced curve. WA - Plot potable water remaining curve. TC - Plot cabin temperature curve. TA - Plot avionics outlet temperature curve. TH - Plot hydraulics inlet temperature curve. TR - Plot radiator inlet temperature curve. TO - Plot cooling outlet temperature curve. TB or ƀƀ - Return to the main program. TI - Time interval of plot.
SYSTEM CONSTRAINT LIMIT (KILOWATTS)?	Enter system limit.	XXX or XXX.X Integer or real numbers.
AVAILABLE HYDROGEN (LB)?	Enter quantity of H_2.	Integer or real numbers. XXX or XXX.X
XMIN, XMAX, YMIN, YMAX	Enter the appropriate limits to appear on the plot.	XXX or XXX.X Integer or real numbers. ƀ - defaults to processor stored limits.

TABLE 4-IV.- Concluded

PROCESSOR LKOUT

Prompt	Meaning	Valid responses
START +HHH:MM:SS : : STOP +HHH:MM:SS : :	Enter times requested.	XXX - Integer values under the appropriate letters.

TABLE 4-V.- PROCESSOR DISPLAY AND DISPLAY PARAMETER DEFINITIONS

(a) Output data generated display

PROCESSOR LKOUT

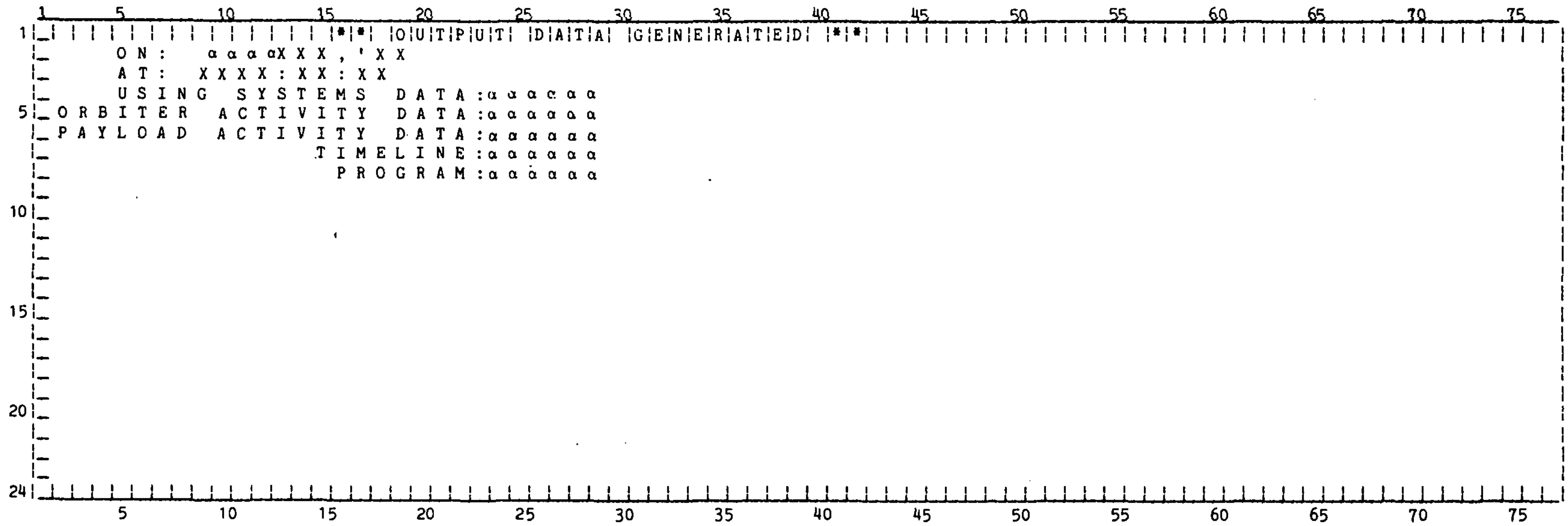


TABLE 4-V.- Continued

(b) Display parameter definition table for the output data generated display

PROCESSOR LKOUT

OUTPUT DATA GENERATED

Display parameter label	Parameter definition
ON:	Date of data generation: MMM, DD, 'YY where MMM - Mnemonic for month DD - Day of month 'YY - Year
AT:	Time of data generation: HHH:MM:SS where HHH - Hours MM - Minutes SS - Seconds
USING SYSTEM DATA:	Name of the systems characteristics data file used
ORBITER ACTIVITY FILE:	Name of the Orbiter activity data file used
PAYLOAD ACTIVITY DATA:	Name of the payload activity data file used
TIMELINE:	Name of the time line used
PROGRAM:	Name of the processor that created the data

TABLE 4-V.- Continued

(c) Energy summary and consumables status display

PROCESSOR <u>LKOUT</u>

```
              ENERGY SUMMARY AND CONSUMABLES STATUS
 ELECTRICAL ENERGY (KWH)          CONSUMABLES REQUIRED (LBS)
    CONNECTED LOAD:XXXXX.X                    OXYGEN:XXXX.X
          BUS LOAD:XXXXX.X                  HYDROGEN:XXXX.X
        DIST. LOSS:XXXXX.X                  NITROGEN:XXXX.X
     SOURCE ENERGY:XXXXX.X                   AMMONIA:XXXX.X
                                         WASTE WATER:XXXX.X  (GENERATED)
        SUB ENERGY:XXXXX.X             POTABLE WATER:XXXX.X  (REMAINING)
```

TABLE 4-V.- Continued

(d) Display parameter definition table for the energy summary and consumables status display

PROCESSOR LKOUT

ENERGY SUMMARY AND CONSUMABLES STATUS	
Display parameter label	Parameter definition
CONNECTED LOAD:	Total electrical energy load, kWh
BUS LOAD:	Load on the bus, kWh
DIST. LOSS:	Amount of line loss, kWh
SOURCE ENERGY:	Amount of energy available from fuel cells, kWh
SUB ENERGY:	Amount of energy required by payload systems, kWh
OXYGEN:	Total amount of oxygen required, lb
HYDROGEN:	Total amount of hydrogen required, lb
NITROGREN:	Total amount of nitrogen required, lb
AMMONIA:	Total amount of ammonia required, lb
WASTE WATER:	Total amount of waste water produced by the crew, lb
POTABLE WATER:	Total amount of potable water remaining, lb

TABLE 4-V.- Continued

(e) Equivalent fuel cell lifetime usage display

PROCESSOR LKOUT

```
**EQUIVALENT FUEL CELL LIFETIME USAGE**
          (MISSION TIME±XXX.X HRS)
      FUEL CELL±X:±XXX.X HRS
      FUEL CELL±X:±XXX.X HRS
      FUEL CELL±X:±XXX.X HRS
```

TABLE 4-V.- Continued

(f) Display parameter definition table for the equivalent fuel cell lifetime usage display

PROCESSOR LKOUT

EQUIVALENT FUEL CELL LIFETIME USAGE	
Display parameter label	Parameter definition
MISSION TIME	Elapsed time of the mission profile time line, hr
FUEL CELL:	Equivalent lifetime of each fuel cell, hr

TABLE 4-V.- Continued

(g) Constraints display

PROCESSOR <u>LKOUT</u>

```
                CONSTRAINTS
      TIME      TYPE    NO      VAL        VGL.  LMT
 ±XXX:XX:XX     αααα ±XXX ±XXXXXXX.XX ±XXXXXX.XX

     OR

XXX:XX:XX:XX
```

±XXX:XX:XX Relative time format
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(h) Display parameter definition table for the contraints display

PROCESSOR LKOUT

CONSTRAINTS	
Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
TYPE	Four-character name for the constraint
NO	Number of the system
VAL	Calculated value that caused the constraint
VAL. LMT	Limit that was violated

TABLE 4-V.- Continued

(i) Quantities display

PROCESSOR LKOUT

```
         QUANTITIES
      TIME          O2         H2         N2         NH3       WSTHRO      H2O
  ±XXX:XX:XX  ±XXXXX.XX±XXXXX.XX±XXXXX.XX±XXXXX.XX±XXXXX.XX±XXXXX.XX

      OR

 XXX:XX:XX:XX
```

±XXX:XX:XX Relative time format
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(j) Display parameter definition table for the quantities display

PROCESSOR LKOUT

QUANTITIES	
Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Denotes a time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
O2	Quantity of oxygen used or remaining, lb
H2	Quantity of hydrogen used or remaining, lb
N2	Quantity of nitrogen used or remaining, lb
NH3	Quantity of ammonia used or remaining, lb
WSTH2O	Quantity of waste water produced, lb
H2O	Quantity of potable water remaining, lb

TABLE 4-V.- Continued

(k) Power summary display

PROCESSOR <u>LKOUT</u>

```
                  POWER SUMMARY
     TIME           SOURCE      BUS       28V        SUB        PEAK       MINIMUM     THERMAL
                                                                                        MARGIN
  ±XXX:XX:XX  ±XXXXXX.X±XXXXXX.X±XXXXXX.X±XXXXXX.X±XXXXXX.X±XXXXXX.X±XXXXXX.X

        OR

 XXX:XX:XX:XX
```

±XXX:XX:XX *Relative time format*
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(1) Display parameter definition table for the power summary display

PROCESSOR LKOUT

POWER SUMMARY

Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Denotes a time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
SOURCE	Source system power at a given point in time, W
BUS	Bus system power at a given point in time, W
28V	28V system power at a given point in time, W
SUB	Subsystem power at a given point in time, W
PEAK	Statistical peak load at a given point in time, W
MINIMUM	Statistical minimum load at a given point in time, W
THERMAL MARGIN	Cooling capacity remaining at any given point in time, W

TABLE 4-V.- Continued

(m) Source power data

PROCESSOR LKOUT

```
 1   SOURCE POWER DATA
 2     TIME
 3                POWER  SOURCEXX CURRENT  VOLTS   POWER  SOURCEXX CURRENT  VOLTS   POWER  SOURCEXX CURRENT  VOLT
 4  ±XXX:XX:XX  ±XXXXXX.X±XXXXX.X±XXX.X±XXXXXX.X±XXXXX.X±XXX.X±XXXXXX.X±XXXXX.X±XXX.
 5
 6     OR
 7
 8 XXX:XX:XX:XX
```

±XXX:XX:XX Relative time format
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(n) Display parameter definition table for the power data display

PROCESSOR LKOUT

SOURCE POWER DATA	
Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Denotes a time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
SOURCE	One of the three main power sources
POWER	The value, in W, of each of the power sources at a given point in time
CURRENT	The value, in amps, of each of the power sources at a given point in time
VOLTS	The value, in volts, of each of the power sources at a given point in time

TABLE 4-V.- Continued

(o) Heat summary display

PROCESSOR LKOUT

```
          HEAT SUMMARY
       TIME          LOAD      RADIATOR    AMMONIA     ASCENT     TOPPING      COOL       CAPACITY
  ±XXX:XX:XX    ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X ±XXXXXX.X

       OR

 XXX:XX:XX:XX
```

±XXX:XX:XX Relative time format
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(p) Display parameter definition table for the heat summary display

PROCESSOR LKOUT

HEAT SUMMARY	
Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Denotes a time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
LOAD	Total heat load on the system at a given point in time, Btu/hr
RADIATOR	Radiator inlet heat load at a given point in time
AMMONIA	Amount of heat rejected at a given point in time
ASCENT	Amount of heat rejected at a given point in time
TOPPING	Amount of heat rejected at a given point in time
COOL	Total heat rejected at a given point in time
CAPACITY	Total cooling capacity of the system

TABLE 4-V.- Continued

(q) Cabin pressure and temperature display

PROCESSOR LKOUT

```
                      CABIN PRESSURE AND TEMPERATURES
      TIME          TOT       N2       O2        CAB        XGR        HYD        RAD        OUT
 ±XXX:XX:XX     ±XXX.X±XXX.X±XX.X±XXXX.X±XXXX.X±XXXX.X±XXXX.X±XXXX.X

      OR

XXX:XX:XX:XX
```

±XXX:XX:XX Relative time format
XXX:XX:XX:XX Absolute time format

TABLE 4-V.- Continued

(r) Display parameter definition table for the cabin pressure and temperatures display

PROCESSOR LKOUT

CABIN PRESSURE AND TEMPERATURES	
Display parameter label	Parameter definition
TIME	A specific time in either one of two formats: (1) Relative time; ±HHH:MM:SS where ± - Denotes a time prior to or after lift-off; the + sign will be suppressed HHH - Hours MM - Minutes SS - Seconds (2) Absolute time; DDD:HH:MM:SS where DDD - Julian day HH - Hours MM - Minutes SS - Seconds
TOT	Total cabin pressure at a given point in time, psi
N2	Partial pressure of N_2 at a given point in time, psi
O2	Partial pressure of O_2 at a given point in time, psi
CAB	Cabin temperature at a given point in time, $^{\circ}F$
XGR	Heat exchanger temperature at a given point in time, $^{\circ}F$
HYD	Hydraulics inlet temperature at a given point in time, $^{\circ}F$
RAD	Radiator inlet temperature at a given point in time, $^{\circ}F$
OUT	Cooling outlet temperature at a given point in time, $^{\circ}F$

TABLE 4-V.- Continued

(s) Plot options and input codes display

PROCESSOR LKOUT

```
-PLOT OPTIONS AND INPUT CODES-
     POWER PROFILE:  PP
 POWER AVAILABLE:  PA
    KWH CONSUMED:  KW
          OXYGEN:  O2
        HYDROGEN:  H2
        NITROGEN:  N2
         AMMONIA:  NH
     WASTE WATER:  WW
   POTABLE WATER:  WA
CAB. TEMPERATURE:  TC
 AVIONICS OUTLET:  TA
HYDRAULICS INLET:  TH
  RADIATOR INLET:  TR
  COOLING OUTLET:  TO
          TABLES:  TB  (OR BLANK)
```

TABLE 4-V.- Continued

(t) Display parameter definition table for the plot options and input codes display

PROCESSOR LKOUT

PLOT OPTIONS AND INPUT CODES	
Display parameter label	Parameter definition
--	Provides the user with a list of the possible plots that may be produced by the processor and the two-character codes that are used in response to the processor prompt - PLOT?

TABLE 4-V.- Continued

(u) Optional tables and input code display

PROCESSOR LKOUT

```
-OPTIONAL TABLES AND INPUT CODES-
     CONSTRAINTS:  CO
      QUANTITIES:  QT
   POWER SUMMARY:  PW
     SOURCE DATA:  SR
    HEAT SUMMARY:  HT
     PRES./TEMP.:  PT
           PLOTS:  PL
            EXIT:  EX  (OR BLANK)
```

TABLE 4-V.- Continued

(v) Display parameter definition table for the optional tables and input code display

PROCESSOR LKOUT

OPTIONAL TABLES AND INPUT CODES

Display parameter label	Parameter definition
--	Provides the user with a list of the possible tables that may be produced by the processor and the two-character codes that are used in response to the processor prompt - TABLES?

TABLE 4-V.- Continued

(w) Power profile display

PROCESSOR LKOUT

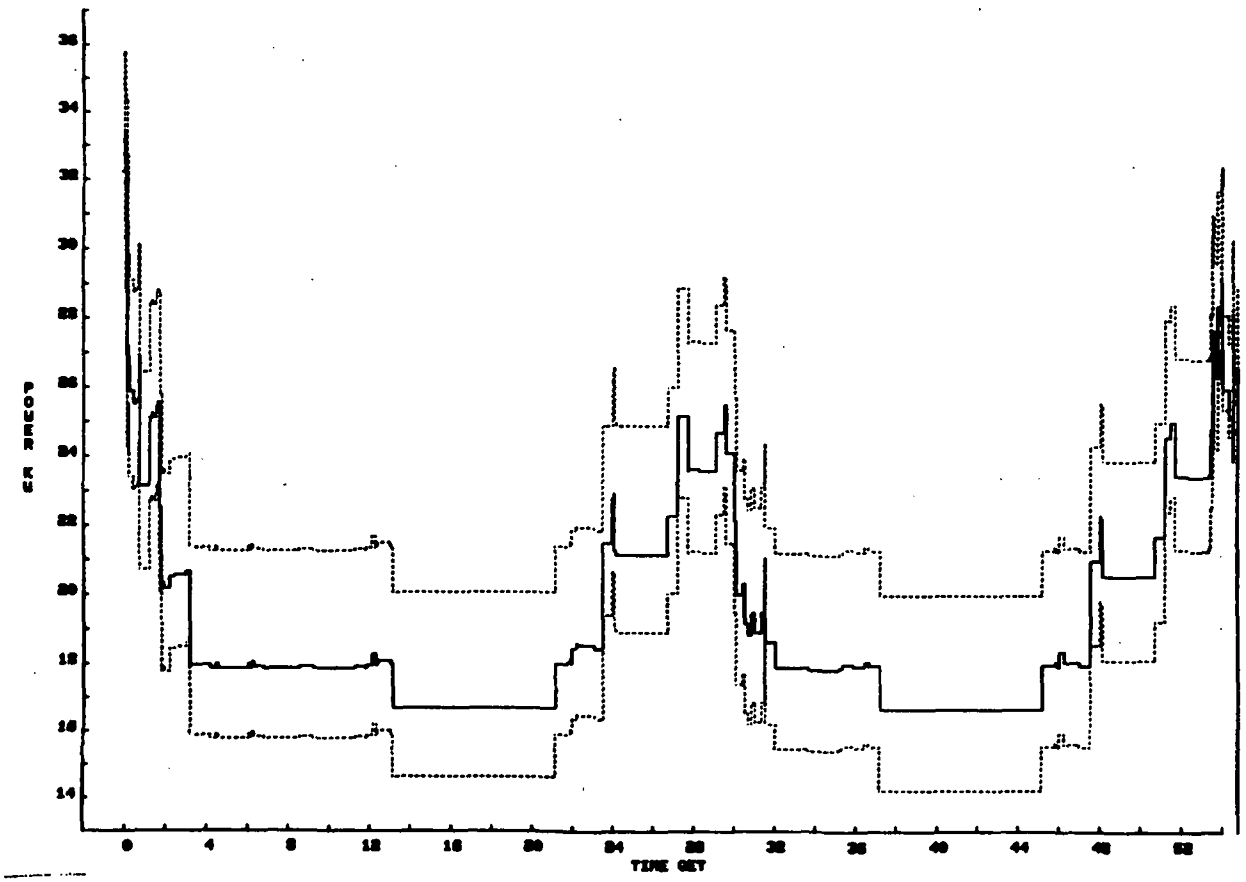


TABLE 4-V.- Continued

(x) Display parameter definition table for the power profile display

PROCESSOR LKOUT

POWER PROFILE

Display parameter label	Parameter definition
POWER KW	Y-axis title - Electrical power, kWh
TIME GET	X-axis title - Ground elapsed time (GET), hr
————	Power required
—·—·—	Statistical peak and minimum

TABLE 4-V.- Continued

(y) Power available display

PROCESSOR LKOUT

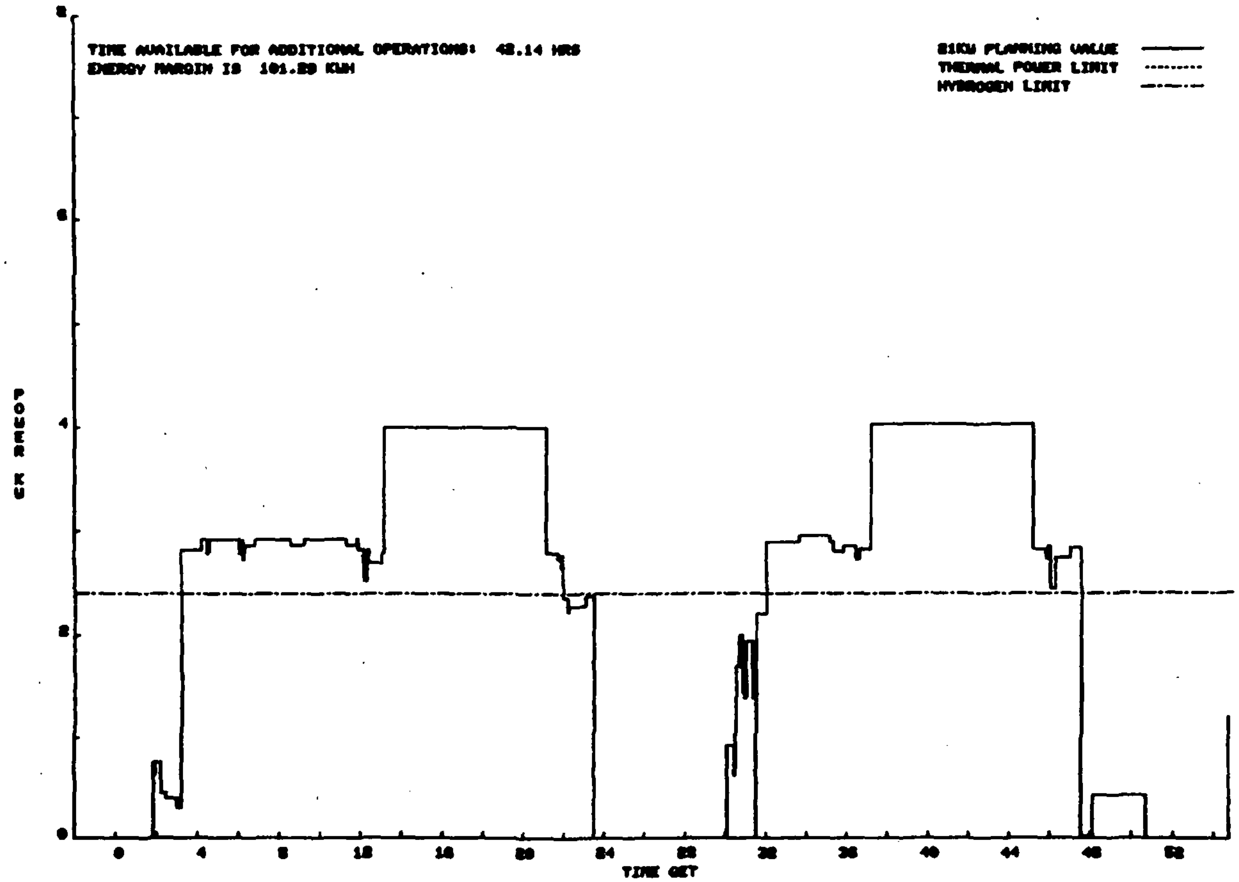


TABLE 4-V.- Continued

(z) Display parameter definition table for the power available display

PROCESSOR LKOUT

POWER AVAILABLE	
Display parameter label	Parameter definition
POWER KW	Y-axis title - Electrical power, kWh
TIME GET	X-axis title - GET, hr
————	Power available
....	Average power requirements
___.___.___.	Maximum power available

TABLE 4-V.- Continued

(aa) kWh consumed display

PROCESSOR LKOUT

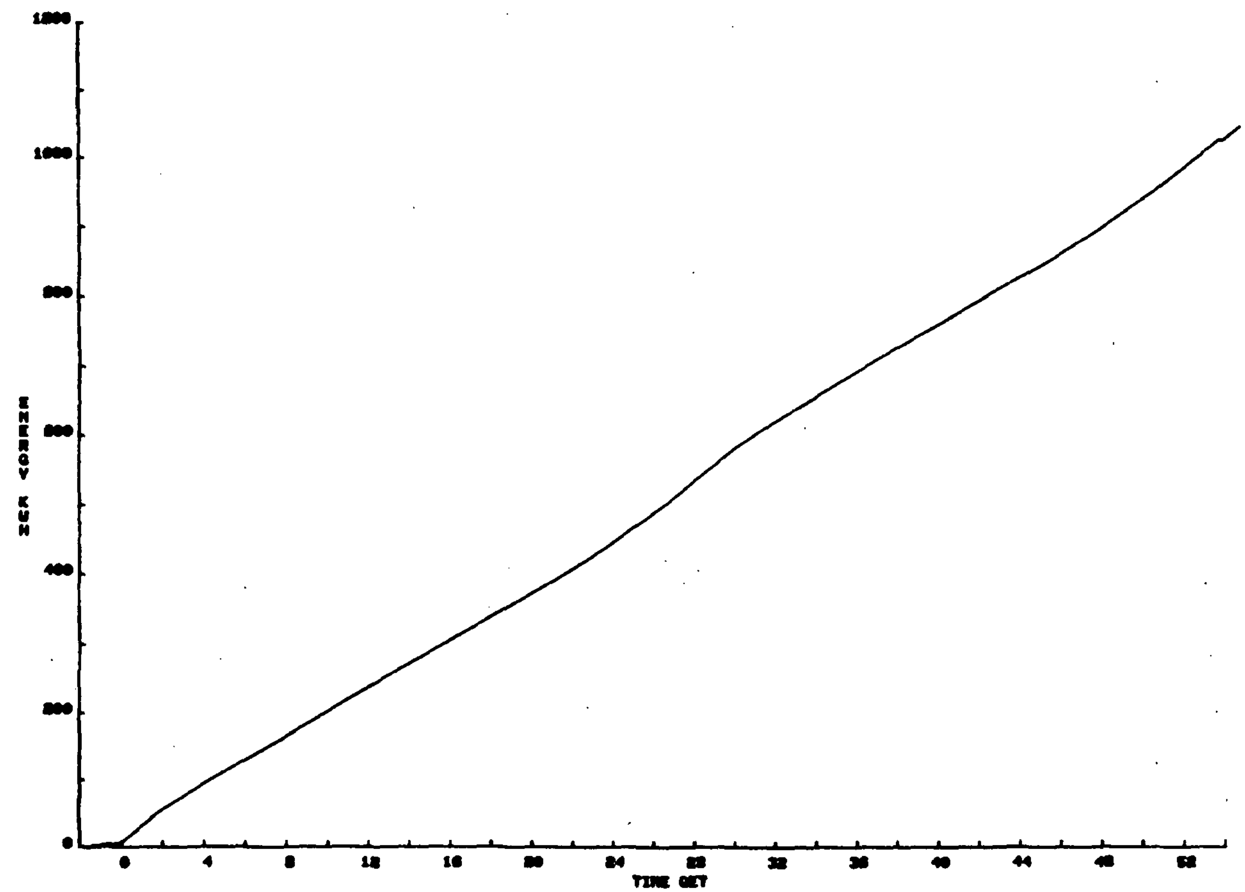


TABLE 4-V.- Continued

(bb) Display parameter definition table for the kWh consumed display

PROCESSOR LKOUT

KWH CONSUMED	
Display parameter label	Parameter definition
ENERGY KWH	Y-axis title - Energy, kWh
TIME GET	X-axis title-GET, hr
________	kWh consumed

TABLE 4-V.- Continued

(cc) Oxygen required display

PROCESSOR <u>LKOUT</u>

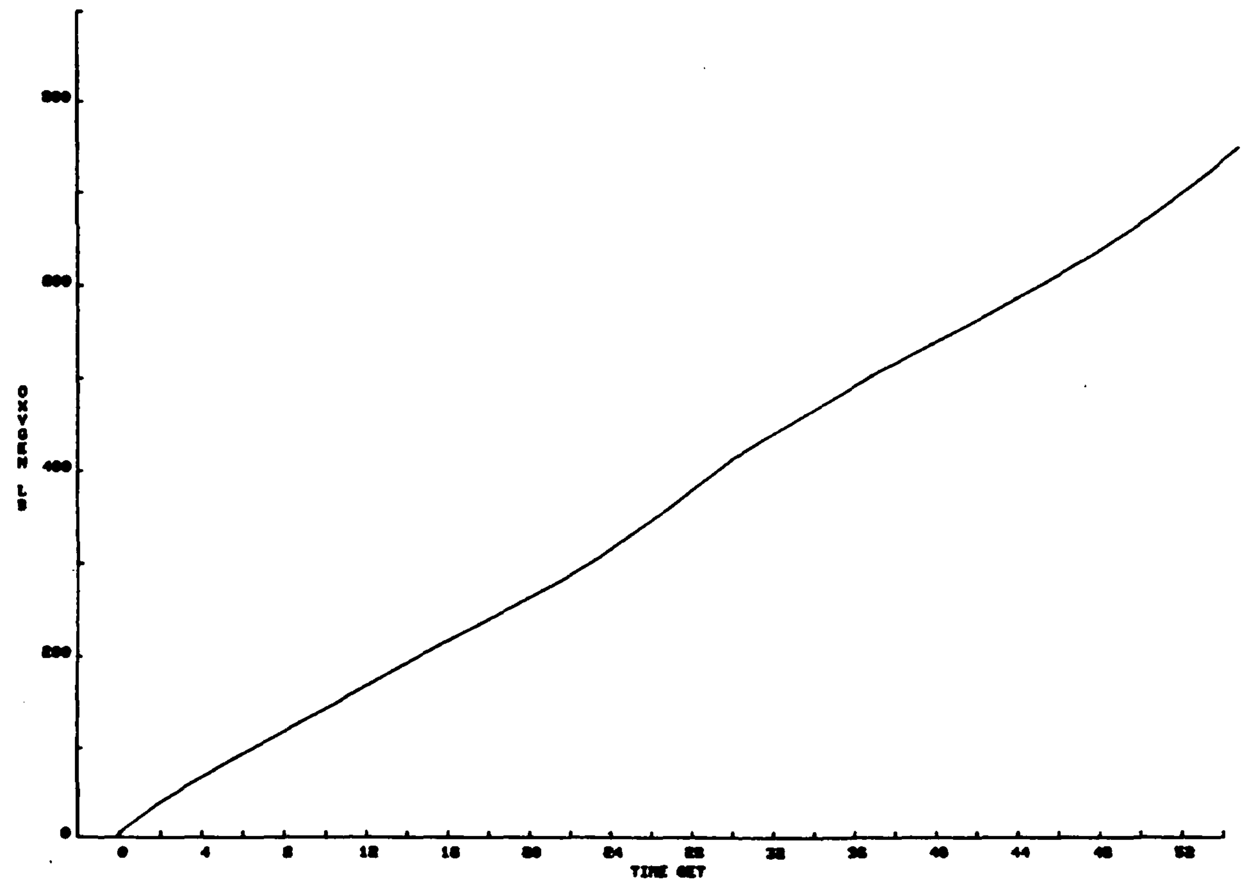


TABLE 4-V.- Continued

(dd) Display parameter definition table for the oxygen required display

PROCESSOR LKOUT

OXYGEN REQUIRED

Display parameter label	Parameter definition
OXYGEN LB	Y-axis title - Oxygen, lb
TIME GET	X-axis title - GET, hr
________	Quantity required

TABLE 4-V.- Continued

(ee) Hydrogen required display

PROCESSOR LKOUT

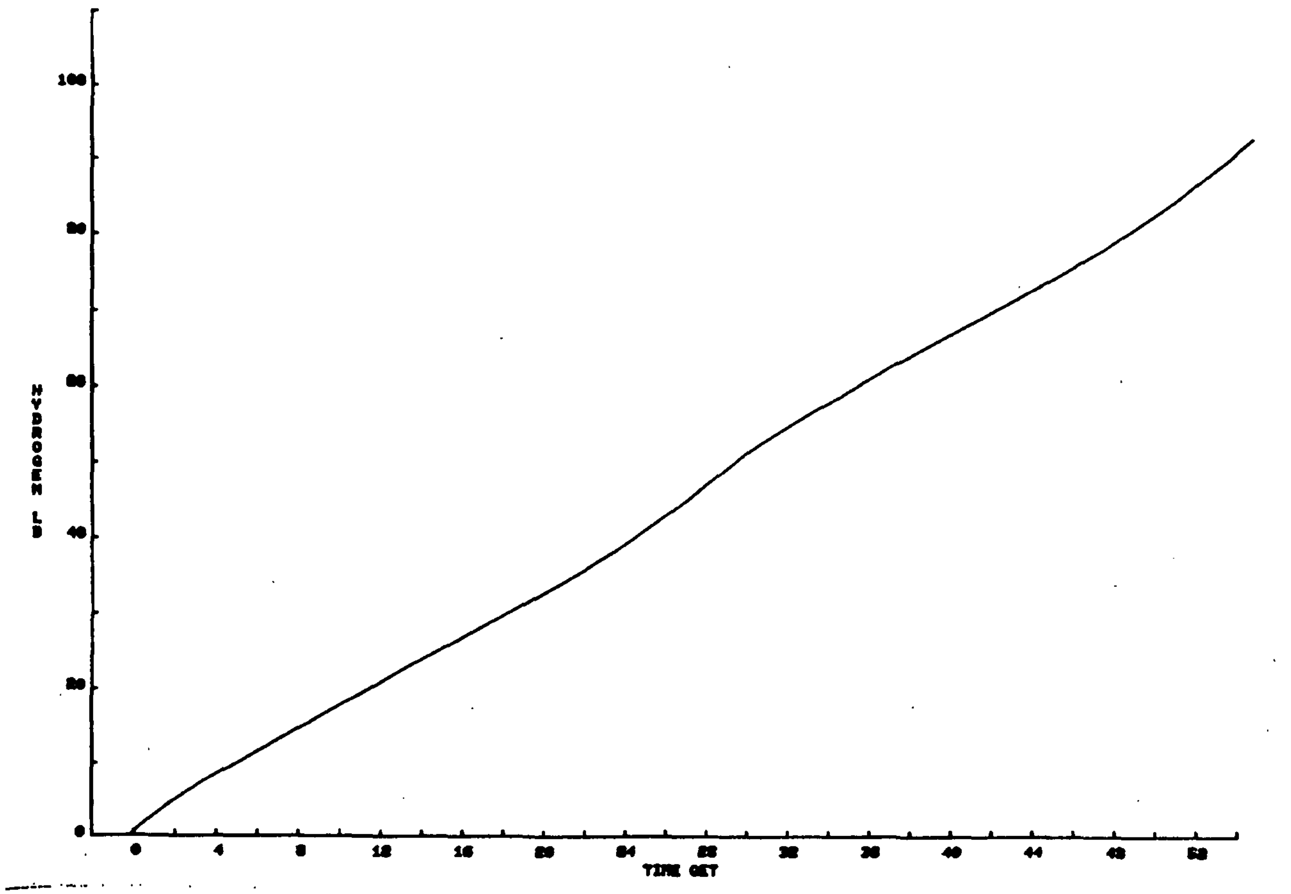


TABLE 4-V.- Continued

(ff) Display parameter definition table for the hydrogen required display

PROCESSOR LKOUT

HYDROGEN REQUIRED

Display parameter label	Parameter definition
HYDROGEN LB	Y-axis title - Hydrogen, lb
TIME GET	X-axis title - GET, hr
________	Quantity required

TABLE 4-V.- Continued

(gg) Nitrogen required display

PROCESSOR <u>LKOUT</u>

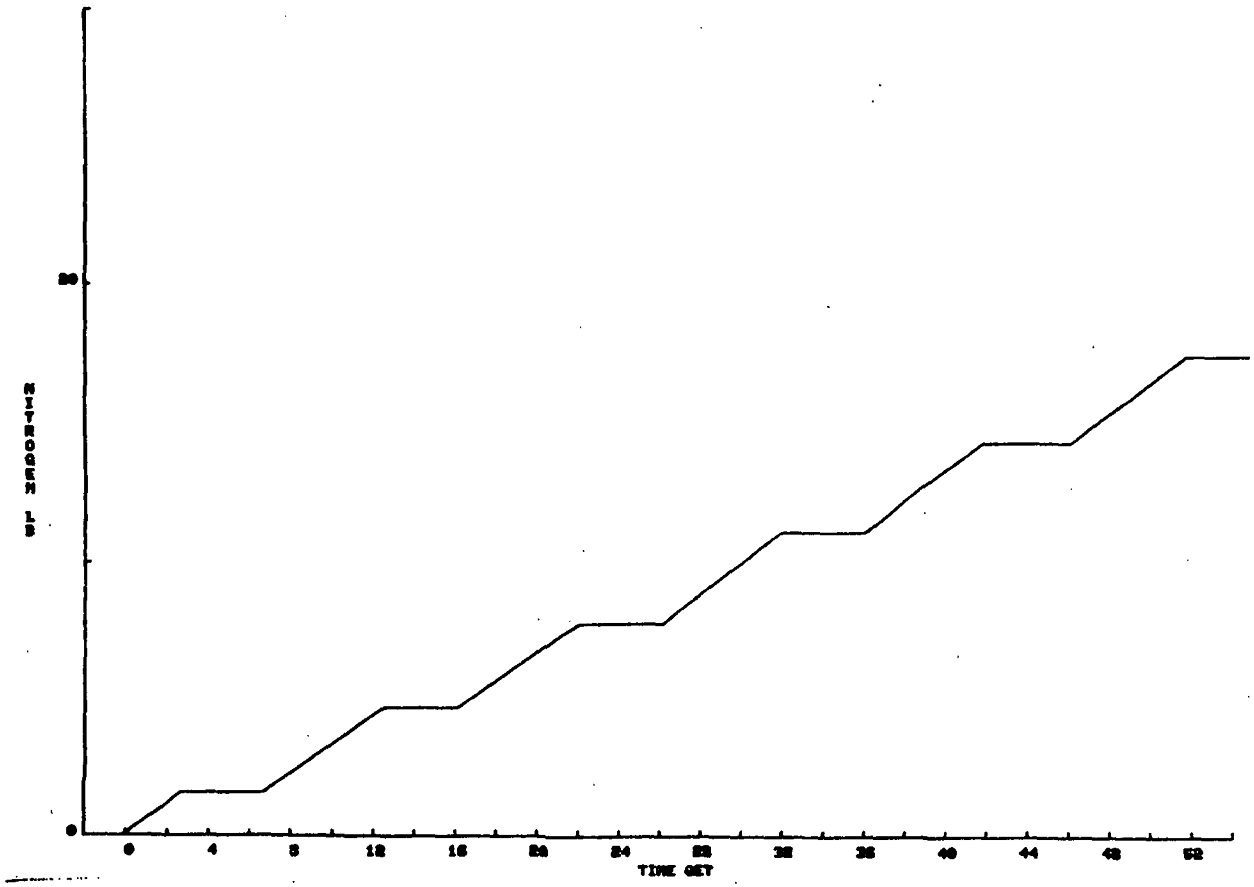


TABLE 4-V.- Continued

(hh) Display parameter definition table for the nitrogen required display

PROCESSOR LKOUT

NITROGEN REQUIRED

Display parameter label	Parameter definition
NITROGEN LB	Y-axis title - Nitrogen, lb
TIME GET	X-axis title - GET, hr
________	Quantity required

TABLE 4-V.- Continued

(ii) Ammonia required display

PROCESSOR LKOUT

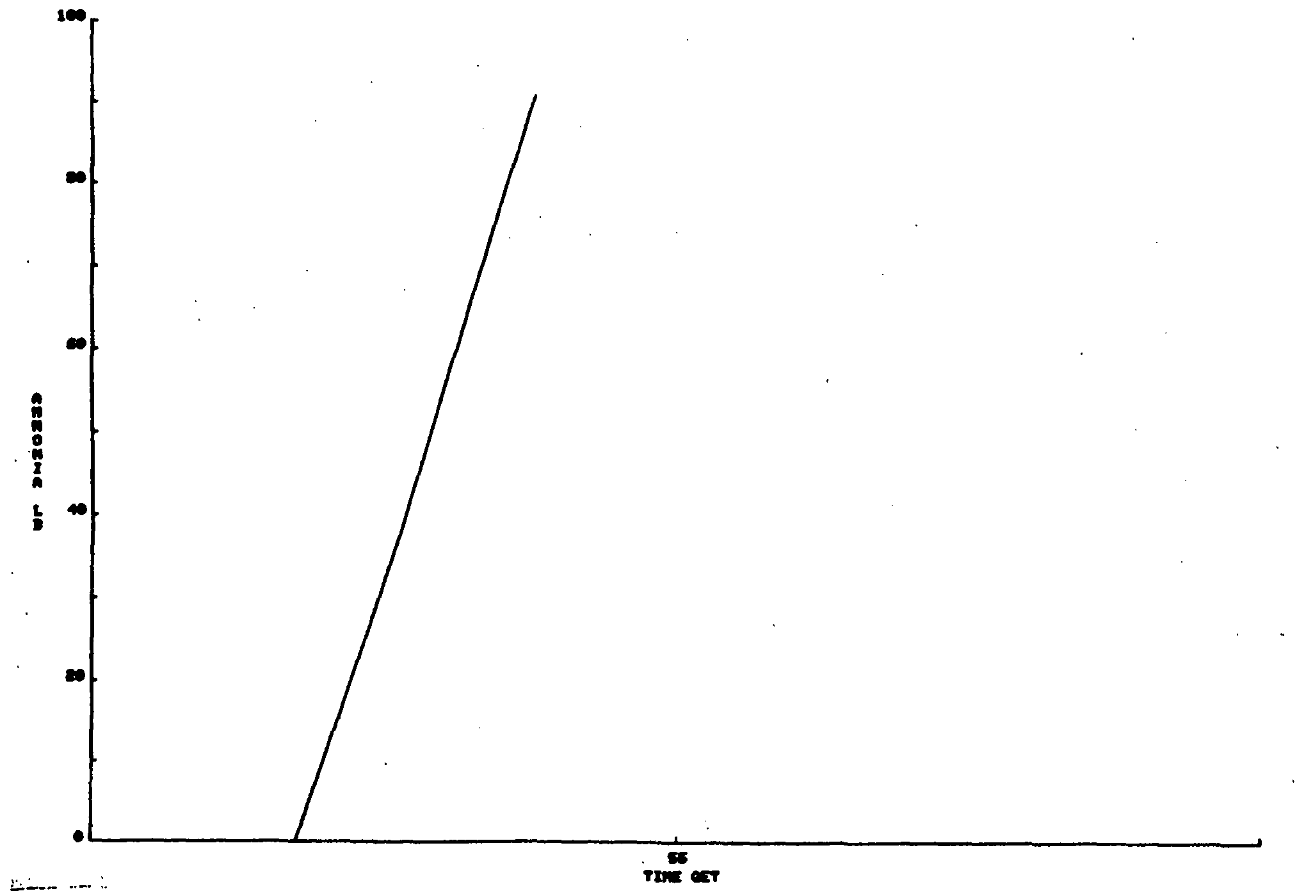


TABLE 4-V.- Continued

(jj) Display parameter definition table for the ammonia required display

PROCESSOR LKOUT

AMMONIA REQUIRED	
Display parameter label	Parameter definition
AMMONIA LB	Y-axis title - Ammonia, lb
TIME GET	X-axis title - GET, hr
————	Quantity required

TABLE 4-V.- Continued

(kk) Waste water produced display

PROCESSOR LKOUT

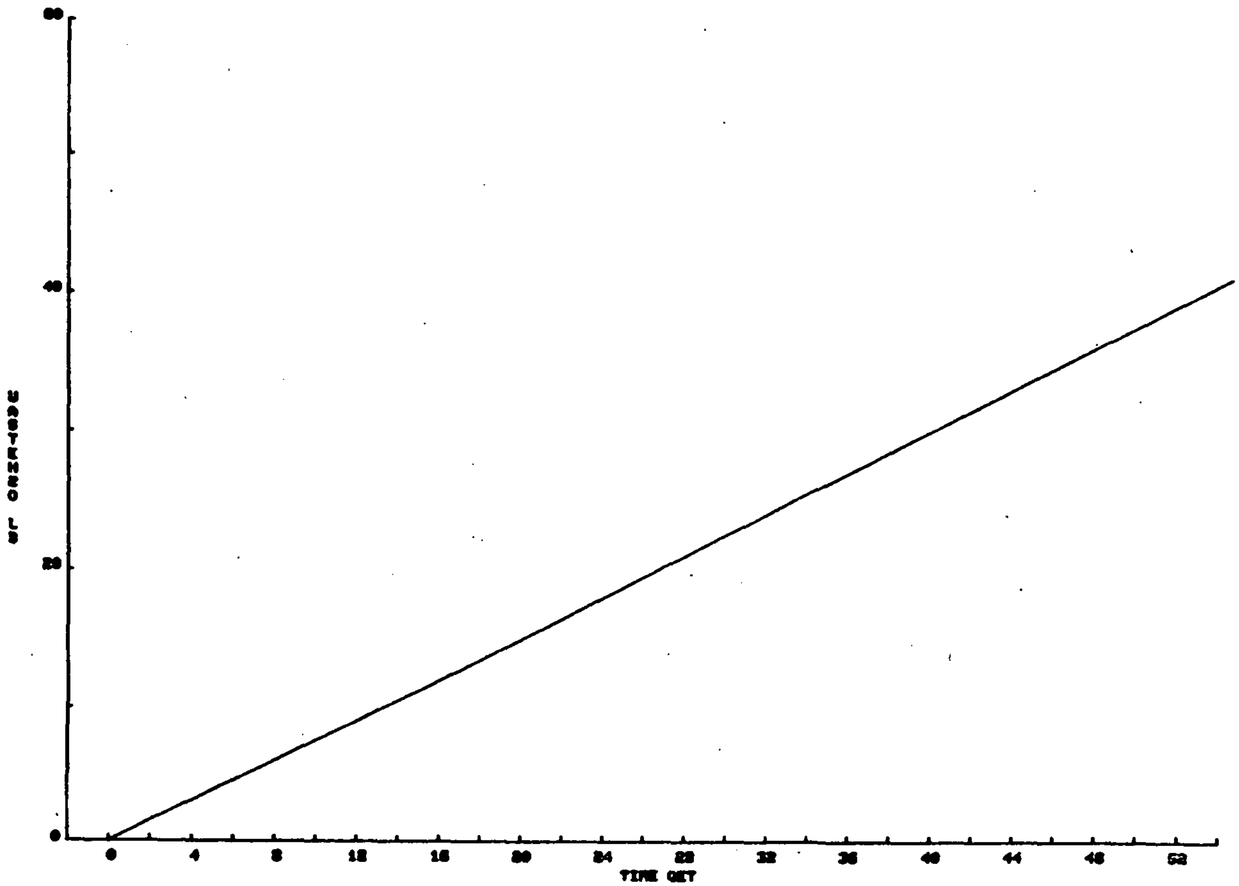


TABLE 4-V.- Continued

(11) Display parameter definition table for the waste water produced display

PROCESSOR LKOUT

WASTE WATER PRODUCED

Display parameter label	Parameter definition
WASTE H_2O LB	Y-axis title - Waste water, lb
TIME GET	X-axis title - GET, hr
________	Quantity produced

TABLE 4-V.- Continued

(mm) Potable water remaining display

PROCESSOR LKOUT

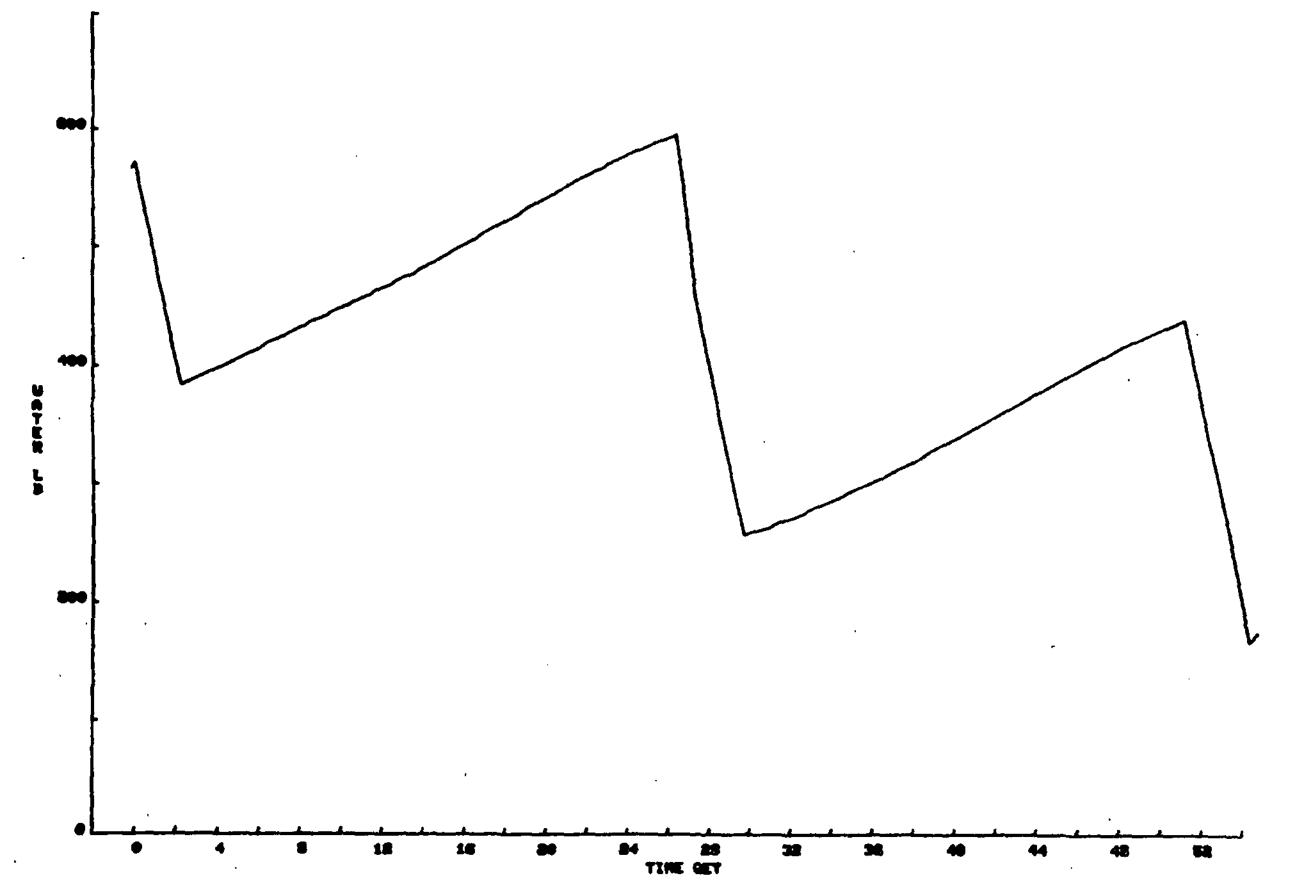


TABLE 4-V.- Continued

(nn) Display parameter definition table for the potable water remaining display

PROCESSOR LKOUT

POTABLE WATER REMAINING

Display parameter label	Parameter definition
WATER LB	Y-axis title - Potable water, lb
TIME GET	X-axis title - GET, hr
___________	Quantity remaining

TABLE 4-V.- Continued

(oo) Cabin temperature display

Processor LKOUT

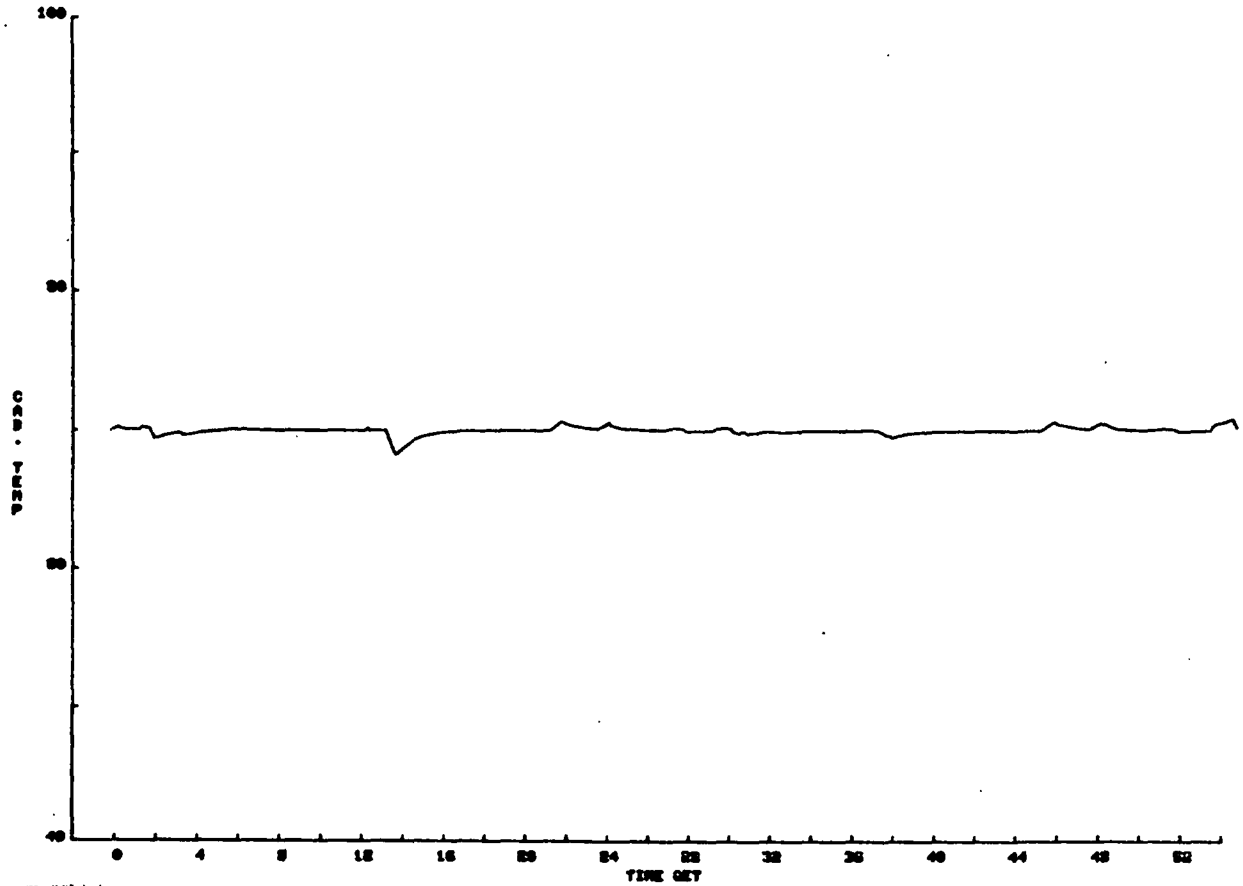


TABLE 4-V.- Continued

(pp) Display parameter definition table for the cabin temperature display

PROCESSOR LKOUT

CABIN TEMPERATURE

Display parameter label	Parameter definition
CAB. TEMP	Y-axis title - Cabin temperature, °F
TIME GET	X-axis title - GET, hr
———	Temperature

TABLE 4-V.- Continued

(qq) Avionics outlet temperature display

Processor LKOUT

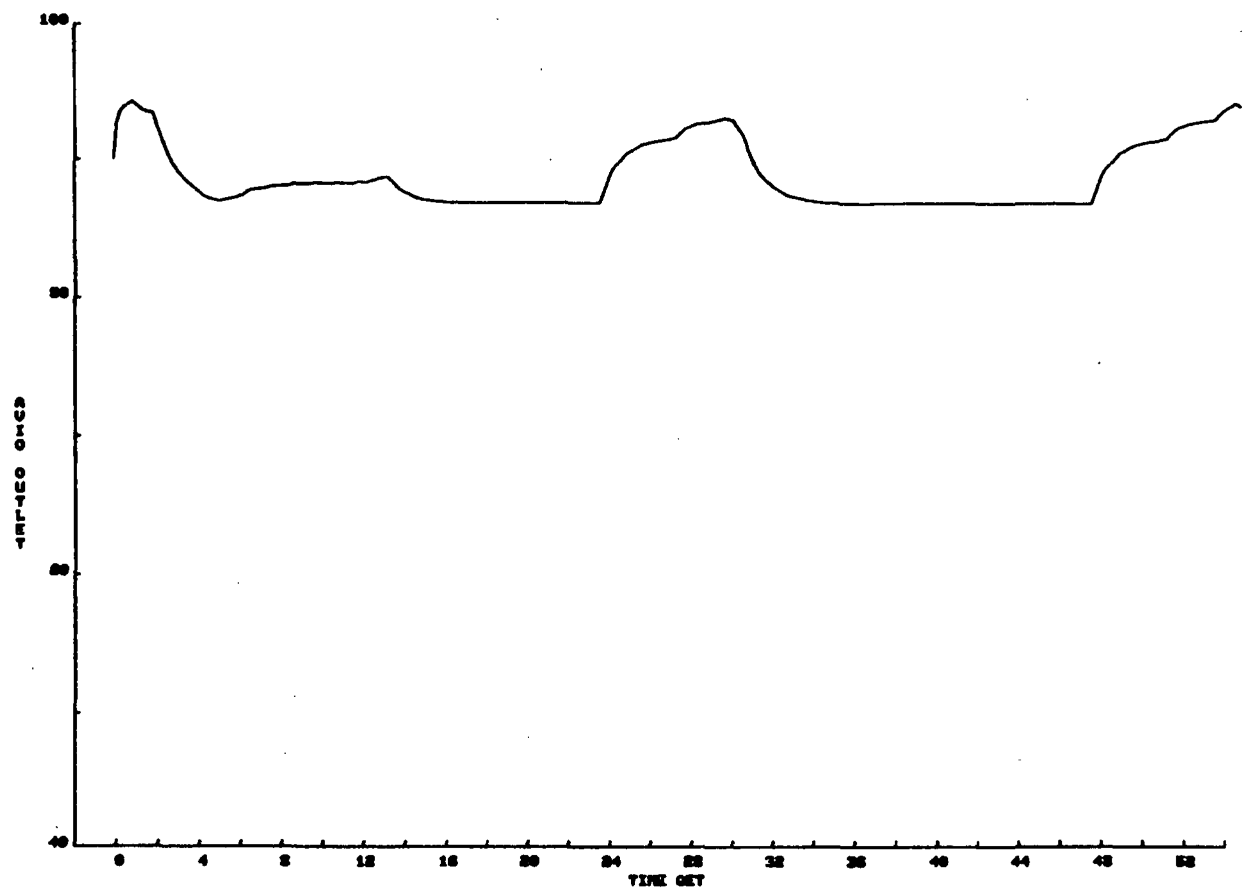


TABLE 4-V.- Continued

(rr) Display parameter definition table for the avionics outlet temperature display

PROCESSOR LKOUT

AVIONICS OUTLET TEMPERATURE

Display parameter label	Parameter definition
AVIO OUTLET	Y-axis title - Avionics outlet temperature, °F
TIME GET	X-axis title - GET, hr
________	Avionics outlet temperature

TABLE 4-V.- Continued

(ss) Hydraulics inlet temperature display

Processor <u>LKOUT</u>

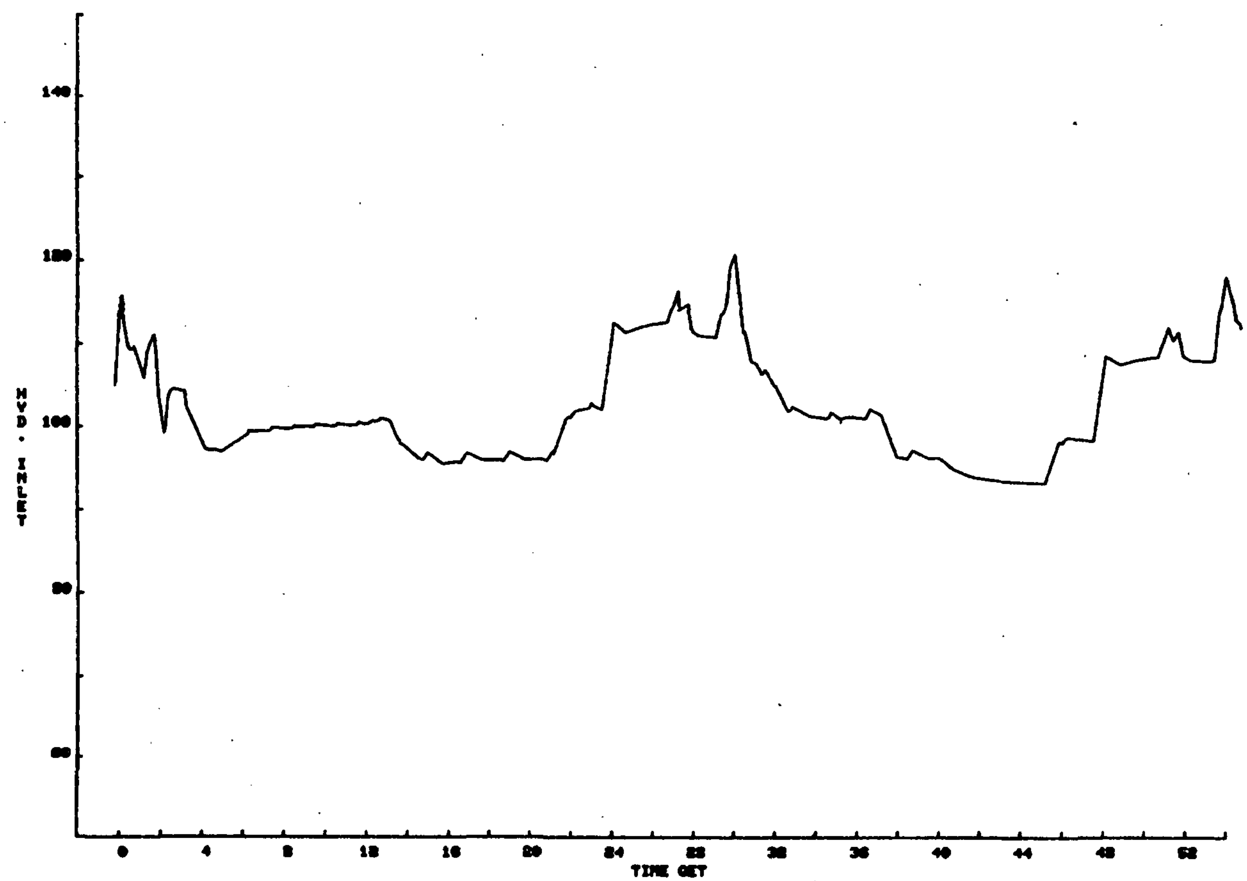


TABLE 4-V.- Continued

(tt) Display parameter definition table for the hydraulics inlet temperature display

PROCESSOR LKOUT

HYDRALICS INLET TEMPERATURE, °F	
Display parameter label	Parameter definition
HYD. INLET	Y-axis title - Hydraulics inlet temperature, °F
TIME GET	X-axis title - GET, hr
————	Hydraulics inlet temperature

TABLE 4-V.- Continued

(uu) Radiator inlet temperature display

Processor LKOUT

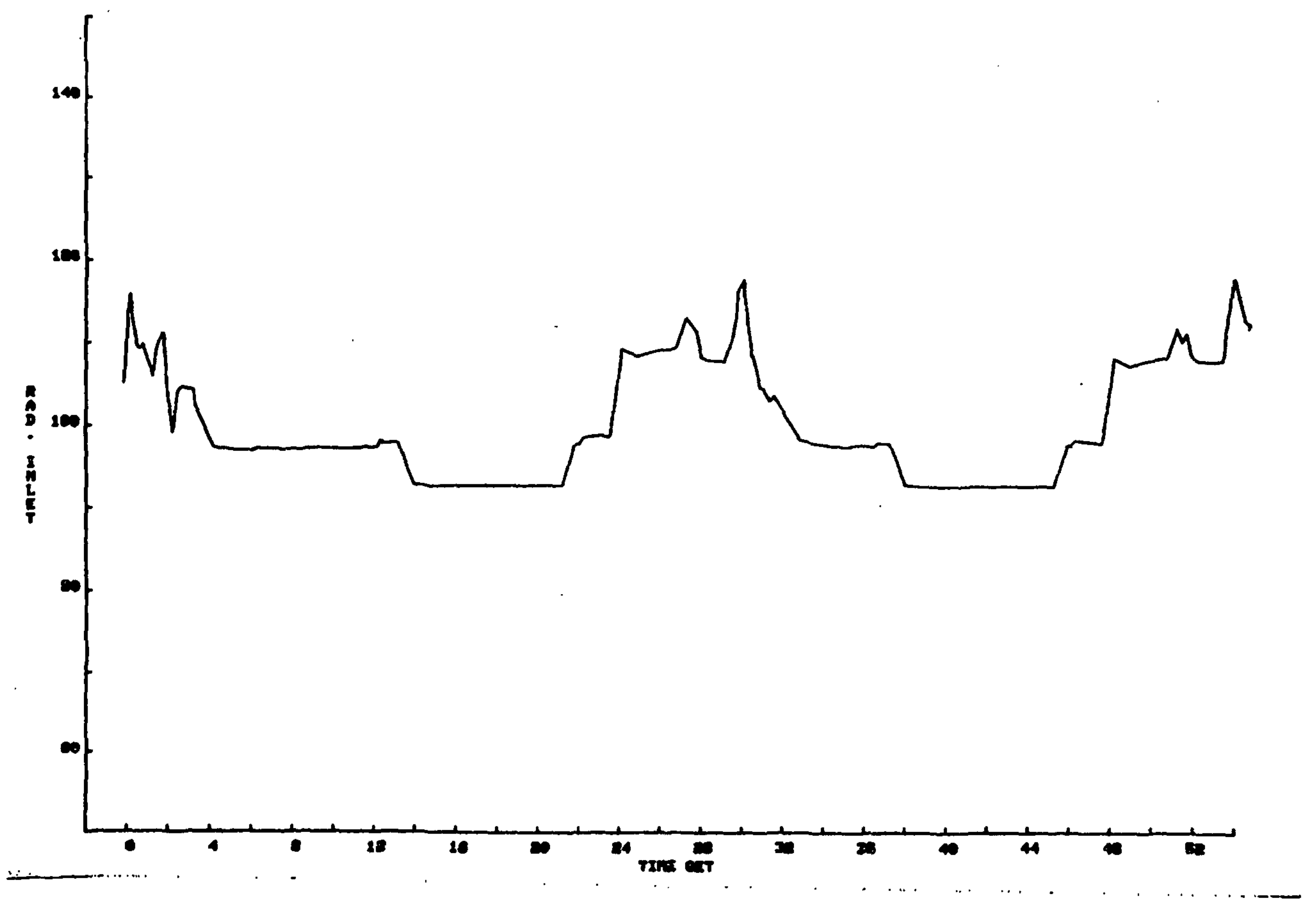


TABLE 4-V.- Continued

(vv) Display parameter definition table for the radiator inlet temperature display

PROCESSOR LKOUT

RADIATOR INLET TEMPERATURE

Display parameter label	Parameter definition
RAD. INLET	Y-axis title - Radiator inlet temperature, °F
TIME GET	X-axis title - GET, hr
________	Radiator inlet temperature

TABLE 4-V.- Continued

(ww) Cooling outlet temperature display

Processor LKOUT

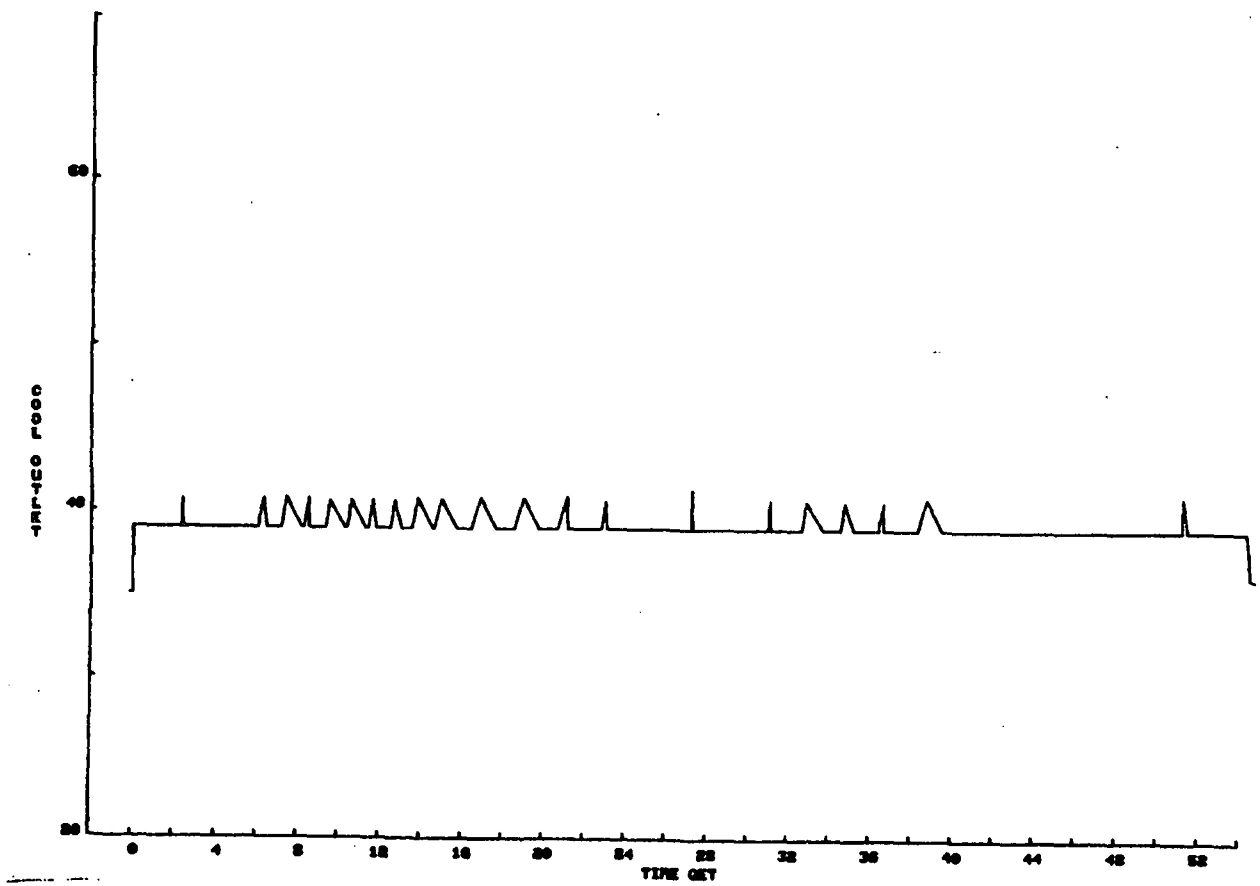


TABLE 4-V.- Concluded

(xx) Display parameter definition table for the cooling outlet temperature display

PROCESSOR LKOUT

COOLING OUTLET TEMPERATURE

Display parameter label	Parameter definition
COOL OUTLET	Y-axis title - Cooling outlet temperature, °F
TIME GET	X-axis title - GET, hr
________	Cooling outlet temperature

TABLE 4-VI.- PROCESSOR MESSAGE TABLE

PROCESSOR LKOUT

MSG no.	Message ID block	Message text block and explanation
1	*LKOUT*	DATA FILE PROBLEM XXX Meaning: Unable to open the /CDATC file, the error number is XXX, or unable to read the data portion of the file. Severity: Terminal Action required by user: Exit the processor when you get the TABLE? prompt
2	*LKOUT*	FILE PROBLEM XXX Meaning: Unable to read the data portion of the file /CDATC, the error number is XXX Severity: Terminal Action required by the user: Exit the processor when you get the TABLE? prompt

TABLE 4-VII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR LKOUT

Processor name	Processor abstract prompt (maximum 256 characters)
LKOUT	LKOUT provides CASKU quick output summaries of electrical energy and consumables required. Detailed tables and plots of power, quantities, heat, pressure/temperature, constraints, and consumables remaining may be obtained.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
TIMREF	Enter "ƀƀ" for ground elapsed time (GET) time reference or "GM" for Greenwich mean time (GMT) time reference.
SQTY	Enter initial quantities of O_2, N_2, H_2, and NH_3 to be used by LKOUT for tables and plots. If none, default values are 0.0.
DATAFL	Name of the DRDE file that contains the input data.
PROCON(1)	Logical unit number of the DRDE file.
PROCON(2)	Logical unit number for the output LU for plotting
PROCON(3)	Print flag - six for line printer.
LDAY	Day of launch
LHOUR	Hour of launch
LYEAR	Year of launch

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

LANDING OPPORTUNITIES PROCESSOR (LOPT)

1.0 PURPOSE

The LOPT processor computes landing opportunities of an orbiting vehicle to ground landing stations based on an orbital crossrange constraint.

2.0 FUNCTIONAL DESCRIPTION

The trajectory information may be input either through a trajectory disk file, which had been generated by the INVAR processor, or by inputting a single invariant elements state vector. The processor then generates a table that contains the times and longitudes of the ascending node, and generates corresponding sets of invariant elements for all orbits that span the start and stop times.

The processor then cycles through logic that determines the landing opportunities on a site-by-site basis. For each site, the processor computes a range of ascending nodes for which the crossrange constraint will be satisfied. This range of nodes is an approximation because the flight time from the ascending node to closest point of approach must be approximated. The computed range of nodes is therefore biased to ensure that all possible orbits containing solutions are considered. However, for very high elliptical orbits, the bias number may need to be increased to catch all possible solutions.

If the ascending node for a given orbit is within the range of nodes, an iterative method is used to find that orbital closest point of approach (CPA). If the crossrange at CPA is less than the crossrange constraint, the output quantities are computed and displayed. The output quantities are orbit number, crossrange, time of landing after sunrise, time of landing before sunset, ground elapsed time of landing, and local landing time. The landing time is obtained by adding a bias time to the orbital CPA time.

3.0 ASSUMPTIONS AND LIMITATIONS

Orbital propagation using invariant elements does not reflect changes to the orbit due to atmospheric drag between the states stored in the trajectory file. Therefore, the vectors in the trajectory file should be stored close enough together so that these inaccuracies will be small enough for the users requirements. Periodic perturbations to the orbit are not considered. Only elliptical orbit formulation is provided. Because of the array storage constraints, the processor cannot process more than 300 orbits or a trajectory segment requiring more than 50 invariant elements arrays.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The interface table for the LOPT processor is contained in table 4-I. The IVEC flag determines whether the trajectory data are a single input state vector (LOVEC) or a disk file (VECFIL). The ISIT flag determines whether the input site definitions are through a single site (GSIT) or a disk file (SITFIL). The OUTFLG parameter determines whether the output quantities are also to be stored on a disk file (NAMOUT).

b. Interface table data array definitions - The PROCON array is defined in table 4-II.

c. Processor solicited (prompted) input - The only solicited input is defined in table 4-III.

d. Processor displays and display parameter definition tables - The processor display is presented in table 4-IV, and the display parameter definitions table is presented in table 4-V. This is a variable length display; it will present all of the solutions calculated by the processor.

e. Processor message table - The processor messages are presented in table 4-VI.

f. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR LOPT

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
IVEC	AWA	2CH	I	1	1		Vector source flag "IN" = single input state (LOVEC) "DF" = vector disk file as specified by VECFIL.
LOVEC	AWA	Real	I	30	15		Input invariant elements state vector; required when IVEC = "IN"
VECFIL	Disk	Real	I	—	--		Name of invariant elements disk file; required when IVEC = "DF"
ISIT	AWA	2CH	I	1	1		Flag indicating source of landing site data "IN" = single input landing site "DF" = landing site disk file as specified by SITFIL
SITNAM	AWA	6CH	I	3	1		Name of input landing site; required when ISIT = "IN"
GSIT	AWA	Real	I	6	3		Radius, geocentric latitude, and longitude of the input landing site; required when ISIT = "IN"
LZONE	AWA	Intg	I	1	1		Time zone of input landing site. Input as number of hours earlier than GMT; required when ISIT = "IN"
SITFIL	AWA	6CH	I	3	1		Name of landing site disk file; required when ISIT = "DF"
GETS	AWA	Real	I	6	3		Start time (hr, min, sec) relative to base time

NOTES

CLASS	TYPE			USE
AWA	Free	2CH	72CH	I = Input
Disk	Intg	6CH	Mix	O = Output
	Real	18CH	Symb	I/O = Input/Output
	Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LOPT

Parameter keyword name	Class	Type	Use	Size	Array dimen- sion (I,J)	Values stored in default interface table	Definition
GETF	AWA	Real	I	6	3		Final time (hr, min, sec) relative to base time
XRNG	AWA	Real	I	2	1		Crossrange constraint (n. mi.)
INORB	AWA	Intg	I	1	1		Initial orbit counter; corresponds to the orbit counter for GETS
ISTIM	AWA	2CH	I	1	1	"ST"	Flag indicating daylight or standard time for local landing time "ST" = standard time "DL" = daylight savings time
OUTFLG	AWA	2CH	I	1	1	"NO"	Flag that indicates whether or not the output quantities are to be output to the disk file specified by NAMOUT. "NO" = no output to disk "YE" = output to disk file
NAMOUT	Disk	Free	O	—	--		Name of the output disk file
GMTR	AWA	Real	I	2	1	!SESCN (11)	Reference time for GET computations
CFA	AWA	Real	I	2	1	!SESCN (27)	Conversion factor for angles - external to internal
CFD	AWA	Real	I	2	1	!SESCN (29)	Conversion factor for distance
CFT	AWA	Real	I	2	1	!SESCN (31)	Conversion factor for time
ROMEGE	AWA	Real	I	2	1	!SESCN (81)	Rotation rate of Earth
PI	AWA	Real	I	2	1	!!GLCN (59)	Ratio of circle circumference to diameter

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR LOPT

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TWOPI	AWA	Real	I	2	1	!!GLCN (61)	Twice pi
DTWOPI	AWA	Dubl	I	3	1	!!GLCN (68)	Double precision TWOPI
AMILE	AWA	Real	I	2	1	!!GLCN (89)	Feet per nautical mile
AMIN	AWA	Real	I	2	1	!!GLCN (97)	Seconds per minute
AHOUR	AWA	Real	I	2	1	!!GLCN (99)	Seconds per hour
ADAY	AWA	Real	I	2	1	!!GLCN (101)	Seconds per day
PROCON	AWA	Free	I	13	13		LOPT constants
SCOEF	AWA	Real	I	38	14	!SCOEF	Sun constants array for analytic ephemeris model

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR LOPT

Array name	Index location	Default value	Definition	
LOVEC	1		Vector time (standard state vector format)	
	2		Invariant elements semimajor axis	Standard position/ velocity state vector definition (items 2–14)
	3		Invariant elements eccentricity	
	4		Invariant elements inclination	
	5		Invariant elements argument of perigee	
	6		Invariant elements argument of ascending node	
	7		Invariant elements mean anomaly	
	8		Invariant elements mean motion	
	9		Invariant elements rate of change of perigee	
	10		Invariant elements rate of change of the ascending node	
	11		CD - drag coefficient	
	12		Area for drag computation	
	13		Vehicle gross mass	
	14		Vector type code (=3120.)	
	15		Vector name code	
GSIT	1		Radius of input landing site	
	2		Geocentric latitude of input landing site	
	3		Longitude of input landing site	
GETS	1		Start time relative to reference time (hour)	
	2		Start time relative to reference time (minutes)	
	3		Start time relative to reference time (seconds)	
GETF	1		Final time relative to reference time (hours)	
	2		Final time relative to reference time (minutes)	
	3		Final time relative to reference time (seconds)	

TABLE 4-II.- Concluded

PROCESSOR LOPT

Array name	Index location	Default value	Definition
PROCON	1	0	Debug print flag 0 = no print 1 = debug print
	2	0	Output destination flag 0 = terminal Positive number = LU of output device
	3	600.	Bias time to account for the difference between the orbital closest point of approach and the actual landing time (seconds)
	5	1.5853407	Angle between the zenith and local sunrise/sunset (reference - Supplement to the American Ephemeris and Nautical Almanac 90° 50 feet)
	7	1.7453292	Bias that is applied to nodal range values to account for uncertainty in nodal computations due to uncertainty in flight time (default = 1 degree)
	9	2.	Iteration tolerance for CPA iteration
	11	48	Maximum number of lines per page
	12	20	Cartridge reference number for data files
	13	20	Maximum number of blocks for the output file
SCOEF	1 . . . 22	!SCOEF	See table 7.2-II of JSC IN 78-FM-60, vol. I for definition of array locations

TABLE 4-III.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR LOPT

Prompt	Meaning	Valid responses
Maximum line number	The maximum number of output lines has been reached that was specified in LOCON (11).	After printing a copy, if desired, enter a space and carriage return to obtain remaining output.

TABLE 4-IV.- PROCESSOR DISPLAY TABLE

PROCESSOR <u>LOPT</u>

```
 LANDING OPPORTUNITIES

 NAME=αααααα   LAT=±XX.XXX   LONG=±XXX.XXX   RADIUS=XXXXXXXX.

  ORBIT      XRNG      TASR      TBSS      GETL          LLT       SELV
 XXX  α     XXXXX     XXX  XX,  XXX  XX   XXX  XX  XX   XXX  XX   ±XXX.
```

TABLE 4-V.- DISPLAY PARAMETER DEFINITION TABLE

PROCESSOR LOPT

LANDING OPPORTUNITIES

Display parameter label	Parameter definition
NAME	Landing site name
LAT	Landing site geocentric latitude
LONG	Landing site geocentric longitude
RADIUS	Landing site radius magnitude from the center of the Earth
ORBIT	Orbit number and indication whether pass is ascending (A) or descending (D)
XRNG	Orbital crossrange at closest point of approach minus (-) indicates that the site is toward the positive angular momentum vector relative to the orbit. Plus (+) indicates the landing site is toward the negative angular momentum vector relative to the orbit.
TASR	Time of landing after sunrise time (hr:min); negative indicates landing time is before sunrise.
TBSS	Time of landing before sunset time (hr:min); negative time indicates landing time is after sunset.
GETL	Ground elapsed time of landing (days:hr:min)
LLT	Local landing time (hr:min)
SELV	Sun elevation angle at landing

TABLE 4-VI.- PROCESSOR MESSAGE TABLE

PROCESSOR LOPT

MSG no.	Message ID block	Message text block and explanation
1	*LOPT*	"STATE VECTOR CODE NOT INVARIANT ELEMENTS" Meaning: Other than invariant elements, state vector is specified. Severity: Processor terminates. Action required by user: Input invariant elements state vector.
2	*LOPT*	"OPEN FILE ERROR=", IIII, "FILE NAME=", AAAAAA Meaning: Either the trajectory file or the ground sites file is not in the proper configuration. Severity: Processor terminates. Action required by user: Check the file name, etc., of the trajectory and ground sites files.
3	*LOPT*	"FILE CREAT ERROR=", IIII, "FILE NAME=", AAAAAA Meaning: Error occured when an attempt was made to create the output file. Severity: Processor terminates; no file is created. Action required by user: Check the name, etc., of the output file for redundancy.
4	*LOPT*	"NO OPPORTUNITIES; OUTSIDE TRAJECTORY LIMITS" Meaning: Ground site is outside the limits of the trajectory, and no opportunities will be generated for that site. Severity: None. Action required by user: None.
5	*LOPT*	"EXCEEDED MAX NO. OF ORBITS" Meaning: The trajectory spans more than the allowable number of orbits. Severity: Only timespan corresponding to the maximum number of orbits will be processed. Action required by user: Rerun processor to obtain desired timespan not included in the output.

TABLE 4-VI.- Concluded

PROCESSOR LOPT

MSG no.	Message ID block	Message text block and explanation
6	*LOPT*	"EXCEEDED MAX NO. OF VECTOR ENTRIES IN ARRAYS" Meaning: The timespan requires more vector storage area than available. Severity: Processor only generates data for the timespan over which the vector storage capability is not exceeded. Action required by user: Rerun processor with a different timespan to obtain desired data not included.

TABLE 4-VII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR LOPT

Processor name	Processor abstract prompt (maximum 256 characters)
LOPT	LOPT computes the landing opportunities of an orbiting vehicle to ground-landing locations based on orbital crossrange constraints.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
	Use definitions on interface table definitions, table 4-I, except for the following.
PROCON	LOPT constants: (1) Debug (2) Output LU (3) Landing bias time (sec) (5) Sunrise/Sunset to zenith angle (rad) (7) Node range bias angle (rad) (9) Iteration tolerance (11) Maximum lines/page (12) Output file cartridge reference number (13) Maximum number of blocks in output file

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - MAIN PROGRAM LOPT

5.1.1 Purpose

The Landing Opportunities Program (LOPT) is the main program routine for the landing opportunities processor. Most of the processor computations are performed in this routine.

5.1.2 Functional Description

All the inputs required for this routine are obtained through the interface table (table 4-I of this processor) and through disk files for the trajectory input and for landing sites input. The processor computes the landing opportunities based on a crossrange constraint. The outputs are displayed on the terminal and optionally written to a disk file. Three subheadings under the method section provide an explanation for the trajectory file processing, the closest point of approach (CPA) calculations, and the landing site sunrise/sunset computations.

5.1.3 Assumptions and Limitations

a. Elliptical orbit propagation is assumed.

b. Periodic perturbations to the orbit are not included in the computations.

5.1.4 Method

a. Trajectory File Processing - If the trajectory input is through a disk file generated by the INVAR processor, special processing must be performed in order to determine the correct set of invariant elements. The structure of the invariant elements disk file is such that the vectors are stored in a time sequenced order, with the last record being indicated by a vector time earlier than the previous record.

   The first call to the RVECF routine opens the trajectory file and reads the first two trajectory records. A vector switch time (TSWCH) is either computed as the average of the two vector times, or set to a large number (2 X 10^9) if the second record is the last record. The RVECF routine is repeatedly called to read the next trajectory record and compute a new switch time until TSWCH is greater than the start time. This ensures that the start time is between the times of the two current trajectory records in memory. The current invariant elements set is the earlier of the two vectors. This set is stored in the first elements array.

   The processor then generates tables which contain the times and longitudes of ascending nodes, and invariant elements sets for all orbits from the start to the final time. The processor first computes the first ascending nodal time prior to the start time by calls to the ADVU and TAU routines. This time along with the longitude of the ascending node and an invariant elements set identifier are stored as the first entries in corresponding arrays. The processor then computes the time of arrival at the next ascending node (TNEXT). If TNEXT is less than the switch time nothing is done to the invariant elements set identifier or the memory storage of the invariant elements sets. If TNEXT is greater than TSWCH, the RVECF routine is called until TSWCH is greater than TNEXT. The invariant elements identifier is incremented and a new set of elements is stored in the invariant elements arrays. This procedure continues until the last orbit is processed.

The processor then begins the processing of each ground site. The ascending node in the nodal array is tested against the range of nodes for that particular site. Each time a change in the invariant elements identifier array is detected, a new set of elements is restored as the current set.

b. Closest Point of Approach - An iterative method is used to obtain the time of closest point of approach for those orbits which are potential solutions. Since the time of the ascending node is known for each orbit, the method computes the delta time from the nodal crossing time. This is done as follows (see figure 5.1-1).

$$\delta_S = \lambda + \omega_e * t$$

$$h = h_K + \dot{h} * (t - t_{VEC})$$

$$\hat{S} = \begin{bmatrix} \cos \delta_S \cos \phi \\ \sin \delta_S \cos \phi \\ \sin \phi \end{bmatrix}$$

$$\hat{H} = \begin{bmatrix} \sin h \sin i \\ -\cos h \sin i \\ \cos i \end{bmatrix}$$

$$\hat{N} = \begin{bmatrix} \cos h \\ \sin h \\ 0 \end{bmatrix}$$

$$\hat{c} = \text{unit } (\hat{H} * (\hat{S} \times \hat{H}))$$

$$\delta = \lambda + \omega_E * t$$

$$h = h_K + \dot{h} * (t - t_{VEC})$$

$$\hat{S} = \begin{bmatrix} \cos\delta\cos\phi \\ \sin\delta\cos\phi \\ \sin\phi \end{bmatrix}$$

$$\hat{H} = \begin{bmatrix} \sin h \sin i \\ -\cos h \sin i \\ \cos i \end{bmatrix}$$

$$\hat{N} = \begin{bmatrix} \cos h \\ \sin h \\ 0 \end{bmatrix}$$

$$\hat{c} = \text{unit } (\hat{H} \times (\hat{S} \times \hat{H}))$$

$$\mu_{CPA} = \cos^{-1} (\hat{C} * \hat{N})$$

$$\Delta t_N = \mu_{CPA}/\eta$$

$$g_{CPA} = g_N + \Delta t_N * \dot{g}$$

$$F_c = \mu_{CPA} - g_{CPA}$$

$$E_c = 2\tan^{-1} ((1 - e)/(1 + e) * \tan (F_c/2))$$

$$\Delta t = ((E_c - \sin E_c) - (E_N - \sin E_N))/\eta$$

If the delta time is greater than iteration tolerance, the current time (t) is updated and the iteration repeated.

The definition of terms is as follows.

- e - orbital eccentricity
- h - argument of the ascending node at the current iteration time
- h_k - argument of the ascending node at the vector time
- $\dot{h}$ - rate of change of the argument of the ascending node
- i - orbital inclination
- g_{CPA} - argument of perigee at the CPA
- g_N - argument of perigee at the nodal crossing time
- $\dot{g}$ - time rate of change of the argument of perigee
- t - current iteration time

t_{VEC} - time of the vector

μ_{CPA} - argument of latitude at CPA

E_C - eccentric anomaly at CPA

E_N - eccentric anomaly at the node

F_C - true anomaly at CPA

$\hat{C}$ - unit vector in the direction of the orbital CPA

$\hat{H}$ - unit vector in the direction of the orbital angular momentum

$\hat{S}$ - unit vector in the direction of the ground site

δ_S - right ascension of the ground site

Δt - delta time

λ - longitude of the ground site

η - orbital mean motion

ϕ - latitude of the ground site

ω_E - rotation rate of the Earth

The following correlates the above symbols with the routine variables.

Math symbol	Internal code symbol
E_C	EAC
F_C	TAC
h	HC
μ_{CPA}	UCPA
δ	SNANG
Δt	DT
$\hat{C}$	CVEC
$\hat{H}$	H
$\hat{S}$	SVEC

c. Landing Site Sunrise/Sunset Computation - The time of local sunrise and sunset is computed by computing the delta time from landing to sunrise or sunset.

The delta longitude from the landing site to sunrise or sunset is computed as follows.

$$\Delta\lambda = \cos^{-1} \{(\cos \alpha - \sin \phi_L \sin \phi_S)/\cos \phi \cos \phi_S\}$$

where

α = the angle from the zenith to the Sun at sunrise or sunset

ϕ_L = latitude of the landing site

ϕ_S = latitude of the Sun

The differences in longitude of the Sun and landing site at the landing time is computed by

$$\Delta\lambda_C = \lambda_S - \lambda$$

where

λ_S = longitude of the Sun

λ = longitude of the landing site

The delta times to sunrise and sunset are computed by

$$\Delta t_{SR} = (\Delta\lambda - \Delta\lambda_c)/\omega_e$$

$$\Delta t_{SS} = (\Delta\lambda + \Delta\lambda_c)/\omega_e$$

where

ω_E = rotation rate of the Earth

The following correlates the above symbols with the routine variables.

Math symbols	Internal code symbol
α	ANGZ
$\Delta\lambda$	DELR
$\Delta\lambda_S$	DELC
λ	LON
λ_S	SLON
ω_E	ROMEGE
ϕ	LAT
ϕ_S	SLAT

5.1.5 Routine Input/Output Variables

The input/output variables are presented in table 5.1-I.

5.1.6 Functional Logic Flow

The trajectory file processing (see method) is required in the generation of the table of ascending nodes and in the generation of the output quantities. The closest point of approach and the landing site sunrise/sunset computations are also required in the generation of the output quantities. The functional logic flow is presented in figure 5.1-2.

5.1.7 Diagnostic and Debug

None.

5.1.8 Special Comments

None.

5.1.9 References

None.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IVEC		2CH	I		IT	IVEC	Vector source flag
LOVEC		Real	I	external	IT	LOVEC	Input invariant elements state vector. Required when IVEC = "IN"
VECFIL		Real	I		IT	VECFIL	Name of invariant elements DRDE. Required when IVEC = "DF"
ISIT		2CH	I		IT	ISIT	Flag indicating source of landing site data "IN" = single input landing site "DF" = landing site DRDE as specified by NAMSIT
NAME		6CH	I		IT	SITNAM	Name of input landing site
GSIT	R, ϕ, λ	Real	I	external	IT	GSIT	Radius, geocentric latitude, and longitude of landing site
LZONE		Intg			IT	LZONE	Time zone of input landing site. Input as number of hours earlier than GMT
SITFIL		6CH	I		IT	SITFIL	Name of landing site DRDE
GETS		Real	I	hr,min, sec	IT	GETS	Start time relative to reference time
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.1-I.- Continued

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GETF		Real	I	hr, min, sec	IT	GETF	Final time relative to reference time
XRNG		Real	I	n. mi.	IT	XRNG	Crossrange constraint
INORB		Intg	I		IT	INORB	Initial orbit counter. Corresponds to orbit counter for GETS
ISTIM		2CH	I		IT		Flag indicating daylight or standard time for local landing time
OUTFLG		2CH	I		IT	OUTFLG	Flag indicating whether or not the output quantities are to be output to a DRDE specified by NAMOUT
NAMOUT		6CH	I		IT		Name of output DRDE
GMTR		Real	I	hr	IT	GMTR	Reference time
CFA		Real	I		IT	CFA	Conversion factor for angles - external to internal
CFD		Real	I		IT	CFD	Conversion factor for distance
CFT		Real	I		IT	CFT	Conversion factor for time
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.1-I.- Continued

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
ROMEGE	ω_E	Real	I	rad/sec	IT	ROMEGE	Rotation rate of Earth
PI		Real	I		IT	PI	π
TWOPI		Real	I		IT	TWOPI	2π
DTWOPI		Real	I		IT	DTWOPI	Double-precision 2π
AMILE		Real	I	feet	IT	AMILE	Feet per nautical mile
AMIN		Real	I	sec	IT	AMIN	Seconds per minute
AHOUR		Real	I	sec	IT	AHOUR	Seconds per hour
ADAY		Real	I	sec	IT	ADAY	Seconds per day
IPRNTL		Intg	I		IT	PROCON(1)	Debug print flag
OLU		Intg	I		IT	PROCON(2)	Output destination flag
DTBIAS		Real	I	sec	IT	PROCON(3)	Bias time to account for the difference between the orbital closest point of approach and the actual landing time.

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.1-I.- Continued

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
ANGZ		Real	I	rad	IT	PROCON(5)	Angle between the zenith and local sunrise/sunset (reference - Supplement to American Ephemeris and Nautical Almanac 90° 50 feet).
ABIAS		Real	I	rad	IT	PROCON(7)	Bias that is applied to nodal range values to account for uncertainty in nodal computations due to flight time.
RTOL		Real	I	sec	IT	PROCON(9)	Iteration tolerance for CPA iteration.
MLINE		Intg	I		IT	PROCON(11)	Maximum number of lines per page
ICR		Intg	I		IT	PROCON(12)	Cartridge reference number for data files
NOBLKS		Intg	I		IT	PROCON(13)	Maximum number of blocks for the output DRDE file
IONA		6CH	O		T	NAME	Landing site name
BLAT		Real	O	external	T	LAT	Landing site geocentric latitude
BLON		Real	O	external	T	LONG	Landing site longitude

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.1-I.- Continued

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IREV IASC		Intg	O		T	ORBIT	Orbit number and indication of whether pass is ascending (A) or descending
XRNGC		Real	O	n. mi.	T	XRNG	Orbital crossrange at closest point of approach. Plus (+) indicates that the site is towards the negative angular momentum vector relative to the orbit
TASR		Real	O	sec	T	TASR	Time of landing after sunrise. Negative time indicates landing time is before sunrise.
TBSS		Real	O	sec	T	TBSS	Time of landing before sunset. Negative time indicates landing time is after sunset.
GETL		Real	O	sec	T	GETL	Ground elapsed time of landing
TLOC		Real	O	sec	T	LLT	Local landing time
SELV		Real	O	external	T	SEA	Sun elevation angle at landing
IONA		6CH	O		F		Landing site name
IREV		Intg	O		F		Orbit number
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.1-I.- Concluded

Routine LOPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IASC		2CH	O		F		ASCII character indicating ascending or descending pass
XRNGC		Real	O	n. mi.	F		Orbital crossrange
TASR		Real	O	sec	F		Time of landing after sunrise
TBSS		Real	O	sec	F		Time of landing before sunset
GETL		Real	O	sec	F		Landing time relative to reference time
TLOC		Real	O	sec	F		Local landing time
SELV		Real	O	deg	F		Sun elevation angle

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

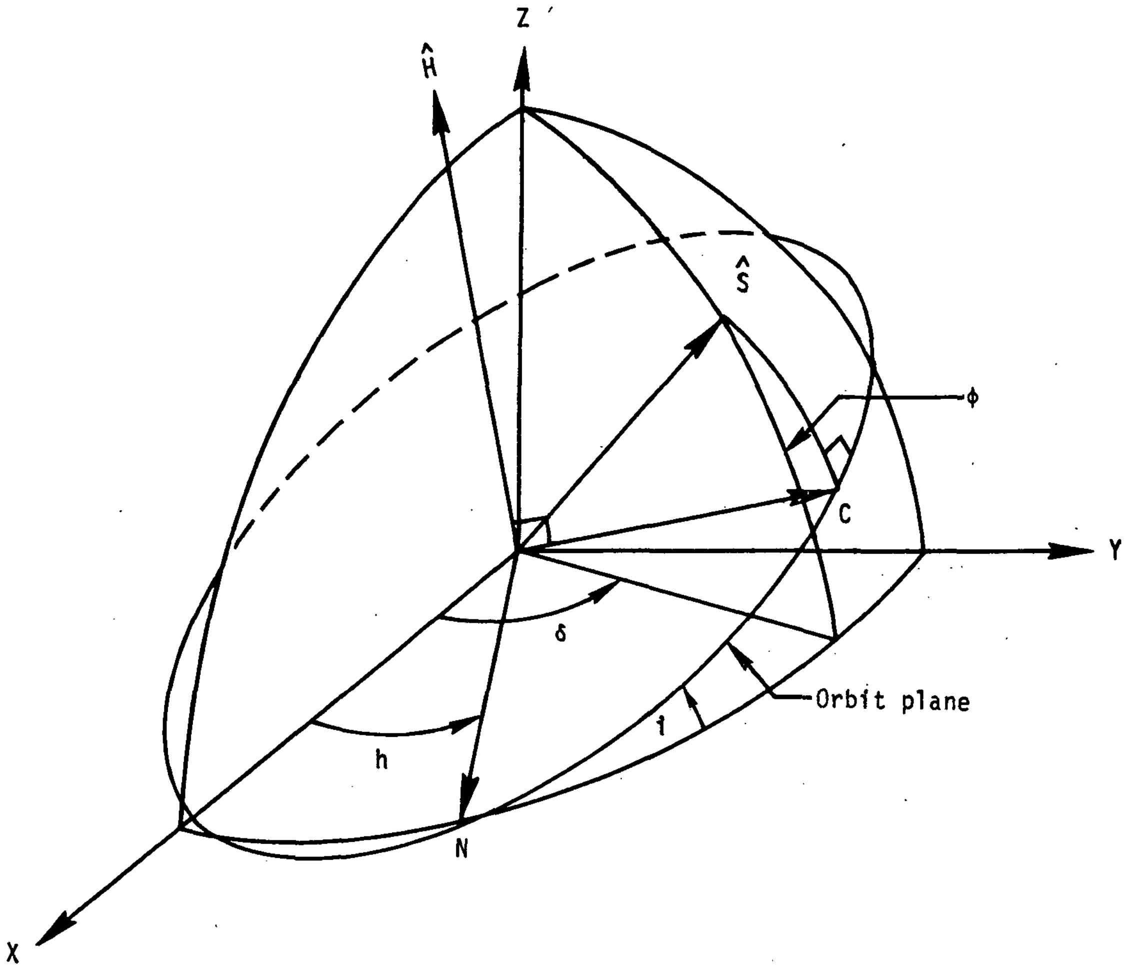
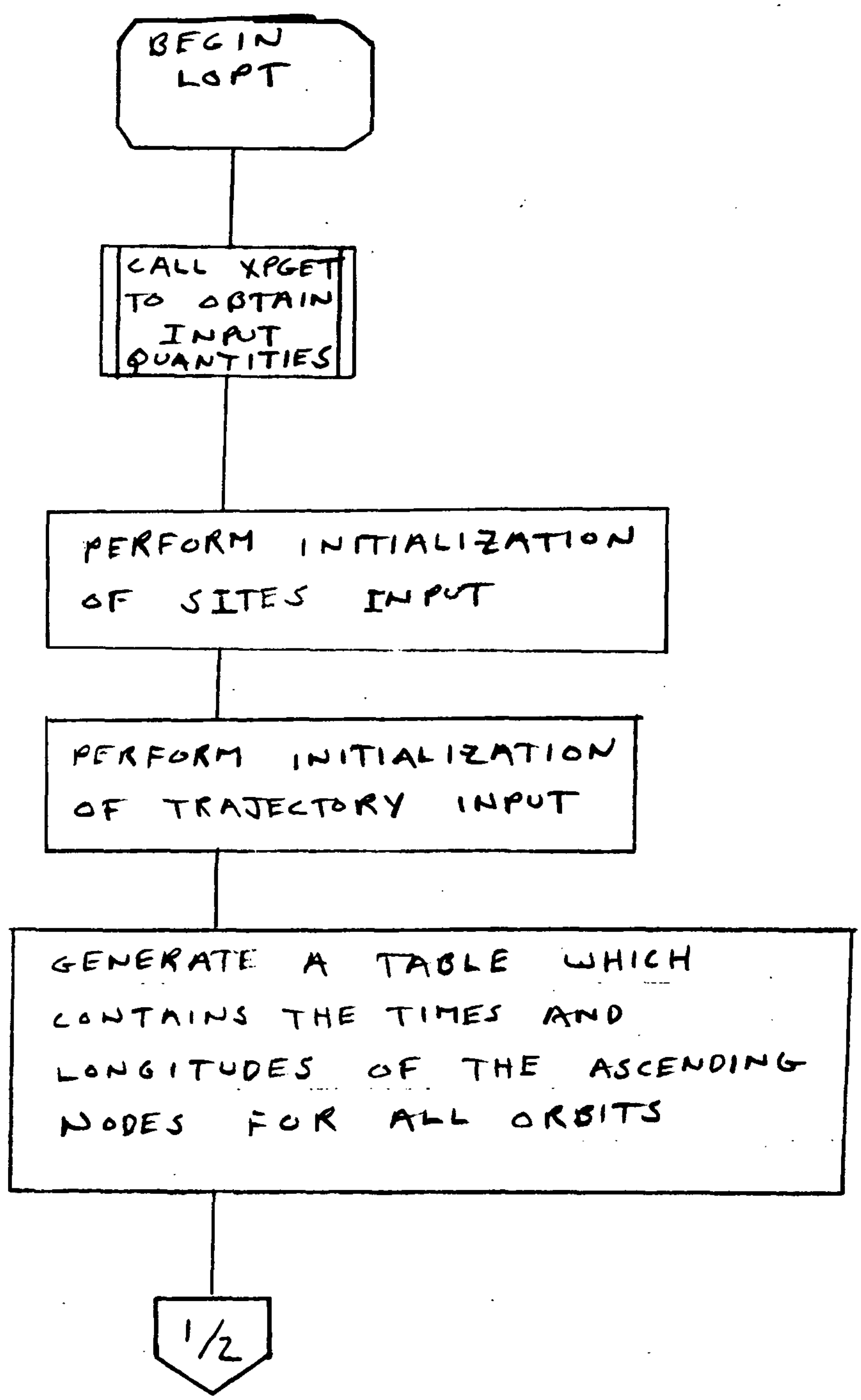


Figure 5.1-1.- Closest point of approach geometry.



Page 1 of 2

Figure 5.1-2.- LOPT functional logic flow.

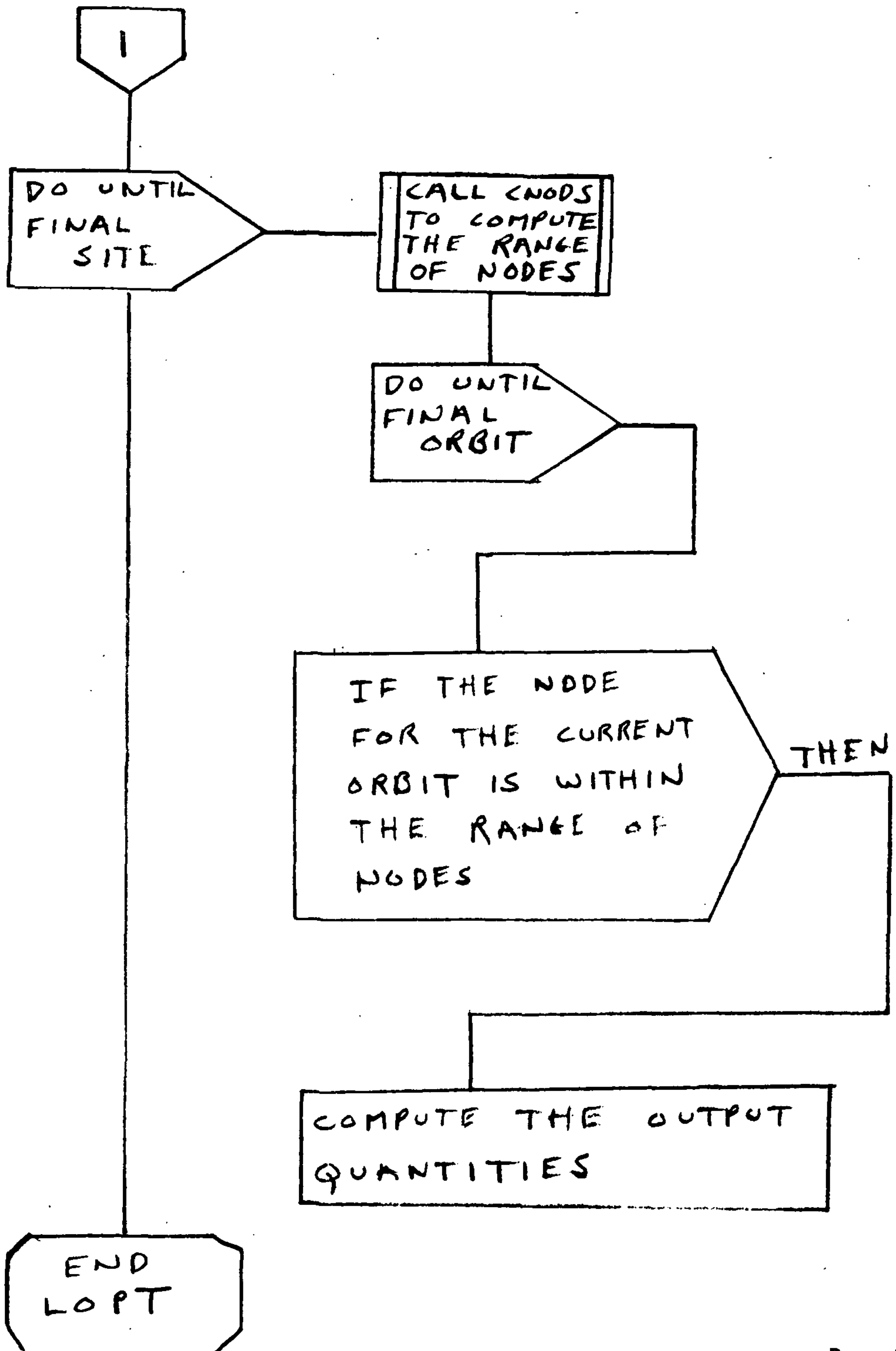


Figure 5.1-2.- Concluded.

5.2 ROUTINE NAME - ADVU

5.2.1 Purpose

This routine finds the argument of latitude for a given time in the range 0 to two pi.

5.2.2 Functional Description

The FORTRAN calling sequence for this routine is

```
CALL ADVU(TIME,ARG)
```

The routine returns the argument of latitude (ARG) for the requested time (TIME).

5.2.3 Assumptions and Limitations

a. Elliptical orbit propagation is assumed.

b. The trajectory vector quantities must come through COMMON.

5.2.4 Method

The mean anomaly at the requ°sted time is computed by

$$M = M_K + \eta * (t - t_V)$$

where

M_K = mean anomaly of the vector

η = mean motion

t_V = time of the vector

The argument of perigee at the requested time is computed by

$$G = G_K + \dot{g} * (t-t_V)$$

where

GK = argument of perigee of the vector

$\dot{g}$ = rate of change of perigee

The mean anomaly is converted to true anomaly (F) by using the function TRUE, and desired argument of latitude (U) is computed by

$$U = G + F$$

where

F = true anomaly

5.2.5 Routine Input/Output Variables

The input/output variables are presented in table 5.2-I.

5.2.6 Functional Logic Flow

The functional logic flow is as follows.

```
1   BEGIN ADVU
2       COMPUTE DELTA-T FROM VECTOR TIME
2       COMPUTE PERIOD
2       MOD DELTA-T BY PERIOD
2       COMPUTE MEAN ANOMALY
2       COMPUTE ARGUMENT OF PERIGEE
2       COMPUTE ARGUMENT OF LATITUDE
1   END ADVU
```

5.2.7 Diagnostics and Debug

None.

5.2.8 Special Comments

None.

5.2.9 References

None.

TABLE 5.2-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine ADVU

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TIME	t	Real	I	sec	A		Requested time
TIMV	t_V	Real	I	sec	C		Vector time
EK		Real	I		C		Eccentricity
GK	G_K	Real	I	rad	C		Argument of perigee
MK	M_K	Real	I	rad	C		Mean anomaly
ETA	n	Real	I	rad/sec	C		Mean motion
GDOT	$\dot{g}$	Real	I	rad/sec	C		Rate of change in perigee
TWOPI		Real	I		C		2π

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.3 ROUTINE NAME - CNODS

5.3.1 Purpose

This routine computes the range of acceptable ascending nodes which will satisfy a given crossrange constraint from a specified latitude and longitude.

5.3.2 Functional Description

The FORTRAN calling sequence for this routine is

```
CALL CNODS(GVEC,ANG,ABIAS,AN1,AN2,AN3,AN4,IFL)
```

The routine computes the ranges of nodes AN1-AN4 and a solution flag IFL.

5.3.3 Assumptions and Limitations

a. Elliptical orbit propagation is assumed.

b. The trajectory vector quantities must come through COMMON.

c. The range of nodes is intended to include all possible nodes. It is not intended to be an exact solution.

5.3.4 Method

A delta longitude is computed from the site to the ascending node by the following.

$$\Delta\lambda^{+} = \sin^{-1}((\sin\phi_S \cos i + \sin\alpha)/\sin i \cos\phi_S) \qquad (1)$$

and

$$\Delta\lambda^{-} = \sin^{-1}((\sin\phi_S \cos i - \sin\alpha)/\sin i \cos\phi_S) \qquad (2)$$

where

ϕ_S = latitude of the site

i = orbital inclination

α = crossrange angle

If the above equations result in an arcsin less than 1 for both signs, then a double range of nodes exist. Then

$\Delta\lambda min(1) = \Delta\lambda^+$

$\Delta\lambda max(1) = \Delta\lambda^-$

$\Delta\lambda min(2) = -\pi$ max (1)

$\Delta\lambda max(2) = -\pi - \Delta\lambda min\ (1)$

If the arcsin term in equation (1) and equation (2) is greater than 1, then no range exists and IFL is set to -1.

If the arcsin term is greater than 1 for one of the equations, and less than 1 for the other then

$\Delta\lambda max(1) = \Delta\lambda^+$ or $\Delta\lambda^-$

$\Delta\lambda min(1) = -\pi - \Delta\lambda min(1)$

For each of the $\Delta\lambda$'s generated above the longitude of the ascending node is given by

$$\lambda N = \lambda S - \Delta\lambda + \Delta T * \omega_E$$

where

λS = longitude of the site

ΔT = orbital flight time

$\Delta\omega_E$ = rotation rate of the Earth

To compute the orbital flight time, the angle from the ascending node to the closest point of approach is computed by

$$\Delta\theta = \cos^{-1} (\cos \phi_S \cos \Delta\lambda / \cos \alpha)$$

The flight time is computed by

$$\Delta T = \Delta\theta / \eta$$

where

η = mean motion

Since the flight time is a circular orbit approximation each ascending node is biased to ensure that the range of nodes include all possible solutions.

The following correlates the above symbols with the routine variables

Math symbols	Internal code symbols
i	IK
$\Delta\lambda min(1)$	DELA1
$\Delta\lambda max(1)$	DELA2
$\Delta\lambda min(2)$	DELA3
$\Delta\lambda max(2)$	DELA4
$\Delta\theta$	TEM
α	ANG
λN	AN1,AN2,AN3,AN4
λS	SLON
ϕ_s	GLAT
η	ETA
ω_E	ROMEG

5.3.5 Routine Input/Output Variables

The input/output variables are presented in table 5.3-I.

5.3.6 Functional Logic Flow

The functional logic flow is presented in figure 5.3-1.

5.3.7 Diagnostic and Debug

None.

5.3.8 Special Comments

None.

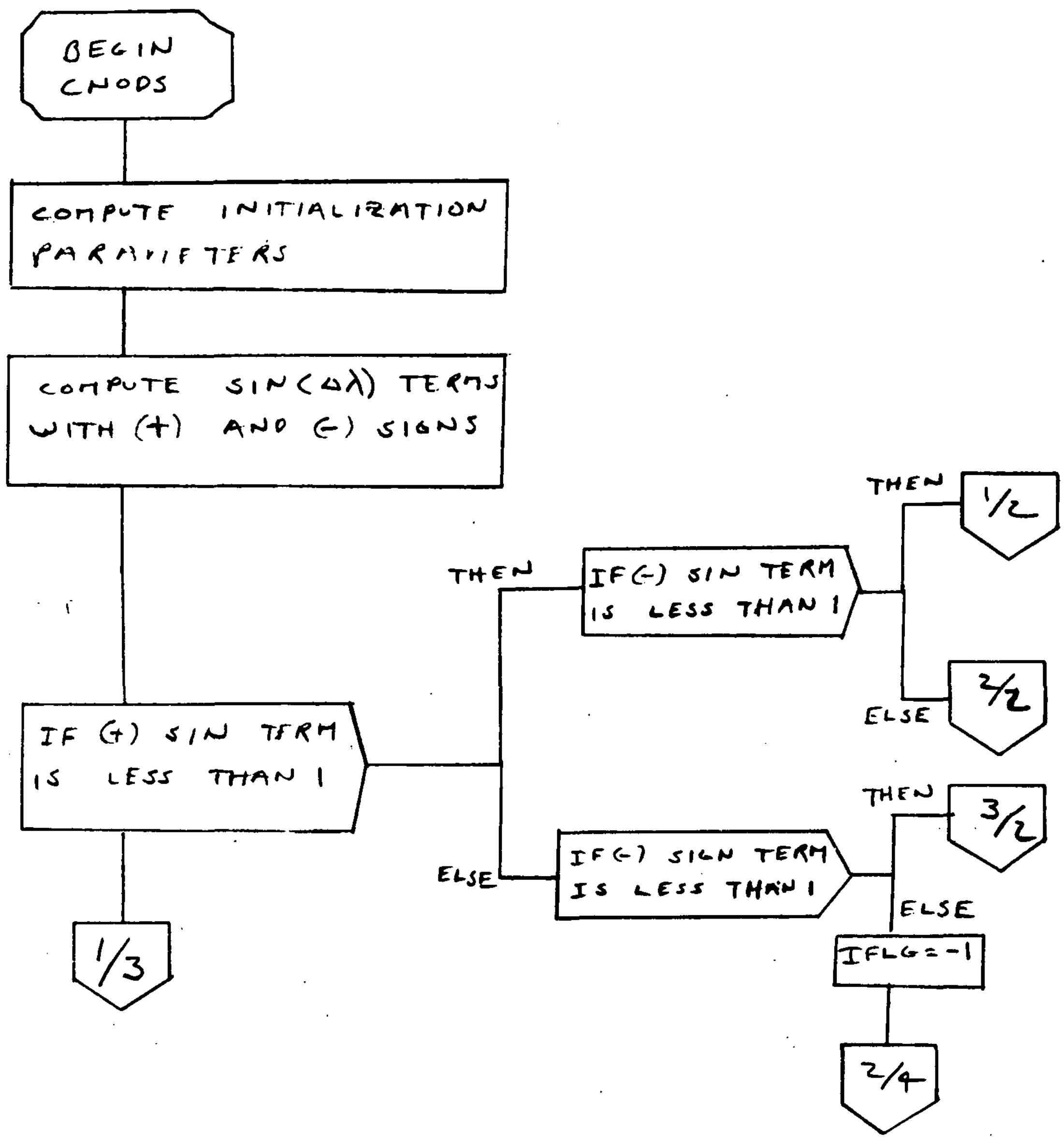
5.3.9 References

None.

TABLE 5.3-I.- ROUTINE INPUT/OUTPUT VARIABLES

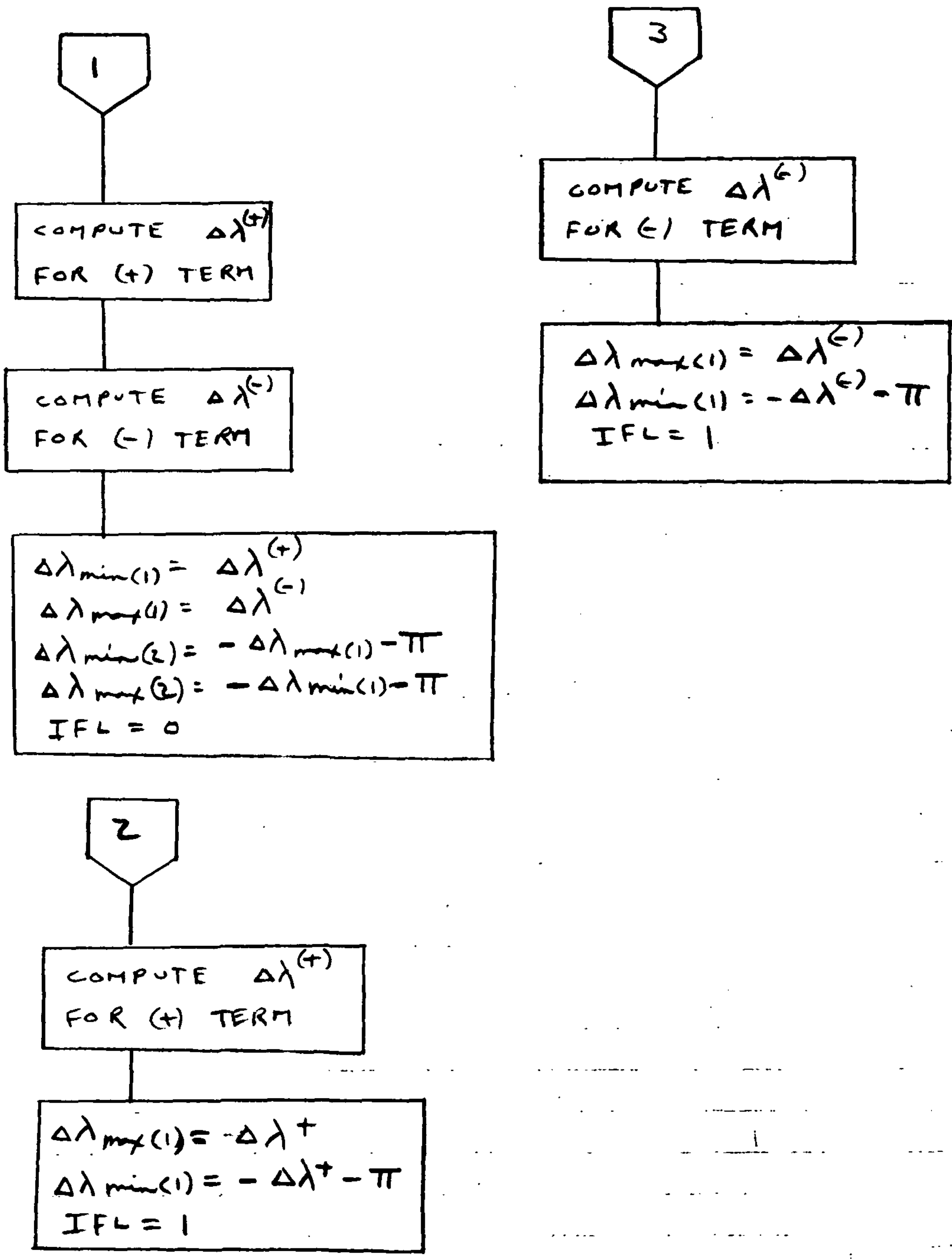
Routine CNODS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GVEC		Real	I	--	A	GVEC	Ground site geographic vector
ANG		Real	I	rad	A	ANG	Crossrange constraint angle
ABIAS		Real	I	rad	A	ABIAS	Bias to be applied to the ascending nodes
IK	i	Real	I	rad	C	IK	Orbital inclination
ETA	η	Real	I	rad/sec	C	ETA	Mean motion
AN1	$\Delta\lambda_{min(1)}$	Real	O	rad	A	AN1	Minimum ascending node, range 1
AN2	$\Delta\lambda_{max(1)}$	Real	O	rad	A	AN2	Maximum ascending node, range 1
AN3	$\Delta\lambda_{min(2)}$	Real	O	rad	A	AN3	Minimum ascending node, range 2
AN4	$\Delta\lambda_{max(2)}$	Real	O	rad	A	AN4	Maximum ascending node, range 2
IFL		Intg	O	rad	A	IFL	Solution range flag -1 = no solution 1 = one range 2 = two ranges
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory



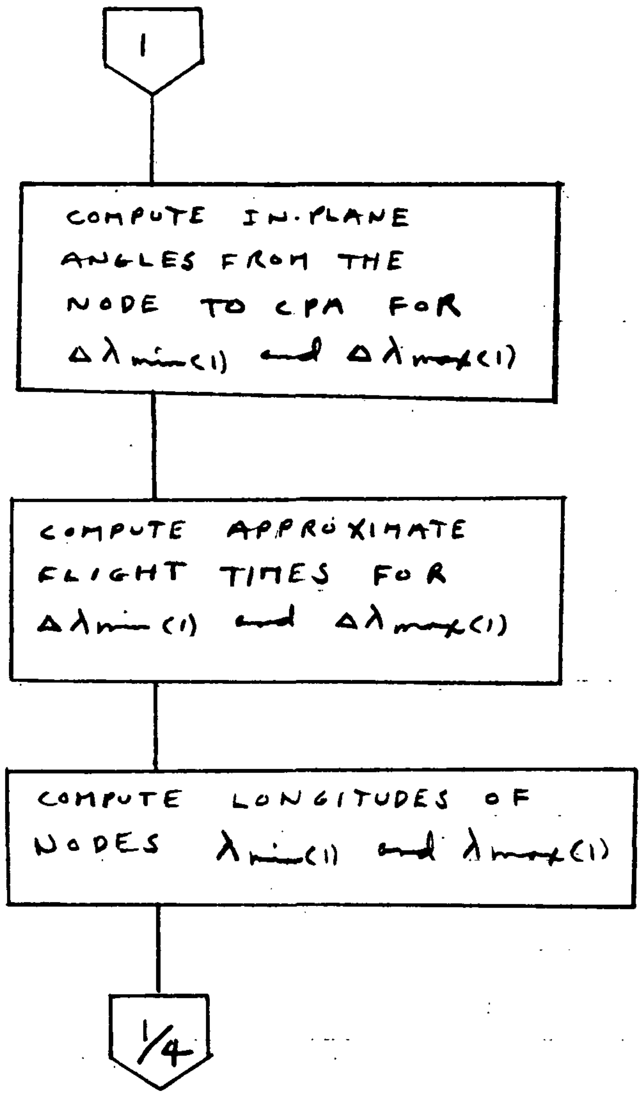
Page 1 of 4

Figure 5.3-1.- CNODS functional logic flow.



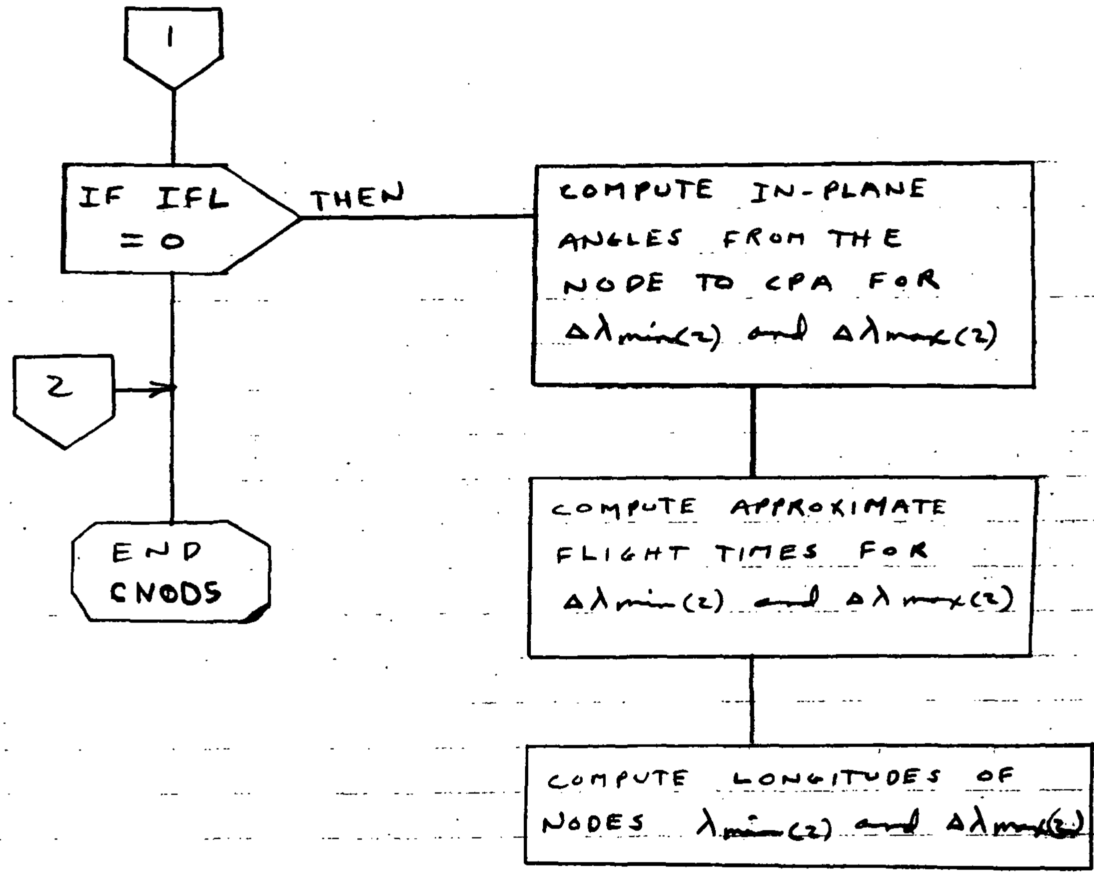
Page 2 of 4

Figure 5.3-1.- Continued.



Page 3 of 4

Figure 5.3-1.- Continued.



Page 4 of 4

Figure 5.3-1.- Concluded.

5.4 ROUTINE NAME - TAU

5.4.1 Purpose

This routine finds the time of arrival at the desired argument of latitude.

5.4.2 Functional Description

The FORTRAN calling sequence for this routine is.

```
CALL TAU(TIMC,UC,UT,TAR,RORB).
```

The routine computes the time of arrival (TAR) at the desired argument of latitude (UT). The TIMC and UC parameters are used to control the orbit where the time of arrival occurs. If the UT argument is greater than the UC parameter, the time of arrival (TAR) will be greater than the current time (TIMC). If UT is less than UC, TAR will be less than TIMC.

5.4.3 Assumptions and Limitations

a. Elliptical orbit propagation is assumed.

b. The vector quantities must be passed through COMMON.

5.4.4 Method

The routine uses an iterative technique to compute the delta time (Δt) from current time (tc). A first guess for this time is obtained by

$$\Delta t = (\mu_T - \mu_C)/(\eta + \dot{g})$$

where

μ_T = desired argument of latitude

μ_C = current argument of latitude

η = mean motion

$\dot{g}$ = rate of change in perigee

The routine then iteratively computes a new Δt until the change in Δt is less than a tolerance. This is done as follows:

$$g_p = g_c + \dot{g}*\Delta t$$

$$F = \mu_T - g_p$$

$$E = 2 \tan^{-1} \left(\frac{1 - e}{1 + e} \tan\left(\frac{F}{2}\right)\right)$$

$$M_p = E - e \sin E$$

$$\Delta t = (M_p - M_c)/\eta$$

where

e = eccentricity

E = eccentric anomaly

F = true anomaly

g_c = argument of perigee at tc

g_p = argument of perigee at estimated time

M_c = mean anomaly at tc

M_p = mean anomaly at estimated time

The orbital radius at the desired argument of latitude is an additional output quantity. It is computed as

$$R = a(1 - e)/(1 + e \cos F)$$

The following correlates the above symbols with the internal code symbols.

Math symbols	Internal code symbols
a	AK
E	EA
e	EK

Math symbols	Internal code symbols
F	TA
g_c	GC
g_p	GP
M_c	MC
M_p	MT
R	RORB
μ_c	UC
μ_T	UT
Δt	DTN

5.4.5 Routine Input/Output Variables

The input/output variables are presented in table 5.4-I.

5.4.6 Functional Logic Flow

The functional logic flow is as follows.

```
1  BEGIN TAU
2      COMPUTE INITIALIZATION PARAMETERS
2      COMPUTE FIRST GUESS DELTA T
2      DO UNTIL DELTA T IS LESS THAN TOLERANCE
3          COMPUTE NEW ARGUMENT OF PERIGEE
3          COMPUTE NEW TRUE ANOMALY
3          COMPUTE NEW ECCENTRIC ANOMALY
3          COMPUTE NEW MEAN ANOMALY
3          COMPUTE DELTA MEAN ANOMALY
```

```
3          IF THE SIGN OF THE DELTA MEAN ANOMALY IS
           NOT THE SAME AS THE SIGN OF (UT-UC) THEN

4             ADD OR SUBTRACT TWOPI

3          COMPUTE NEW DELTA T

2       END DO

2       COMPUTE RADIUS

1    END TAU
```

5.4.7 Diagnostic and Debug

None.

5.4.8 Special Comments

None.

5.4.9 References

None.

TABLE 5.4-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine TAU

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TIMC		Real	I	sec	A	TIMC	Current time
UC	μ_C	Real	I	rad	A	UC	Current argument of latitude
UT	μ_T	Real	I	rad	A	UT	Desired argument of latitude
TIMV		Real	I	sec	C	TIMV	Vector time
AK	a	Real	I	feet	C	AK	Semimajor axis
EK	e	Real	I		C	EK	Eccentricity
GK		Real	I	rad	C	GK	Argument of perigee
MK		Real	I	rad	C	MK	Mean anomaly
ETA	n	Real	I	rad/sec	C	ETA	Mean motion
GDOT	$\dot{g}$	Real	I	rad/sec	C	GDOT	Rate of change of argument of perigee
RORB	R	Real	O	feet	A	RORB	Radius of vehicle
TAR		Real	O	sec	A	TAR	Time of arrival at desired argument of latitude
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

5.5 ROUTINE NAME - RVECF

5.5.1 Purpose

This routine performs trajectory file processing for trajectory DRDE files that were produced by the INVAR processor.

5.5.2 Functional Description

The FORTRAN calling sequence for this routine is

```
CALL RVECF (IFLG,NAME,IERR,TSWCH)
```

The input parameter IFLG is used to control the function to be performed (OPEN,READ, or CLOSE) on the file specified by NAME. The routine returns the error flag (IERR) and the vector switch time (TSWCH). It also places the updated set of invariant elements in the COMMON block.

5.5.3 Assumptions and Limitations

a. The disk file must be in the format as produced by the INVAR processor.

b. The trajectory parameters are placed in COMMON.

5.5.4 Method

This routine uses the RTE operating system FMP routines OPEN, READF, and CLOSE to perform the desired function.

5.5.5 Routine Input/Output Variables

The input/output variables are presented in table 5.5-I.

5.5.6 Functional Logic Flow

The functional logic flow is presented in figure 5.5-1.

5.5.7 Diagnostics and Debug

None.

5.5.8 Special Comments

None.

5.5.9 References

Section 5.0 INVAR processor in this volume.

TABLE 5.5-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine RVECF

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IFLG		Intg	I		A		Flag that determines whether the open, read, or close function is desired.
NAME		6CH	I		A		Name of the disk file.
TVEC		Real	O	sec	C		Time of vector.
AK		Real	O	feet	C		Semimajor axis.
EK		Real	O	—	C		Eccentricity.
IK		Real	O	rad	C		Inclination.
GK		Real	O	rad	C		Argument of perigee.
HK		Real	O	rad	C		Argument of ascending node.
MK		Real	O	rad	C		Mean anomaly.
ETA		Real	O	rad/sec	C		Mean motion.
GDOT		Real	O	rad/sec	C		Rate of change of argument of perigee.

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.5-I.- Concluded

Routine RVECF

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
HDOT		Real	O	rad/sec	C		Rate of change of argument of ascending node.
IERR		Intg	O		A		FMGR error return code. Negative indicates error condition.
TSWCH		Real	O	sec	A		Vector switch time.

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

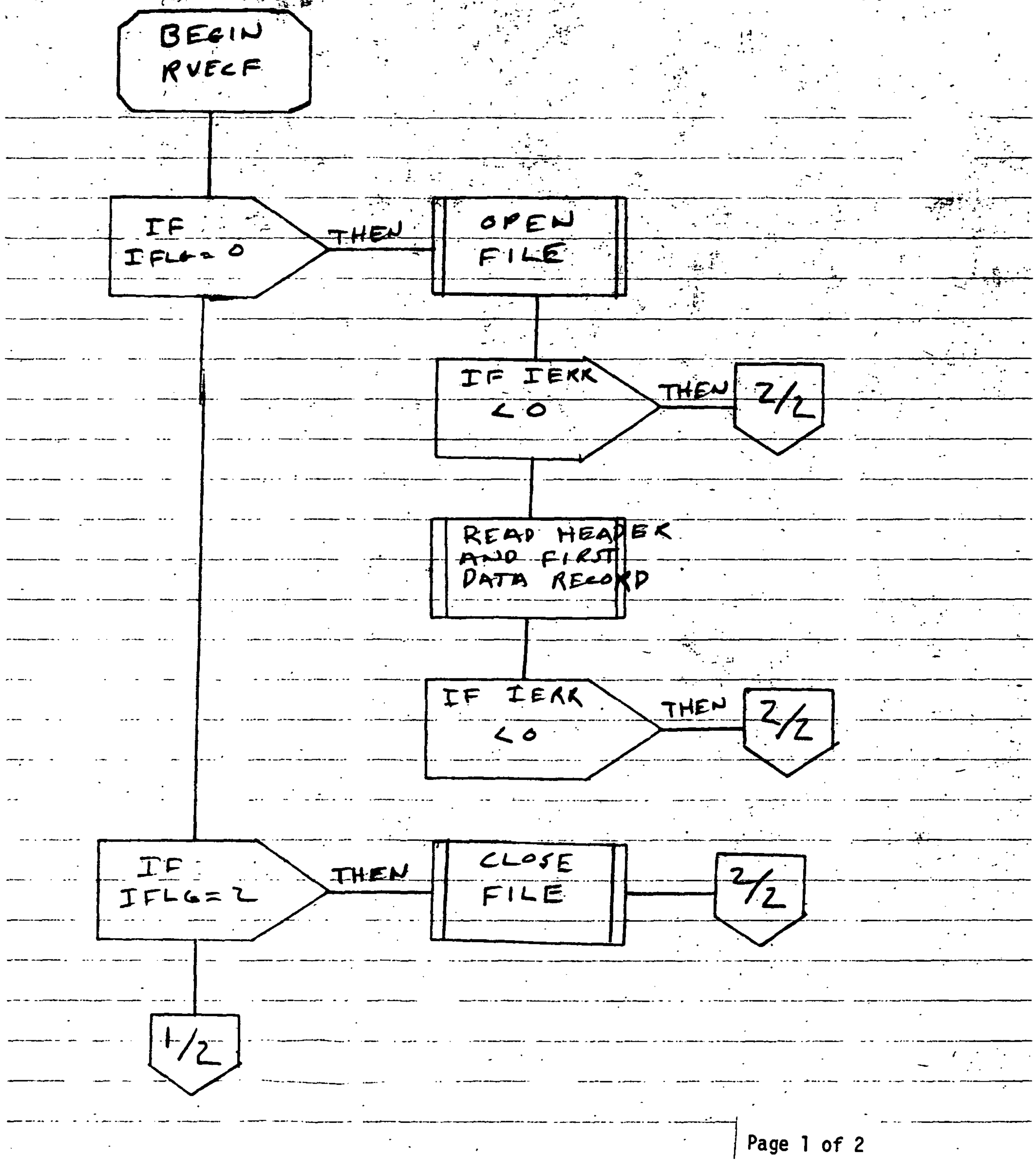
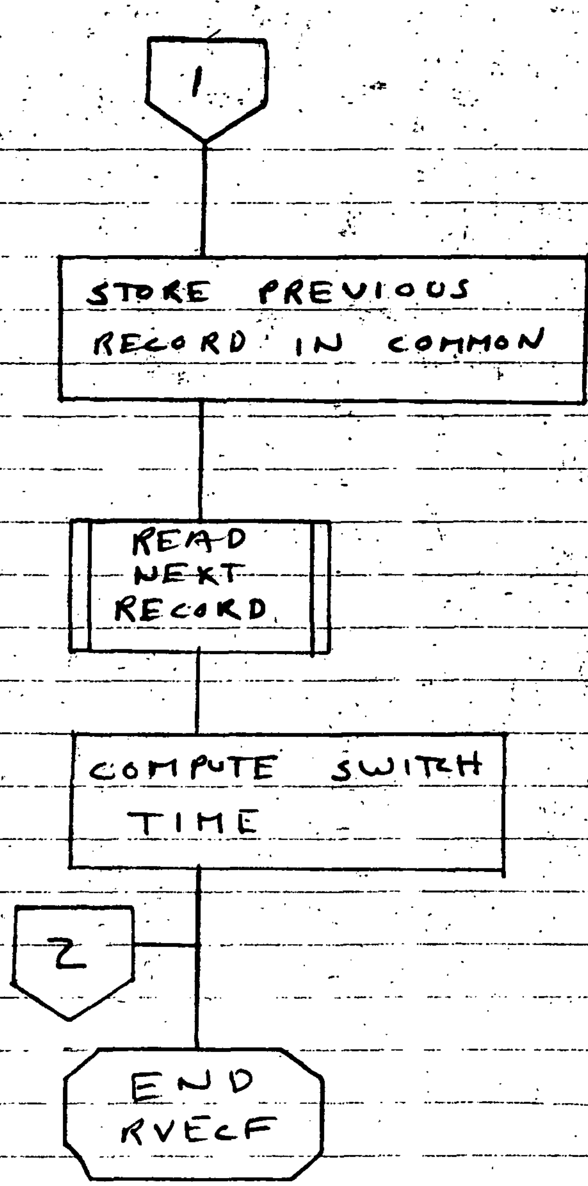


Figure 5.5-1.- RVECF functional logic flow.



Page 2 of 2

Figure 5.5-1.- Concluded.

LOAD STATE VECTOR PROCESSOR (LSV)

1.0 PURPOSE

The load state vector utility processor is known as LSV and provides the FDS-1 user with an easy and convenient means for loading, modifying, displaying, and storing position/velocity state vectors in any one of the 15 sets supported by FDS-1. (See table 7.3-VI of JSC IN 78-FM-60, volume I.)

2.0 FUNCTIONAL DESCRIPTION

The load state vector is composed of a main routine LSV and a state vector prompt option subroutine SVPRO. Through the interface table parameters SVIN and SVOUT, the FDS-1 user specifies the names of the AWA data elements that contain the input and output position/velocity state vectors. During the operation of the LSV processor, it first prompts the FDS-1 user for the mode option. It then prompts the user for the appropriate vector components and affords the user on opportunity to verify inputs and change or correct them.

There are five LSV mode options, INPUT, LIST, EDIT, STOP, and EXIT. The mode prompt is the first prompt issued to the user, and by response to this prompt, the user controls the operational flow of the LSV processor. The mode prompt is of the following form:

LSV MODE
(INPUT, LIST, EDIT, STOP, EXIT) : (USER RESPONSE)

The user responds with one of the mode options listed within the parenthesis. The flow of processor execution then proceeds as described in one of the following mode operation paragraphs.

NOTE: On initial entry into the LSV processor, a user response of LIST, EDIT, or EXIT to the mode prompt will cause the processor to request (from the FDS manager) the input state (SVIN) specified in the interface table; this input state is then stored in the processor's internal state vector working buffer.

INPUT MODE:

When the user responds to the mode prompt with the INPUT option, the LSV processor zeros the internal state vector working buffer and proceeds to prompt the user for the required state vector component inputs. The order of prompts are as follows:

a. The processor first prompts for the reference axis,
REFAX : (User responds with one of the five reference axis systems (JSC IN 78-FM-60, vol. I, table 7.3-VI).)

b. The processor next prompts for the element set,
ELSET : (User responds with one of the fifteen element sets (JSC IN 78-FM-60, vol. I, table 7.3-VI).)

Once a valid reference axis and a valid element set are entered, the LSV processor tests to see if the element set is contained within the given reference axis. If the element set entered is not contained within the given reference axis, an error message

LSV INVALID ELEMENT SET XXXXXX FOR REFERENCE AXIS XXXXXX

is displayed, and the processor prompts again for the element set. The processor will remain in this loop until a suggested combination of REFAX and ELSET is entered or the user responds with % (a return to Executive). If the element set entered is contained within the previously entered reference axis, the LSV processor prompts the user for the appropriate state vector components. The element sets and the prompt mnemonics are defined in volume I, table 7.3-VI. The form of the element prompts a function of the element set mnemonics specified by the user via ELSET : prompt. In responding to prompts for the state vector components, the user may enter one or more components per line separated by commas even though the prompt identifies only a single component. Such additional entries are applied to subsequent components of the state vector, and the processor will then not individually prompt for the corresponding components, rather prompting will resume with the next component for which no entry has yet been supplied. If an error occurs in entering input data, an appropriate diagnostic is displayed, and the user is reprompted with the last prompt component. Any excess data entered are ignored. Consider the following example:

```
REFAX : "M50"
ELSET : "CARXYZ"
```

where : REFAX : is the processor prompt for reference axis. The user response is M50 indicating that the vector to be input is referenced to the mean Equator and equinox of 1950 axis system.

ELSET : is the processor prompt for element set. The user response of CARXYZ indicates that the vector to be input is in Cartesian form $(x,y,z,\dot{x},\dot{y},\dot{z})$.

The next prompt from the processor will be for the first element of the standard position/velocity state vector, which is time (T=:). (See JSC IN 78-FM-60, volume I, figure 7.3-2.) Subsequent prompting will be for the appropriate vector component. Two of the ways in which the user might respond are shown as case A and case B as follows:

CASE A

```
T    = :10.5
X    = : 1.5
Y    = : 1.7
Z    = : 1.9
DX   = :.52,-.32,.44
MASS = :150000.
```

CASE B

```
T  = :10.5,1.5,1.7,1.9
DX = :.52
DY = :-.32
DZ = :.44,150000.
```

The Hollerith characters and the colon (:) are the prompts issued by the LSV processor (e.g.,T = :), and the real numbers are the user's response. In case A, the user lets the processor individually prompt for the components T, X, Y, and Z, then enters the three velocity components $(\dot{x},\dot{y},\dot{z})$ on one line in response to the DX prompt, and then the vehicle mass in response to the MASS prompt.

In case B, the user chose to enter the time and the three position components on one line in response to the T prompt. The processor then prompts for the next undefined component, which is DX, and the user chooses to let the processor on the same line with the DZ component. When the user has entered an input for all elements in the standard position/velocity state vector of the specified element set, the processor computes the JKNN code word using REFAX and ELSET inputs and stores it in SVIN(14). The processor then issues a MODE prompt, thus enabling the user to change or list the values of the state vector or terminate the processing.

LIST MODE:

When the user responds to the mode prompt with the LIST option, the LSV processor displays the contents of its internal state vector working buffer. The format of this display is the same as that of the State Vector Print processor (PSV). (For documentation of this format see the PSV processor documentation in book 3 of this volume.) This mode of operation allows the user to verify the elements of the state vector and list them for future reference before they are stored into the user AWA. After display, the processor will reissue the MODE prompt, thus enabling the user to change the values of the state vector or terminate the processing.

EDIT MODE:

When the user responds to the MODE prompt with the EDIT option, the LSV processor prompts the user for a mnemonic as follows:

```
EDIT: (user response)
```

In this mode, the user response is expected to be the mnemonic for the state vector element that is to be updated. If the user response is recognized as a valid mnemonic for the input vector, the processor will then prompt the user for the input of that element. For instance, consider the following example:

```
LSV MODE
(INPUT, LIST, EDIT, STOP, EXIT): "EDIT"
EDIT :"T"
T   =:25.8
EDIT :"R"
R   =:175.45
EDIT :b̸(CR)
LSV MODE
(INPUT, LIST, EDIT, STOP, EXIT):(next response)
```

In this example, the user updates the time (T) and the radius magnitude (R) elements of the state vector that is currently stored in the internal working buffer. Then the user terminates the EDIT mode by responding with a blank (b̸) to the EDIT mode (EDIT:) prompt. In the EDIT mode, the user may also change the reference axis (REFAX) and/or element set (ELSET) designations for the state vector. However, if the ELSET designation is changed it must be remembered that the current values of the state vector elements may no longer be consistent with the new definition of the state vector elements. There is no error message to warn the user of the problem if the REFAX/ELSET combination is valid.

Notes: (1) If the user responds to the EDIT mode prompt (EDIT:) with a mnemonic that is not valid for the current ELSET designation, the processor will display an appropriate error message and reissue the EDIT mode prompt (EDIT:).

(2) In the EDIT mode, the user may only update one state vector element at a time. Multiple entries on a single line will cause an error message display, and all entries beyond the first will be ignored.

(3) To exit from the EDIT mode, the user must respond to the EDIT mode prompt (EDIT:) with a blank(s) (b̸) and a carriage return. After exiting from the EDIT mode, the processor will reissue the MODE prompt.

STOP MODE:

When the user responds to the MODE prompt with the STOP option, the LSV processor terminates execution without transferring the contents of the internal state vector working buffer into the output state (SVOUT). This mode of operation allows the user to terminate execution of the processor without affecting the contents of the AWA. The user may also abort processor execution at any time without affecting the AWA contents by responding with a % to any processor prompt.

EXIT MODE:

When the user responds to the mode prompt with the EXIT option, the LSV processor transfers the contents of the internal state vector working buffer into the output state vector buffer SVOUT and stores the SVOUT buffer in the user AWA. The LSV processor terminates execution with a normal return to FDS.

3.0 ASSUMPTIONS AND LIMITATIONS

a. The LSV processor uses either predefined interface state vector buffer data or generates its own data in INPUT mode.

b. The state vector used, generated and possibly changed is not necessarily stored into the user AWA but ignored completely via STOP mode response or % response.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the LSV interface table parameters is provided in table 4-I.

b. Interface table data array definitions - A definition of the input/output data arrays appearing in the LSV interface table is provided in table 4-II.

c. Interface table data file definitions - None.

d. Processor solicited (prompted) inputs - The LSV processor mode prompt, the EDIT mode prompt, and the position/velocity state vector parameter keywords are provided in table 4-III.

e. Processor displays and display parameter definition tables - Only one display is generated by the LSV processor; the optional display is controlled by the user response of LIST to the mode prompt. The LSV processor displays the state vector parameters utilizing the same display as the PSV processor. The format of this display is described in the PSV processor documentation.

f. Processor message table - Table 4-IV contains the processor messages that are displayed to the user whenever an invalid entry is made. Error messages are also displayed for unsupported reference axis and element sets and the reference axis/element set combination. None of the messages are fatal. The LSV processor will continue to execute until a user response of STOP or EXIT to the mode prompt or a user response of % to any prompt.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VI.

TABLE 4-I.- PROCESSOR INTERFACE TABLE.

PROCESSOR LSV

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
SVIN	AWA	Real	I	30	15	—	Standard format 15; element input position/velocity state vector.
SVOUT	AWA	Real	O	30	15	--	Standard format 15; element output position/velocity state vector.

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR LSV

Array name	Index location	Default value	Definition
SVIN	(1) . . . (15)		Input position/velocity state vector, standard format; see JSC IN 78-FM-60, volume I, figure 7.3-2 for definition of state vector array locations.
SVOUT	(1) . . . (15)		Output position/velocity state vector, standard format; see JSC IN 78-FM-60, volume I, figure 7.3-2 for definitions of state vector array locations.

TABLE 4-III.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR LSV

Prompt	Meaning	Valid responses
LSV mode (INPUT, LIST, EDIT, STOP, EXIT):	LSV mode prompt	"INPUT" - Input mode "LIST" - List mode "EDIT" - Edit mode "STOP" - Stop mode; terminate without transferring the contents to AWA. "EXIT" - Exit mode; transfer the state vector contents to AWA and terminate execution. % - Return to Executive.
REFAX:	Reference axis prompt	"M50" - Mean Equator and equinox of 1950 (inertial) "MEE" - Mean Equator and equinox of epoch (inertial) "TEE" - True Equator and equinox of epoch (inertial) "TEG" - True Equator and Greenwich meridian of epoch (inertial) "EFE" - Earth-fixed equatorial of epoch (rotational) % - Return to executive
ELSET:	Element set prompt	"CARXYZ - Cartesian "POLRDC" - Polar-radius, declination, right ascension "POLRGC" - Polar-radius, geocentric latitude, longitude "POLRGD" - Polar-radius, geodetic latitude, longitude "POLHGC" - Polar-altitude, geocentric latitude, longitude "POLHGD" - Polar-altitude, geodetic latitude, longitude "GEODED" - Polar-altitude, geodetic -ϕ, γ, Ψ "AEIMA" - Orbital-mean anomaly "AEITA" - Orbital-true anomaly "AEITP" - Orbital-perigee time "HAPRWP" - Apogee/perigee altitude - true anomaly "HAPLDC" - Apogee/perigee altitude - longitude/declination "RRDOT" - Radius, radial velocity, argument latitude "INVRNT" - Invariant elements "KSELMT" - KS elements % - Return to Executive

TABLE 4-III.- Concluded

PROCESSOR LSV

Prompt	Meaning	Valid responses
EDIT:	Edit prompt for EDIT mode. The user enters a mnemonic and waits to be prompted for that mnemonic.	"REFAX" - Reference axis prompt request. "ELSET" - Element set prompt request. "ELEMENT MNEMONIC" - Element mnemonic prompt request, element mnemonics are shown in table 4-V and definition of elements are described in JSC IN 78-FM-60, volume I, table 7.3-VI. b̸ - Exit from EDIT mode and reissue the mode prompt % - Return to Executive
T : MASS :	Prompt for user-requested element value in the EDIT mode.	FLOATING POINT VALUE for the element mnemonic selected via EDIT: prompt. See table 4-V for complete list of mnemonics.

TABLE 4-IV.- PROCESSOR MESSAGE TABLE

PROCESSOR LSV

MSG no.	Message ID block	Message text block and explanation
1	*LSV*	ILLEGAL MODE PROMPT, TRY AGAIN Meaning: The user has entered a bad mnemonic to a mode prompt. Severity: None, the prompt is issued again. Action required by user: Correct input via a mode reprompt.
2	*LSV*	UNSUPPORTED REFAX/ELSET, JKNN=XXXXX Meaning: Input state vector specified in the interface table and received from the FDS manager had unsupported vector type code. Severity: None, see user action. Action required by user: The user should respond with an EDIT or INPUT mode to the mode prompt to correct the state vector before storing it in the user AWA.
3	*LSV*	INVALID REFAX/ELSET COMBINATION, JKNN=XXXXX Meaning: Input state vector specified in the interface table and received from the FDS manager had invalid reference axis/element set combination. Severity: None, see user action. Action required by user: The user should respond with an EDIT or INPUT mode to the mode prompt to correct the state vector before storing it in the user AWA.
4	*LSV*	ILLEGAL REFERENCE AXIS PROMPT, TRY AGAIN Meaning: The user responded with an illegal mnemonic to the reference axis REFAX prompt. Severity: None, the prompt is issued again. Action required by user: Valid reference axis mnemonic input to the REFAX prompt.
5	*LSV*	ILLEGAL ELEMENT SET PROMPT, TRY AGAIN Meaning: The user responded with an illegal mnemonic to the element set ELSET prompt. Severity: None, the prompt is issued again. Action required by user: Valid element set mnemonic response to the ELSET prompt.

TABLE 4-IV.- Continued

PROCESSOR LSV

MSG no.	Message ID block	Message text block and explanation
6	*LSV*	INVALID ELEMENT SET XXXXXX FOR REFERENCE AXIS XXXX Meaning: The user responded with an element set mnemonic that is not supported for the specified reference axis. Severity: None, see user action. Action required by user: (1) INPUT mode Correct input via ELSET prompt (2) EDIT mode The user is prompted with an EDIT: The user can respond with ELSET solicited prompt and enter correct element set mnemonic. Or, the user can respond with REFAX solicited prompt and change reference axis.
7	*LSV*	ILLEGAL VALUE FOR ELEMENT XXXX Meaning: The user response to an element value prompt was illegal; not real value. Severity: None, the prompt is issued again. Action required by user: Correct input via same prompt.
8	*LSV*	INVALID ELEMENT NAME FOR XXXXXX Meaning: The user responded with an element set mnemonic that is not supported in the FDS-1. Severity: None, the prompt is issued again. Action required by user: Correct response to the EDIT prompt.
9	*LSV*	INVALID FLOATING POINT Meaning: The user response to an element prompt was invalid. Severity: None, the prompt is issued again. Action required by user: Correct floating point input to the same prompt.
10	*LSV*	INVALID ENTRY TO *EDIT* PROMPT Meaning: The user responded with an invalid mnemonic to the EDIT prompt. Severity: None, the prompt is issued again. Action required by user: Correct response to the EDIT prompt.

TABLE 4-IV.- Concluded

PROCESSOR LSV

MSG no.	Message ID block	Message text block and explanation
11	*SPSV*	UNSUPPORTED REFAX/ELSET COMBINATION, JKNN=XXXXX Meaning: The user requested to list the state vector data and SPSV subroutine found that vector type code had unsupported reference axis/element set combination. Severity: None, see user action. Action required by user: The user should respond with an EDIT or INPUT mode to the mode prompt to correct the state vector before storing it in the user AWA. Note: See PSV processor documentation in this volume for further information on *SPSV* messages.

TABLE 4-V.- ELEMENT SET TABLE

ELEMENT SET MNEMONICS

STATE VECTOR	CARXYZ	POLRDC	POLRGC	POLRGD	POLHGC	POLHGD	GEODET	AEIMA	AEITA	AEITP	HAPRWP	HAPLDC	RRDOT	INVRNT	KSELMT
1	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
2	X	R	R	R	H	H	H	A	A	A	HA	HA	R	AK	A1
3	Y	DECL	GCLT	GDLT	GCLT	GDLT	GDLT	ECC	ECC	ECC	HP	HP	RDOT	ECCK	A2
4	Z	RASC	LONG	LONG	LONG	LONG	LONG	INC	INC	INC	INC	INC	A	INCK	A3
5	XDOT	V	V	V	V	V	V	RAN	RAN	RAN	RAN	LONG	U	RNK	A4
6	YDOT	GAM	GAM	GAM	GAM	GAM	GAMD	WP	WP	WP	WP	DECL	INC	WPK	B1
7	ZDOT	AZ	AZ	AZ	AZ	AZ	AZD	MA	TA	TP	TA	TA	RAN	MAK	B2
8														RNRK	B3
9														WPRK	B4
10														MARK	WO
11	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD	CD
12	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS	ACS
13	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS	MASS
14															
15															

Sample LSV execution:

```
32>RUN,FDS,32,T
*** XCO1 LU 32 SIGNED ON TO FDS!
 %:INTE,LSV,,LSV1
  SVIN= :13R1.0,2008.0,15.0
  SVOUT@:SVT
   :EXIT
 %:MANU
 $:LSV,LSV1
LSV MODE
   (INPUT, LIST, EDIT, STOP, EXIT):"LIST"
 *LSV*  PROCESSOR   STATE        TEE                          AEIMA
 TIME= 1.000000E+00 GMT     MASS= 1.000000E+00
   CD= 1.000000E+00         AREA= 1.000000E+00
    A= 1.000000E+00          ECC= 1.000000E+00          INC= 1.000000E+00
  RAN= 1.000000E+00           NP= 1.000000E+00           MA= 1.000000E+00
LSV MODE
   (INPUT, LIST, EDIT, STOP, EXIT):"STOP"
 $:LSV,LSV1
LSV MODE
   (INPUT, LIST, EDIT, STOP, EXIT):"ED"
  EDIT:"ELSET"
  ELSET :"CARXYZ"
  EDIT:"ELSET"
  ELSET :"POLRDC"
  EDIT:"REFAX"
  REFAX :"M50"
  EDIT:"T"
  T    = :15.0
  EDIT:"XDOT"
*LSV*  INVALID ELEMENT NAME FOR POLRDC
  EDIT:"GAM"
  GAM = :6.0
  EDIT:"DECL"
  DECL= :3..4.
*** XPO2   DATE AREA OVERFLOW
  DECL= :3.
  EDIT:
LSV MODE
   (INPUT, LIST, EDIT, STOP, EXIT):"LI"
 *LSV*  PROCESSOR   STATE        M50                          POLRDC
 TIME= 1.500000E+01 GAT     MASS= 1.000000E+00
   CD= 1.000000E+00         AREA= 1.000000E+00
    R= 1.000000E+00         DECL= 3.000000E+00         RASC= 1.000000E+00
    V= 1.000000E+00          GAM= 6.000000E+00           AZ= 1.000000E+00
LSV MODE
 (INPUT, LIST, EDIT, STOP, EXIT):"EXIT"

 $:
```

TABLE 4-VI.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR LSV

Processor name	Processor abstract prompt (maximum 256 characters)
LSV	The LSV position/velocity load state vector provides the FDS user with an easy and convenient means for loading and storing position/velocity state vectors in any one of the many coordinate sets supported by FDS (section 7.3 of JSC IN 78-FM-60, volume I). It also provides the capability to edit position/velocity state vector parameters already stored

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
SVIN	Standard format 15-element input position/velocity state vector.
SVOUT	Standard format 15-element ouput position/velocity state vector.

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - MAIN PROGRAM LSV

5.1.1 Purpose

LSV is the Load State Vector (LSV) utility processor. It provides the FDS-1 user with an easy and convenient means for loading, modifying, displaying, and storing position/velocity state vectors in any one of the 15 sets supported by FDS-1. (See table 1.2-VI of reference 1.)

5.1.2 Functional Description

There are seven parts for the LSV processor; the MODE prompt, initial pass and LIST, EDIT, or EXIT mode, and the five option modes (INPUT, LIST, EDIT, STOP, EXIT).

MODE prompt - The MODE prompt is the first prompt issued to the user and, by response to the first and subsequent MODE prompts, the user controls the operational flow of the LSV processor. The LSV main routine calls subroutine SVPRO to execute and analyze a MODE prompt (see SVPRO documentation, section 5.2). Initial pass and LIST, EDIT, EXIT mode - an initial user response to the MODE prompt of LIST, EDIT, or EXIT causes the processor to request (from the FDS manager) the input position/velocity state vector SVIN specified in the interface table. This input state vector is then stored in the internal position/velocity state vector working buffer SVIN, and the processor sets the initial pass flag. The LSV processor then deciphers the SVIN(14) or JKNN vector-type code into the reference axis index (JRA) and element set index (I15) as follows:

```
JKNN = IFIX(SVIN(14) + .01 tolerance)

JRA  = JKNN/1000 + 1

I15  = JKNN-(JRA-1)*1000
```

The LSV processor then proceeds to check the limits of reference axis index JRA, the limits of element set index I15, and validity of reference axis/element set combination. If JKNN vector code is valid, the flow of the processor execution proceeds to the LIST, EDIT, or EXIT mode section. If reference axis and/or element set is invalid, an appropriate message is displayed and the processor reissues the MODE prompt. If both reference axis and element set are valid but their combination is invalid, the processor displays an invalid reference axis/element set combination message and reissues the MODE prompt. (See table 1.2-VI of reference 1.)

INPUT, LIST, EDIT, STOP and EXIT modes - Valid user responses to the MODE prompts are INPUT, LIST, EDIT, STOP, EXIT, or the first two characters of each (e.g., IN, LI, ED, ST, EX), and %, return to Executive. For details, see section 2.0, Functional Description.

5.1.3 Assumptions and Limitations

a. The LSV processor uses either predefined interface state vector buffer data or generates its own data in INPUT mode.

b. The state vector used, generated, and possibly changed, is not necessarily stored into the user AWA. It will be ignored completely when the STOP mode response or % response is given.

c. The vector-type code may be invalid in the predefined interface state vector buffer.

5.1.4 Method

The vector type code (JKNN) residing in SVIN(14) is deciphered into the reference axis index (JRA) and element set index (I15) as follows:

JKNN = IFIX (SVIN(14) + .01 tolerance)

JRA = JKNN/1000 + 1

I15 = JKNN - (JRA - 1) * 1000

Validity checks are then made on JRA and I15 indexes as follows:

$1 \leq JRA \leq 5$

$1 \leq I15 \leq 12$, $I15 = 19$, $I15 = 120$, $I15 = 130$

If the indexes fail the validity check, an appropriate message is displayed and the processor reissues the MODE prompt. If both reference axis and element set are valid, the program checks the validity of reference axis/element set combination as follows:

JRA = TEE or TEG

JRA = EFE and $I15 \leq 7$

JRA = MEE and $I15 \leq 2$

$JRA = M50$ and $I15 \leq 2$ or $8 \leq I15 \leq 13$ or $JRA = 15$

If the above tests fail, the routine displays an invalid reference axis/element set combination message and reissues the MODE prompt. If the user selects an INPUT or EDIT mode, responds to all the prompts properly, and is ready to store the state vector information in the AWA, the routine computes the vector-type code and stores it in SVIN(14) as follows:

$$JKNN = (JRA - 1) * 1000 + KNN$$

where

$$KNN = 1,12 \text{ if } 1 \leq I15 \leq 12$$
$$= 19 \text{ if } I15 = 13$$
$$= 120 \text{ if } I15 = 14$$
$$= 130 \text{ if } I15 = 15$$

5.1.5 Routine Input/Output Variables

The LSV input/output variables are presented in table 5.1-I.

5.1.6 Functional Logic Flow

The functional logic flow for LSV is presented in figure 5.1-1.

5.1.7 Diagnostics and Debug

None.

5.1.8 Special Comments

The user can respond to the MODE prompt using the complete word, INPUT, LIST, EDIT, STOP, EXIT, or the first two characters of each (IN, LI, ED, ST, EX).

5.1.9 Reference

1. Flight Design System-1, System Design Document, Standards. Vol. VI, JSC IN 77-FM-18, January 1978.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LSV

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
SVIN	--	Real	I	--	IT	SVIN	Standard format 15-element input position/velocity state vector.
SVOUT	--	Real	O	--	IT	SVOUT	Standard format 15-element input position/velocity state vector.

NOTES:

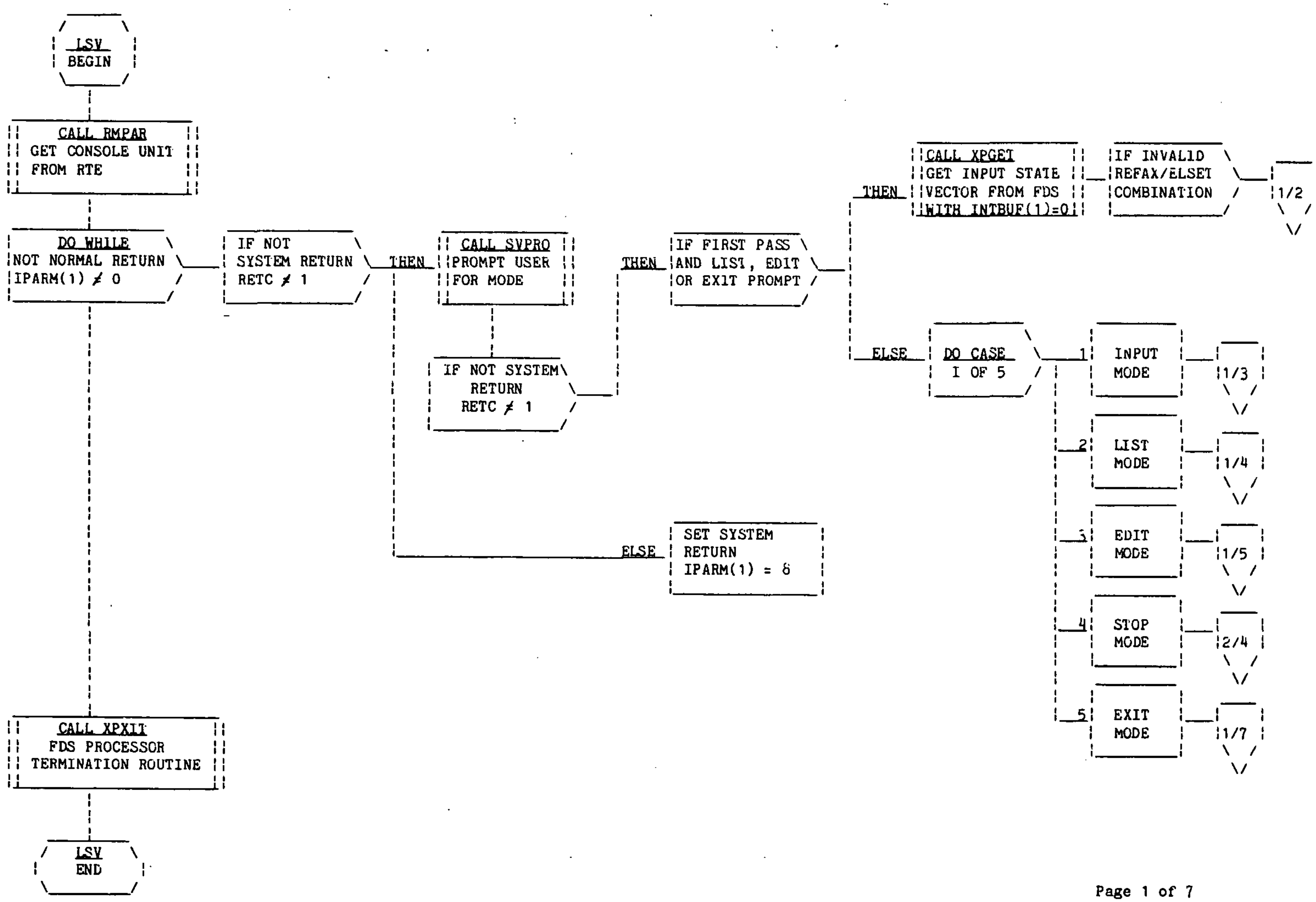
TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

LSV UTILITY PROCESSOR



Page 1 of 7

Figure 5.1-1.- LSV functional logic flow.

LSV UTILITY PROCESSOR - Continued

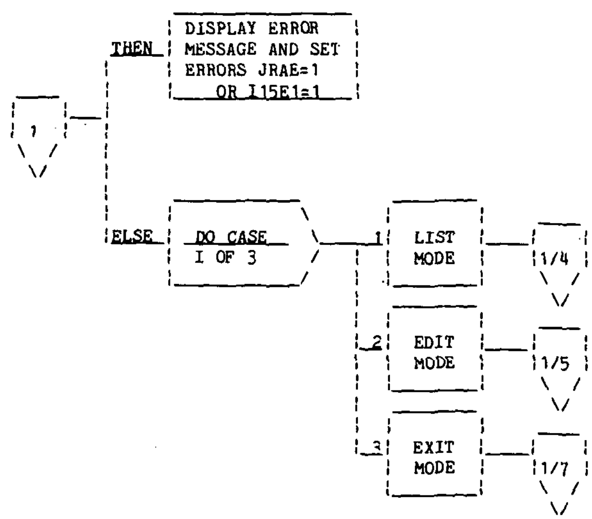


Figure 5.1-1.- Continued

LSV UTILITY PROCESSOR - Continued

INPUT MODE

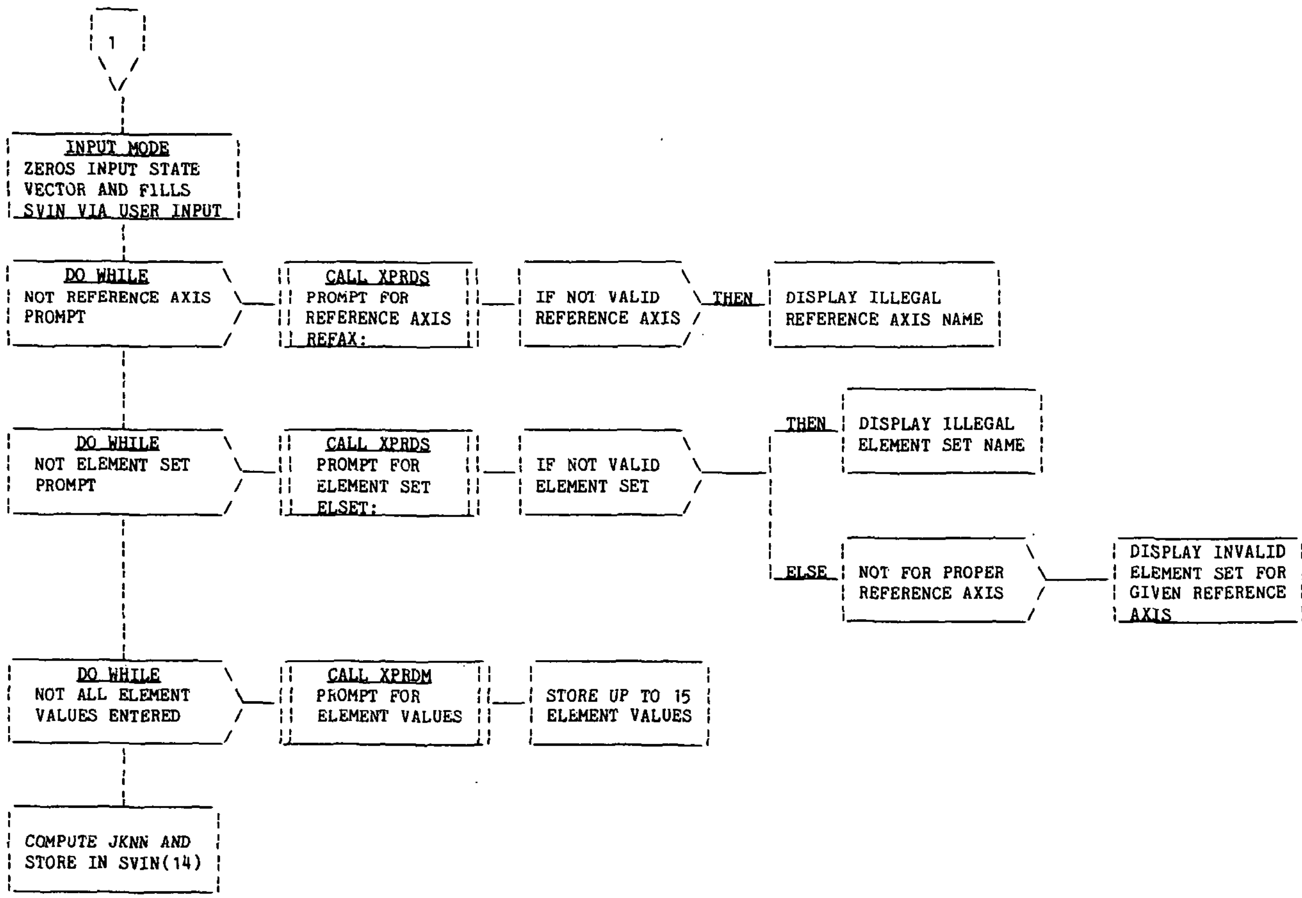
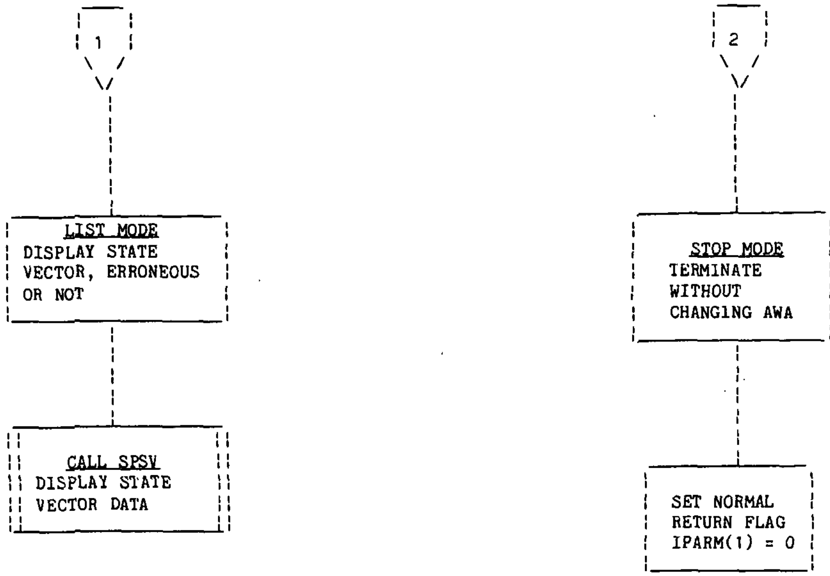


Figure 5.1-1.- Continued.

LSV UTILITY PROCESSOR - Continued

LIST AND STOP MODE



Page 4 of 7

Figure 5.1-1.- Continued.

LSV UTILITY PROCESSOR

EDIT MODE

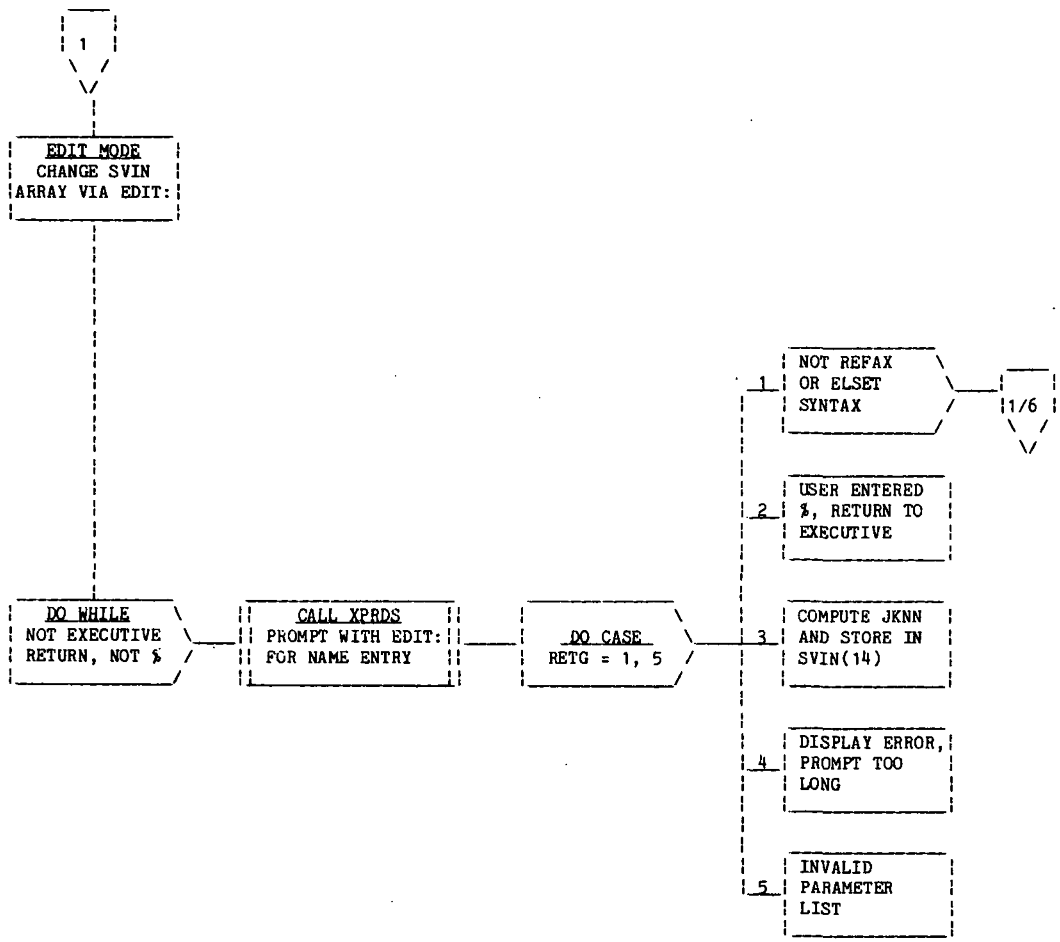


Figure 5.1-1.- Continued.

LSV UTILITY PROCESSOR

EDIT MODE - Concluded

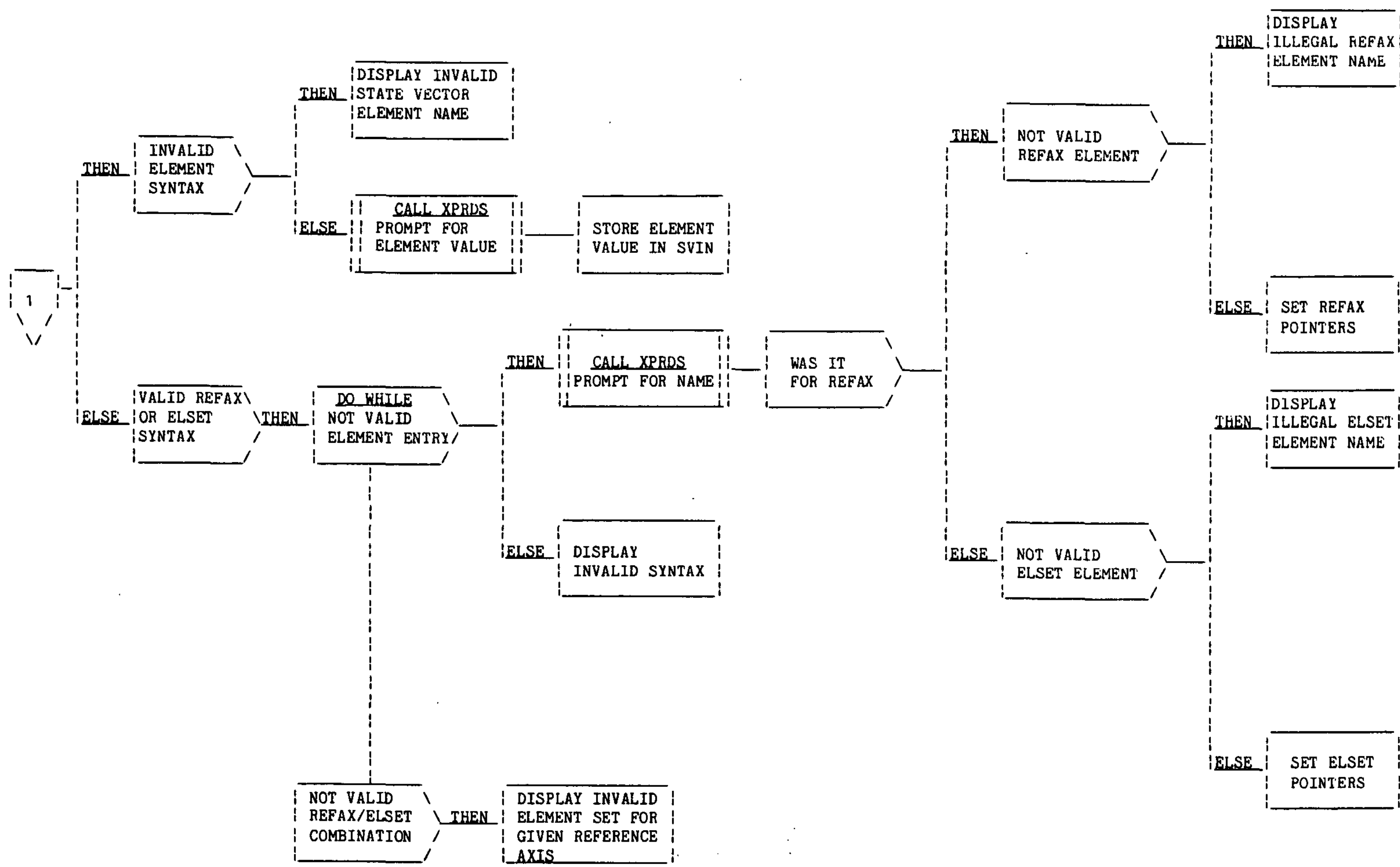


Figure 5.1-1.- Continued.

LSV UTILITY PROCESSOR

EXIT MODE

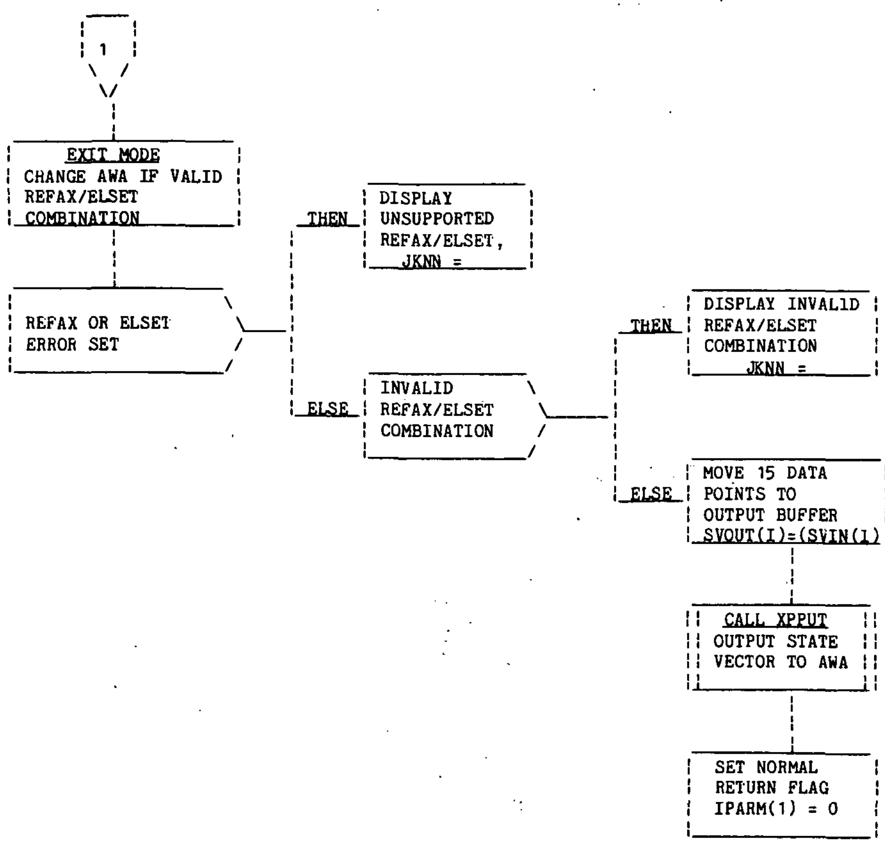


Figure 5.1-1.- Concluded.

5.2 ROUTINE NAME - SVPRO

5.2.1 Purpose

The subroutine SVPRO prompts the FDS user for the LSV mode options and performs analysis of the user response. The FDS user remains in the subroutine until a legal response has been entered or %, return to Executive.

5.2.2 Functional Description

The load state vector is composed of a main routine LSV and a state vector prompt option, subroutine SVPRO. During the operation of the LSV utility processor, LSV main routine calls SVPRO subroutine to prompt the FDS-1 user for the mode option. There are five LSV mode options: INPUT, LIST, EDIT, STOP, and EXIT. The MODE prompt is the first prompt issued to the user, and by response to this prompt the user controls the operational flow of the LSV processor. The MODE prompt is of the form

LSV MODE

(INPUT, LIST, EDIT, STOP, EXIT):(user response)

The user responds with one of the mode options within the parentheses. Upon return to LSV main routine, the flow of the processor execution then proceeds to one of the mode operations.

5.2.3 Assumptions and Limitations

The subroutine SVPRO handles only MODE prompts.

5.2.4 Method

The subroutine SVPRO issues a MODE prompt to the FDS-1 user

LSV MODE

(INPUT, LIST, EDIT, STOP, EXIT):(user response)

and waits for the user response. The user response is then analyzed for proper input and valid response. The user may respond with a full word (e.g., INPUT), or the first two characters of the word; both responses are acceptable. If the user response is invalid, the subroutine displays an appropriate error message and reissues the MODE prompt.

To exit from the subroutine SVPRO, the user must respond to the MODE prompt with a valid mode option or a %, abort execution, and return to the FDS Executive. For abort execution, a flag (RETC) is set to 1 to let the LSV main program know to terminate processor execution and return to the Executive. For valid mode response, a mode pointer (MODE) is set as follows:

MODE	= 1,	INPUT mode
	= 2,	LIST mode
	= 3,	EDIT mode
	= 4,	STOP mode
	= 5,	EXIT mode

The mode pointer tells the LSV main program which mode of operation to execute.

5.2.5 Routine Input/Output Variables

Routine solicited (prompted) inputs - The routine MODE prompt is provided in table 5.2-I.

Routine messages - Table 5.2-II contains the routine messages that are displayed whenever an invalid entry is made.

5.2.6 Functional Logic Flow

The functional logic flow for SVPRO is presented in figure 5.2-1.

5.2.7 Diagnostics and Debug

None.

5.2.8 Special Comments

The FDS user may respond to the MODE prompt using the complete word, INPUT, LIST, EDIT, STOP, EXIT, or the first two characters of each (IN, LI, ED, ST, EX).

5.2.9 References

None.

TABLE 5.2-I.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR SVPRO

Prompt	Meaning	Valid responses
LSV MODE (INPUT, LIST, EDIT, STOP, EXIT):	LSV MODE prompt	"INPUT" - Input mode "LIST" - List mode "EDIT" - Edit mode "STOP" - Stop mode; terminate without transferring to AWA "EXIT" - Exit mode; transfer the state vector contents to AWA and terminate execution % - Return to Executive

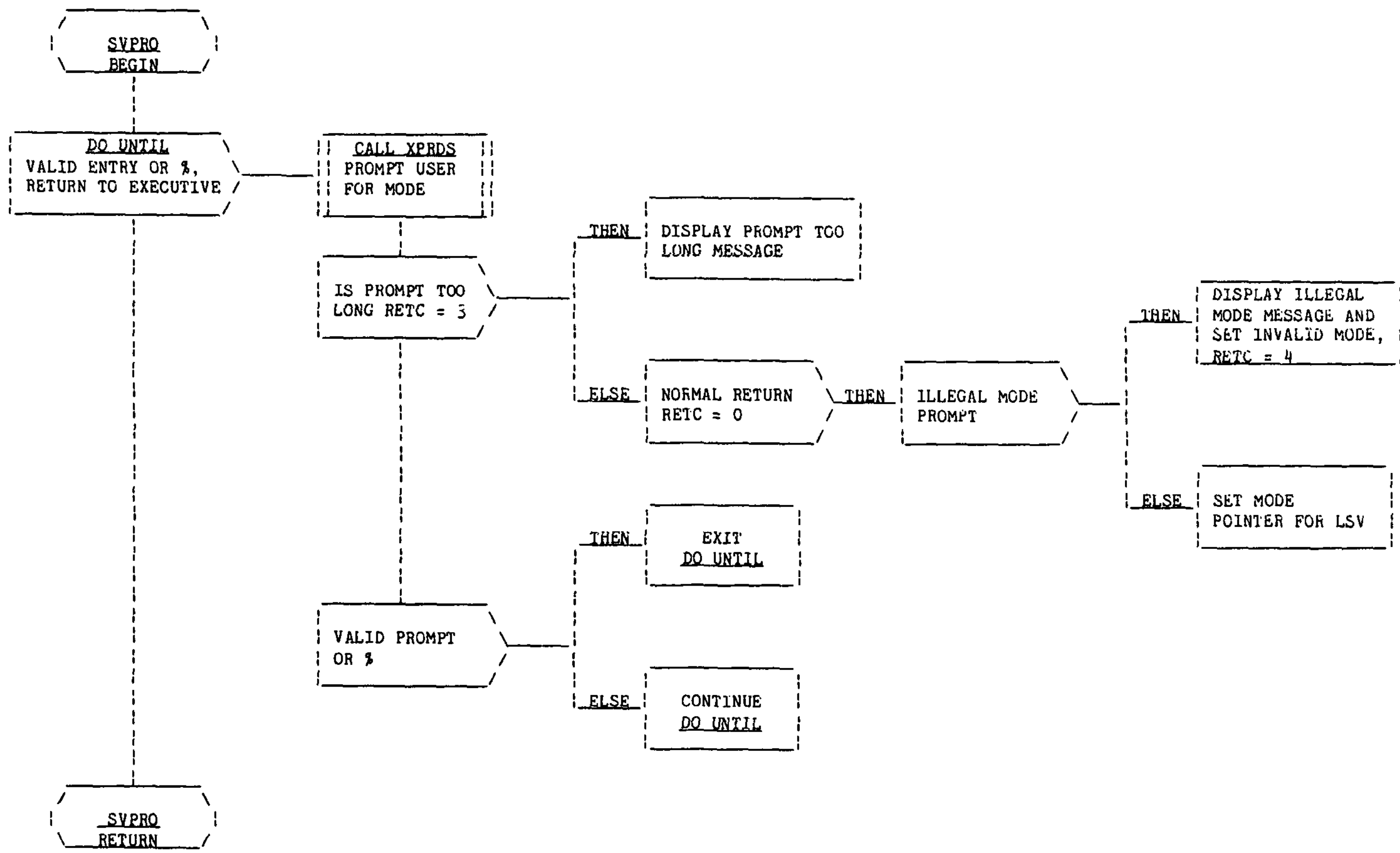
TABLE 5.2-II.- PROCESSOR MESSAGE TABLE

PROCESSOR SVPRO

MSG no.	Message ID block	Message text block and explanation
1	*LSV*	"*LSV* PROMPT TOO LONG, TRY AGAIN." Meaning: The user response to a prompt was too long. Severity: Not applicable. Action required by user: Correct input via same prompt.
2	*LSV*	"*LSV* ILLEGAL MODE PROMPT, TRY AGAIN." *Meaning:* *The user has entered a bad mnemonic to a MODE prompt.* Severity: N/A Action required by user: Correct input via a MODE reprompt.

MODE PROMPT SUBROUTINE

SVPRO



Page 1 of 1

Figure 5.2-1.- SVPRO functional logic flow.

LAUNCH WINDOW PROCESSOR (LWP)

1.0 PURPOSE

The Launch Window Processor (LWP) performs two main functions. First, the LWP finds the analytical inplane launch time, then adjusts this time by an input value to determine the optimum launch lift-off time (minimum payload loss lift-off time). Second, the LWP calculates the insertion (MECO) conditions necessary for a chaser vehicle to effect rendezvous at some time later with a target vehicle already in orbit.

2.0 FUNCTIONAL DESCRIPTION

The LWP is composed of two main programs: the Launch Window Time (LWT) program and the Recommended Lift-off Time (RLOT) program. Given an inorbit target vehicle state vector, the LWT calculates the inplane time of the launch site with the target orbit plane. Within LWT, the GMTLO* time (i.e., phase match or zero-phase angle at insertion point) table may be selected by the user. Also, the Launch Window Parameter table (LWPT) may be generated upon user request. LWPT presents a time history of selected launch targeting quantities around the inplane point.

For a rendezvous mission, the Shuttle launch window is the overlay of two separate windows: the plane window and the phase window. The plane window is centered on the lift-off time that will provide an inplane launch with minimum payload loss by the booster. Because the booster does not begin steering toward the desired launch plane at lift-off, the minimum payload loss lift-off time does not coincide with the analytical inplane lift-off time, but occurs some time earlier. The plane window opens and closes when the booster propellant allotted for yaw steering is depleted.

LWT is used to find the analytical lift-off time. This time is adjusted by an input value to determine the optimum launch lift-off time. The phase window is a function of the rendezvous sequence chosen. It is not considered in LWT.

Whereas LWT is a general information program that determines the inplane times for a specified day, RLOT uses analytical equations to determine the insertion vector required for a chaser vehicle to rendezvous at some later time with a target vehicle already in orbit.

For a rendezvous mission, the Shuttle must be inserted into orbit coplanar, or nearly coplanar, with the target orbit. The insertion (MECO) conditions necessary to effect rendezvous are dependent upon the Shuttle position in orbit relative to the target position in orbit at the time of insertion (MECO). For the Apollo Program, in order for the Saturn ascent guidance equations to provide steering commands to the second stage, the quantities V_i, γ_i, I, Θ and $\dot{\Theta}$ are required as input. The last three quantities are used to define the desired chaser plane. The second stage is steered so that, at guidance termination, the chaser velocity vector has no component perpendicular to this plane, and the displacement between the desired plane and the actual plane is zero.

To compensate for the effects of oblateness of the Earth, the guidance is provided with targeting elements that define a phantom chaser plane that does not exactly correspond with the plane of the actual target orbit.

For Shuttle, an I_{YD} vector (unit vector normal to the target plane) replaces the quantities I, Θ and $\dot{\Theta}$. However, the guidance philosophy remains the same as in Apollo, and in fact, I_{YD} is calculated after I, Θ and $\dot{\Theta}$ are determined. Consequently, it is still necessary to calculate the quantities so the I_{YD} may be derived.

The targeting quantities are defined as follows: V_i is the insertion (MECO) velocity, γ_i is the insertion (MECO) flightpath angle, R_i is the insertion (MECO) radius, I is the osculating inclination, Θ is the angle between the upcoming descending node of the phantom plane and the launch site meridian at the time of guidance reference release, and $\dot{\Theta}$ is the secular nodal regression rate of the phantom plane.

RLOT is designed to calculate and present as output the six targeting quantities (actually V_i, γ_i, and R_i are input), the time of guidance reference release, and a corresponding chaser insertion orbit state vector. This vector, the target vector updated to the time of insertion, and some of the displayed quantities are also stored by LWP so that they are available for onorbit work in other processors.

3.0 ASSUMPTIONS AND LIMITATIONS

The LWT will compute the inplane times for one launch day at a time. If the user desires to see more than one launch day, it is necessary to run the LWP more than once, changing the requested day of launch (DAY).

The logic of the LWT requires that the threshold time (computed using DAY as input) to start the search for the inplane times be equal to or greater than the input vector time for the target vehicle. This means, in effect, that no vector will be propagated backwards in time. Specifically, this means that the input target vector will always be updated forward in time from input time to the time of the calculated chaser insertion (MECO) time. Because of the characteristics of some of the launch targeting options, this restriction does not hold for RLOT.

LWT does not generate or display a chaser insertion vector. It is not necessary to call the launch targeting logic in order to compute the inplane times. Consequently, no vector is determined.

The GMTLO* table in LWT can compute a maximum of 10 star times. This number is determined by sizing restrictions and practicality.

A possible area for confusion are the terms chaser, target, insertion, and MECO. Although LWP on FDS-1 will be used mainly in Shuttle flight planning, LWP originally was designed for the Apollo Program. Also, other missions besides Shuttle may be simulated. Thus, two sets of terms found in this text are considered interchangeable. The chaser vehicle is the vehicle that is being launched into orbit. It can mean the CSM, the Orbiter, the Soyuz, etc. The target vehicle is the vehicle already in orbit. It may be a satellite, the Soyuz, Skylab, etc. Similarly, insertion and MECO are considered to be synonymous words in LWP documentation and software. The equations in LWP were derived and are used under the assumption that they are valid at the time the main engine cutoff of the booster occurs. In Apollo, Skylab, and ASTP this time is called insertion. In Shuttle, this time is called MECO. In LWP, insertion is used as the more general term, and the assumption is that, for a Shuttle launch, insertion and MECO indicate the same time (the time of MECO).

All altitude computations are referenced to a spherical Earth radius. In LWP, the reference radius is the equatorial Earth radius defined for the Fischer ellipsoid and stored in the !!GLCN array.

DTOPT is needed to adjust the analytical solution for the inplane launch time to the integrated launch trajectory inplane launch time. DTOPT is used in computing the northerly and southerly inplane launch points, the parallel wedge angle and the parallel azimuth in the targeting display and in the launch parameter table.

LOT="RSTAR" and LOT="PSTAR" are the same option if the threshold time (either GMTLOR or TPLANE) is input. LOT="PSTAR" is most useful when LW="WT". Then, TPLANE is computed in the launch window and used in launch targeting.

LOT="GMTLOR", "TPLANE" and "TYAW" are the same option if the lift-off time (either GMTLOR or TPLANE + TRANS) is input. LOT="TPLANE" and "TYAW" is most useful when LW="WT." Then, TPLANE is computed in the launch window and used in launch targeting.

The input for LOT="TYAW" is the same as for LOT="TPLANE." Following the first iteration, TYAW, the inplane lift-off time based on the target orbit and phase angle (i.e., the lift-off time required when no yaw steering by second stage guidance is required) is calculated. Then, it is used during the second iteration to find GMTLOR.

On input, OFFSET should be measured using the same sign convention as desired for insertion phase angle. This determined by NEGTIV. Then, if it is desired to have the insertion phase angle wrapped using WRAP, OFFSET will be wrapped correspondingly.

Positive DELNO places the launch vehicle node to the east of the target node. Negative DELNO places the launch vehicle node to the west of the target node. Therefore, for posigrade orbits, DELNO should be positive if the launched vehicle node will regress faster than the node of the target vehicle; it should be negative if the reverse is true. The opposite sign convention should be used for retrograde orbits.

If STABLE="YE" is called, NS determines which inplane nodes will be used. For NS="OP", STARS and STARE will define the time limits around the opening inplane launch time. For NC="CL", STARS and STARE will define the limits around the closing node. For NS="OC", STARS and STARE will be used around both nodes.

Note that an integrated launch simulation is not performed by LWP. Using such quantities as PFA, PFT, LATLS, LONGLS, RINS, VINS, GAMINS, and the target vector, which are either defined or derived from a launch simulation program, analytical equations derive the chaser insertion vector. This drastically reduces the computation time required to define the chaser orbit without significantly affecting the accuracy. However, it does make LWP dependent on data obtained from a previously run launch simulation program.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the LWP interface table parameters is provided in table 4-I. GLOCON is a set of constants universal to all the processors that will be maintained in the master data base. The LWP will access this array for the constants it requires. The default values will be stored in !!GLCN.

   The LWP will access the SESCON array to obtain the session related constants generated by the user upon execution of the system utility processor BASTM. The default values will be stored in !SESCN and will be standard for all processors.

   Specific constants and tolerances required by the LWP will be maintained in PROCON. These parameters are used mainly for iteration tolerances and internal print analysis.

   TRGVEC is the position/velocity input state vector of the target vehicle. This vector is updated to the time of insertion by the LWP. All internal LWP calculations involving the state vector use the true equator and Greenwich meridian of epoch coordinate system (TEG) and the Cartesian element set. Thus, if the input vector is not in this coordinate system and element set, the vector conversion processor must be used prior to using the LWP in order to place the state vector in the acceptable coordinate system and element set. The vector times are in Greenwich mean time (GMT).

   The chaser vehicle characteristics are input in CCD, CAREA, and CWHT. The quantities are used by RLOT for vector propagation and then stored in the chaser vehicle position/velocity phase table for use by other processors.

   LW is the launch window/launch targeting options flag. This flag determines that LWT or RLOT, or both, are called by the user.

   The following set of parameters are used in LWT. NS is a flag to determine which inplane times on a specified day will be calculated. DAY defines the day on which the launch window times will be calculated. LPT, TSTART, TEND, and TSTEP are used by the LWPT to define the limits and the time step of the LWPT calculations, STABLE, STARS, and STARE are used to define the limits of the GMTLO* table calculations.

   LATLS, LONGLS, PFT, PFA, and YSMAX, are used in both LWT and RLOT to help define the chaser plane at insertion. LATLS is the geocentric latitude and

LONGLS the geographic longitude of the launch site. PFT is the powered flight time to insertion, and PFA is the powered flight arc of the chaser from lift-off to insertion. YSMAX is the maximum yaw steering allowed by the booster.

DTOPT is used by LWT to determine the empirical inplane lift-off time after the analytical inplane time has been calculated. DTGRR is the delta time from lift-off that defines the time of guidance reference release (TGRR). LAZCOE is an array of four real words defining the launch azimuth coefficients so that the optimum launch azimuth (LAZ) may be calculated in RLOT. RINS, VINS, and GAMINS are used in both LWT and RLOT. RINS is the radius desired at chaser insertion (MECO), VINS is the velocity desired, and GAMINS is the flightpath angle desired at insertion.

The following set of parameters are used only in RLOT. LOT is the lift-off time option flag for launch targeting. GMTLOR is the recommended or threshold lift-off GMT. OFFSET is the input to specify the phase angle desired at insertion. BIAS is a delta time that is added to the GMTLO* time to obtain the desired lift-off time. TPLANE is the GMT of inplane lift-off. Depending on the choice of LW, it may be computed by LWT or input by the user. TRANS is the delta time added to the inplane time to obtain the desired lift-off time. INSCO is the insertion cutoff conditions option flag. DHW is the desired height difference between target and chaser or the desired altitude of the chaser at an input angle from insertion. DU is the angle from insertion to obtain the desired altitude or height difference input in DHW. ANOM is the nominal semimajor axis at chaser insertion. DELNOF is the differential nodal precession from insertion to rendezvous option flag. DELNO is the angle that is added to the target descending node to account for the differential nodal precession between the chaser and the target. Depending on the value of DELNOF, DELNO may be input by the user or calculated by RLOT. NEGTIV is the insertion phase angle control flag. It determines if the user wishes the phase angle to be measured from 0 to 2π, -2π to 0, or $-\pi$ to π. This choice affects the size of DELNO when it is calculated. WRAP is the flag to wrap the phase angle. This choice will affect the size of the rendezvous maneuvers if the user chooses to run the OMP Processor after RLOT.

SVPROP is the state vector propagation flag. Either a conic, two-body, propagation or the analytic ephemeris generator (AEG) may be chosen. Note, however, that the conic propagator does not account for nodal precession. "AEG" should be the default value for SVPROP.

PVTABT is the position/velocity state vector phase table output quantities for the target vehicle. Similarly, PVTABC is the phase table output quantities for the chaser vector. PVTABT outputs two vectors; the initial vector and a vector defined at the time of insertion. PVTABC outputs the chaser insertion vector. Note, however, they are not output when LW="LW" is chosen by the user. The vectors will be output to the tables in the TEG coordinate system and the Cartesian element set.

SUMTAB is the output summary table data array. Information placed there by the LWP is available for recall at a later time in the user session.

By executing SMPRT, the information will be listed in tabular form. Note that the summary table is not output when LW="LW" is chosen by the user.

GMTLO is the Greenwich mean time, in hours, of the computed chaser vehicle lift-off. The user has the option of updating the !SESCN array with this value. The default value is !SESCN(11).

b. Interface table data array definitions - The definition of the input/output data arrays appearing in the LWP interface table is provided in table 4-II.

c. Interface table data file definitions - None

d. Processor solicited (prompted) inputs - The processor solicited (user prompts) are provided in table 4-III. These prompts allow the user to cause a pause in the execution of the processor while the present results are considered. Also, the target vehicle vector in LWT and the target or chaser vehicle vector in RLOT may be chosen for display by the user.

e. Processor displays and display parameter definition tables - LWT generates four displays. The format of the launch window time display is shown in table 4-IV(a), and a definition of the display variables is provided in table 4-IV(b). The phase angle and apogee/perigee parameters are in user-selected external units. The latitude and longitude of launch site are given in degrees. This display is generated whenever LWT is called.

Three optional displays are available through LWT. The GMTLO* table computes the time of zero phase angle (star time) for each orbit over the time limits defined by the user. Table 4-IV(c) shows the format of this display, and table 4-IV(d) gives the definition of the display variables. None of the units for the parameters are user selected.

The optional launch window parameter table (LWPT) display is available through LWT. This table presents a time history of certain targeting quantities through a launch window. The format of this display is shown in table 4-IV(e), and a definition of the display parameters is provided in table 4-IV(f). All time display parameters and the latitude and longitude of launch site are internally defined. All the other display parameters are in the user-selected external units.

A state vector display for the target vehicle may also be generated, but the format is the same as the one generated in RLOT. It will be described with the other RLOT display.

RLOT generates two displays. The Shuttle prelaunch targeting display format is shown in table 4-IV(g), and a definition of the display variables is provided in table 4-IV(h). All of the display parameters are in the user-selected external units except for the following time parameters. GMTLO, TINS, GETLO, TGRR, GMTLO*, TYAW, and TPLANE are in hours, minutes, and seconds.

An optional LWP state vector display may be generated by RLOT. The format of this display is shown in table 4-IV(i), and a definition of the display variables is provided in table 4-IV(j). All the display parameters are in the user selected external units except for PERIOD, which is in minutes.

The user may select which vehicle state to display (in LWT, the target is the only one displayed).

f. Processor message table - Table 4-V contains the processor messages that may be displayed to the user. The first two messages are fatal input error types. When the error is encountered, the processor will return control to the FDS-1 Executive. The remaining error messages involve warnings only, such as convergent tolerances not being met. The processor will continue to execute in these cases.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VI.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimen-sion (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array; master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
PROCON	AWA	Free	I	24	24		LWP constants and tolerances
TRGVEC	AWA	Real	I	30	15		Position/velocity input state vector for target vehicle
CCD	AWA	Real	I	2	1	2.0	Chaser vehicle coefficient of drag
CAREA	AWA	Real	I	2	1		Chaser vehicle reference area
CWHT	AWA	Real	I	2	1		Chaser vehicle weight at insertion
LW	AWA	2CH	I	1	1		Launch window/launch targeting options ="LW"; compute launch window only ="LT"; compute launch targeting only ="WT"; compute launch window and launch targeting

NOTES	CLASS	TYPE	USE
	AWA Disk	Free 2CH 72CH Intg 6CH Mix Real 18CH Symb Dubl 36CH	I = Input O = Output I/O = Input/Output

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
NS	AWA	2CH	I	1	1		Inplane launch window opening and closing times option ="OP"; compute inplane opening only ="CL"; compute inplane closing only ="OC"; compute opening and closing
DAY	AWA	Intg	I	1	1	0	Day on which launch window times are computed, relative to base date
LPT	AWA	2CH	I	1	1		Launch window parameter table options ="NO"; do not generate table ="OP"; around launch window opening ="CL"; around launch window closing ="OC"; around both opening and closing ="EN"; through entire window
TSTART	AWA	Real	I	2	1		Delta time prior to inplane time to start parameter table
TEND	AWA	Real	I	2	1		Delta time after inplane time to stop parameter table
TSTEP	AWA	Real	I	2	1		Time step in parameter table
STABLE	AWA	2CH	I	1	1		GMTLO* table flag ="NO"; do not compute table ="YE"; compute table

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
STARS	AWA	Real	I	2	1		Delta time prior to each inplane launch point to start GMTLO* search
STARE	AWA	Real	I	2	1		Delta time after each inplane launch to end GMTLO* search
LATLS	AWA	Real	I	2	1		Geocentric latitude of launch site
LONGLS	AWA	Real	I	2	1		Geographic longitude of launch site
PFT	AWA	Real	I	2	1		Powered flight time
PFA	AWA	Real	I	2	1		Powered flight arc
YSMAX	AWA	Real	I	2	1		Yaw steering limit
DTOPT	AWA	Real	I	2	1		Delta time to be subtracted from analytical inplane launch time to obtain empirical inplane launch time
DTGRR	AWA	Real	I	2	1		ΔT from lift-off, which defines the time of guidance reference release
LAZCOE	AWA	Real	I	8	4		Launch azimuth coefficients

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
RINS	AWA	Real	I	2	1		Radius of insertion
VINS	AWA	Real	I	2	1		Velocity magnitude at insertion
GAMINS	AWA	Real	I	2	1		Flightpath angle at insertion
LOT	AWA	6CH	I	3	1		Lift-off time options for launch targeting ="GMTLOR"; lift-off on input time ="OFFSET"; compute lift-off time to achieve a desired phase angle (OFFSET) at insertion ="RSTAR"; lift-off on GMTLO* + BIAS, using GMTLOR as threshold time ="PSTAR"; lift-off on GMTLO* + BIAS, using TPLANE as threshold time ="TPLANE"; lift-off based on inplane launch time (GMTLO = TPLANE + TRANS). ="TYAW"; iterate to lift-off on inplane launch time based on target orbit phase angle (final GMTLO = TYAW + TRANS).
GMTLOR	AWA	Real	I	6	3		Recommended or threshold lift-off Greenwich mean time; hours, minutes, seconds

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
OFFSET	AWA	Real	I	2	1		Phase angle desired at insertion
BIAS	AWA	Real	I	2	1		Bias that is added to GMTLO* to produce lift-off time
TPLANE	AWA	Real	I	6	3		Greenwich mean time of inplane lift-off; hours, minutes, seconds (computed internally when LW = "LW" or LW = "WT")
TRANS	AWA	Real	I	2	1		Delta time added to inplane time to obtain lift-off time
INSCO	AWA	6CH	I	3	1		Insertion cutoff conditions option flag ="VGAMR"; Input VINS, GAMINS, RINS ="DHGAMR"; Input GAMINS, RINS and height difference desired at an input angle from insertion. Input: GAMINS RINS DHW DU ANOM

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
INSCO (cont'd)	AWA	6CH	I	3	1		="HGAMR"; Input GAMINS, RINS and altitude desired at an input angle from insertion. Input: GAMINS RINS DHW DU ANOM
DHW	AWA	Real	I	2	1		Desired height difference between chaser and target, or altitude of chaser, at input angle from insertion
DU	AWA	Real	I	2	1		Angle from insertion to obtain a given altitude, or delta altitude
ANOM	AWA	Real	I	2	1		Nominal semimajor axis at insertion
DELNOF	AWA	2CH	I	1	1		Flag for option to compute differential nodal regression from insertion to rendezvous. ="IN"; input DELNO ="CO"; compute DELNO
DELNO	AWA	Real	I	2	1		Angle that is added to the target descending node to account for differential nodal regression

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
NEGTIV	AWA	2CH	I	1	1		Initial phase angle control flag ="PO"; use positive value ="NE"; use negative value ="LO"; use lowest absolute value
WRAP	AWA	Intg	I	1	1	0	Flag to wrap initial phase angle =N; add 2N† to phase angle (N is an integer)
SVPROP	AWA	6CH	I	3	1	"AEG"	State vector propagator selection flag ="CON"; conic ="AEG"; analytic ephemeris generator
PVTABT	AWA	Real	O	120	30,2		Target vehicle position/velocity phase table; two vectors defined at: (1) initial time (2) time of insertion (MECO). This is not output when LW = "LW".
PVTABC	AWA	Real	O	60	30		Chaser vehicle position/velocity phase table; one vector defined at time of insertion (MECO). This is not output when LW = "LW".
SUMTAB	AWA	Free	O	136	8,17		Output summary table. This is not output when LW = "LW".

NOTES	CLASS	TYPE	USE
	AWA Disk	Free 2CH 72CH Intg 6CH Mix Real 18CH Symb Dubl 36CH	I = Input O = Output I/O = Input/Output

TABLE 4-I.- Concluded

PROCESSOR LWP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GMTLO	AWA	Real	O	2	1	!SESCN(11)	Greenwich mean time of chaser vehicle lift-off, hours. This is not output when LW = "LW".

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR LWP

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array, master data base element; see JSC IN 78-FM-60, volume 1, table 7.2-III for definition of contents.
SESCON	(1) . . . (90)	!SESCN	Input/output session constants array; see JSC IN 78-FM-60, volume I, table 7.2-II for definition of contents.
PROCON	(1)	0	Output unit on which to write all print; the default value is a flag, saying to use the user terminal; a positive value identifies the output unit.
	(2)	0	Not used
	(3)	0	Internal print flag option. = 0; no internal print = 1; internal print
	(4)	0	Not used
	(5)	20	Maximum iterations limit
	(6)	0	Not used
	(7)	25.	First guess change in value of independent variable (used in subroutine ITERV)
	(9)	2.E-4	Transfer angle tolerance, radians
	(11)	6.E-4	Eccentricity tolerance
	(13)	2.E-4	Travel angle convergence tolerance, radians
	(15)	0	Independent variable guess switch = 0; no independent variable input = 1; a guess for independent variable is input
	(16)	0	Not used
	(17)	0.	An initial guess for independent variable
	(19)	250.	Radius iteration tolerance, feet
	(21)	1.0	Time iteration tolerance, seconds
	(23)	0.1	Tolerance on plane change ΔV for calculating launch window inplane launch points, feet per second
TRGVEC	(1) . . . (15)		Position/velocity input state vector for target vehicle; see JSC IN 78-FM-60, volume I, figure 7.3-2 for definition of contents.

TABLE 4-II.- Continued

PROCESSOR LWP

Array name	Index location	Default value	Definition
LAZCOE	(1)		A0 , Degrees — Launch azimuth coefficients
	(2)		A1 , Dimensionless
	(3)		A2 , Dimensionless
	(4)		A3 , $Degrees^{-1}$
GMTLOR			Recommended or threshold lift-off Greenwich mean time
	(1)		Hours
	(2)		Minutes
	(3)		Seconds
TPLANE			Greenwich mean time of inplane lift-off
	(1)		Hours
	(2)		Minutes
	(3)		Seconds
PVTABT			Position/velocity phase table for target vehicle; see JSC IN 78-FM-60, volume I, figure 7.3-13 for definition of contents.
			Position/velocity state and propagation data at:
	(1,1) . . . (30,1)		1. Initial time
	(1,2) . . . (30,2)		2. Time of insertion (MECO)
PVTABC			Position/velocity phase table for chaser vehicle; See JSC IN 78-FM-60, volume I, figure 7.3-13, for definition of contents.
			Position/velocity state and propagation data at:
	(1) . . . (30)		1. Time of insertion (MECO)

TABLE 4-II.- Concluded

PROCESSOR LWP

Array name	Index location	Default value	Definition
SUMTAB	(1,1)		ERROR = Reserved error flag for summary table output processing
	(1,2)		GMTLO = Greenwich mean time of lift-off
	(1,3)		GETLO = Ground elapsed time of target vehicle at time of chaser vehicle lift-off
	(1,4)		TINS = Greenwich mean time of insertion (MECO)
	(1,5)		AZL = Optimum launch azimuth
	(1,6)		VIGM = Velocity magnitude of chaser at insertion (MECO)
	(1,7)		RIGM = Radius of chaser at insertion (MECO)
	(1,8)		GIGM = Flightpath angle of chaser at insertion (MECO)
	(1,9)		IIGM = Inclination of chaser at insertion (MECO)
	(1,10)		TIGM = θ, angle measured from launch site meridian to the chaser descending node, measured at SRB ignition command
	(1,11)		TDIGM = Rate of change of TIGM, measured at SRB ignition command, degrees per second
	(1,12)		DN = Geographical descending node of chaser
	(1,13)		DELNO = Angle to account for differential nodal precession between the chaser and target orbits, measured at insertion
	(1,14)		PA = Phase angle between target and chaser, measured at insertion
	(1,15)		TPLANE = Greenwich mean time of inplane lift-off
	(1,16)		LATLS = Geocentric latitude of launch site
	(1,17)		LONGLS = Geographic longitude of launch site

TABLE 4-III.- PROCESSOR SOLICITED (PROMPTED) INPUTS

PROCESSOR LWP

Prompt	Meaning	Valid responses
LWT DISPLAY:	Execution of LWT is completed; ready to show displays.	= b̸b̸; continue: no display shown = "LW"; launch window display = "TV"; target vehicle vector = "ST"; GMTLO* table display
LWPT DISPLAY:	Execution of LWPT is completed; ready to show displays.	= b̸b̸; continue; no display shown = "LP"; launch window parameter table display
RLOT DISPLAY:	Execution of RLOT is completed; ready to show displays.	= b̸b̸; continue; no display shown = "LT"; launch targeting display = "CV"; chaser vehicle vector = "TV"; target vehicle vector

TABLE 4-IV.- PROCESSOR DISPLAY AND DISPLAY PARAMETER DEFINITIONS TABLE

(a) Launch window time display

PROCESSOR LWP

```
 1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75
                        LAUNCH WINDOW TIMES FOR DAY XX

                               DATE      XX,XXXX

 INPLANE LAUNCH TIME

                         GMT             GMT               EST                PST
                      DAY HR MIN SEC   SECONDS        DAY HR MIN SEC     DAY HR MIN SEC
*OPENING OF WINDOW     XX XX  XX  XX   XXXXXX.X        XX XX  XX  XX      XX XX  XX  XX
*
*CLOSING OF WINDOW     XX XX  XX  XX   XXXXXX.X        XX XX  XX  XX      XX XX  XX  XX

   OPEN PHASE ANGLE  = +XXX.XX                                    LAYLS  =  XX.XX DEG
   CLOSE PHASE ANGLE = +XXX.XX                                    LONGLS = XXX.XX DEG

   APOGEE ALTITUDE ON DAY   XX                  ABOVE REF. XXXXXXXXX.XX
   PERIGEE ALTITUDE ON DAY
```

TABLE 4-IV.- Continued

(b) Display parameter definition table for the launch window times for day display

PROCESSOR LWP

LAUNCH WINDOW TIMES FOR DAY

Display parameter label	Parameter definition
DAY	Day on which inplane times are computed (number of days from base time)
DATE	Month, day of month, and year on which inplane time occurs
GMT	Greenwich mean time of opening or closing inplane lift-off time
EST	Eastern standard time of opening or closing inplane lift-off time
PST	Pacific standard time of opening or closing inplane lift-off time
OPEN	Ascending node inplane lift-off point
CLOSE	Descending node inplane lift-off point
LATLS	Geocentric latitude of launch site, degrees
LONGLS	Geographic longitude of launch site, degrees
APOGEE	Target apogee altitude
PERIGEE	Target perigee altitude

TABLE 4-IV.- Continued

(c) GMTLO* table display

PROCESSOR LWP

```
                             GMTLO TABLE

  LATLS   =     XX.XX
  LONGLS       XXX.XX

                  GMT                                                    STAR - PLANE
               D     H     M     S        GMT SECONDS         D        H      M       S             SECONDS

*STAR    1  =  XX    XX    XX    XX        XXXXXX.XX          XX       XX     XX      XX           +XXXXX.XX
*STAR    2  =  XX    XX    XX    XX        XXXXXX.XX          XX       XX     XX      XX           +XXXXX.XX
   .     .                                     .                              .                         .
   .     .                                     .                              .                         .
   .     .                                     .                              .                         .
*STAR   10  =  XX    XX    XX    XX        XXXXXX.XX          XX       XX     XX      XX           +XXXXX.XX
```

TABLE 4-IV.- Continued

(d) Display parameter definition table for the GMTLO* table display

PROCESSOR LWP

GMTLO* TABLE

Display parameter label	Parameter definition
GMT	Greenwich mean time
STAR-PLANE	Delta time between GMTLO* and inplane times
D	Days
H	Hours
M	Minutes
S	Seconds
*STAR	GMTLO* - time of phase match (i.e., phase angle = 0.)
LATLS	Geocentric latitude of launch site, degrees
LONGLS	Geographic longitude of launch site, degrees

TABLE 4-IV.- Continued

(e) Launch window parameter table display

PROCESSOR <u>LWP</u>

```
                        LAUNCH WINDOW PARAMETER TABLE

  DATE          XX,XXXX
                                                       LATLS  =    XX.XX DEG
                                                       LONGLS =   XXX.XX DEG

       TIMES BETWEEN WHICH LAUNCH PARAMETERS ARE COMPUTED
                                  GMT                   EST
                               H   M   S             H   M  S
     TI =  XXXXXX.XXX         XX  XX  XX.X          XX  XX XX.X
     TF =  XXXXXX.XXX         XX  XX  XX.X          XX  XX XX.X

   GMTLO          GPAZ      AZL        YP      WEDGE     DVPC     PHASE      TIGM
  H  M  S
 XX XX XX.X      XX.XX     XX.XX      X.XXX    X.XXX   XXXX.XX  XXX.XXX    XXX.XX
```

TABLE 4-IV.- Continued

(f) Display parameter definition table for the launch window parameter table display

PROCESSOR LWP

LAUNCH WINDOW PARAMETER TABLE

Display parameter label	Parameter definition
GMT	Greenwich mean time
EST	Eastern standard time
H	Hour
M	Minutes
S	Seconds
TI	GMT initial time, seconds
TF	GMT final time, seconds
GMTLO	GMT lift-off
GPAZ	Gemini parallel launch azimuth
AZL	Optimum launch azimuth
YP	Gemini parallel launch wedge angle
WEDGE	Actual insertion wedge angle
DVPC	Wedge angle delta velocity magnitude
PHASE	Insertion phase angle
TIGM	Angle measured from launch site meridian to chaser descending node, defined at SRB ignition command
LATLS	Geocentric latitude of launch site, degrees
LONGLS	Geographic longitude of launch site, degrees

TABLE 4-IV.- Continued

(g) Shuttle prelaunch targeting display

PROCESSOR <u>LWP</u>

```
 1   DATE            XX,XXXX        SHUTTLE   PRELAUNCH   TARGETING

     GMTLO        XX    XX    XX.X          GETLO       XX    XX    XX.X
 5   TINS         XX    XX    XX.X                                          TGRR      XX    XX    XX.X
                                            AZL                XXX.XXX      VIGM            XXXXX.XXX
     GMTLO*       XX    XX    XX.X          LATLS               XX.XXX      RIGM         XXXXXXXX.XX
     PFA                  XX.XXX            LONGLS             XXX.XXX      GIGM          X.XXXXXXXX
     PFT                 XXX.XXX            GPAZ               XXX.XXX      IIGM               XXX.XXX
10   DN                   XX.XXX            YP                  X.XXXX      TIGM               XXX.XXX
                                                                            TDIGM         +X.XXXXXXX

     TYAW         XX    XX    XX.X          DELNO          X.XXXXXXXX
     TPLANE       XX    XX    XX.X          DELNOD         X.XXXXXXXX
15
           TARGET  ORBIT                                                    LAUNCH  ORBIT

      APOGEE       =      XXXXXX.XX       ABOVE  REF.  XXXXXXXXX.XX  APOGEE        =       XXXXXX.X
      PERIGEE      =      XXXXXX.XX                                  PERIGEE       =       XXXXXX.X
20    INCLINATION  =        XXX.XX                                   T ANOMALLY    =        +XXX.XX
      INS PHASE    =       +XXX.XXX                                  ALTITUDE      =       XXXXXX.X
      DN TARGET    =       +XXX.XXX                                  DH            =       XXXXXX.X
      BIAS         =      XXXXX.XXX                                  TIME          = .XXXXXXXXE+X
24
```

TABLE 4-IV.- Continued

(h) Display parameter definition table for the Shuttle prelaunch targeting display

PROCESSOR LWP

SHUTTLE PRELAUNCH TARGETING

Display parameter label	Parameter definition
DATE	Month, day of month, and year of launch
GMTLO	Greenwich mean time of lift-off, in hours, minutes, seconds
TINS	Greenwich mean time of insertion (MECO), in hours, minutes, seconds
GMTLO*	Greenwich mean time of phase match, in hours, minutes, seconds
PFA	Powered flight arc
PFT	Powered flight time
DN	Descending node of chaser
TPLANE	Greenwich mean time of inplane launch, in hours, minutes, seconds
GETLO	Ground elapsed time of lift-off, in hours, minutes, seconds
AZL	Optimum launch azimuth
LATLS	Geocentric latitude of launch site
LONGLS	Geographic longitude of launch site
GPAZ	Gemini parallel launch azimuth
YP	Gemini parallel launch wedge angle
DELNO	Angle between the target and chaser descending nodes, defined at insertion
DELNOD	Rate of change of DELNO, defined at insertion
TYAW	Greenwich mean time of lift-off to achieve minimum yaw steering, in hours, minutes, seconds
TGRR	Greenwich mean time of guidance reference release
VIGM	Velocity magnitude at insertion (MECO)
RIGM	Radius magnitude at insertion (MECO)
GIGM	Flightpath angle at insertion (MECO)
IIGM	Inclination at insertion (MECO)
TIGM	Angle measured from launch site meridian to chaser descending node, defined at TGRR
TDIGM	Rate of change of TIGM, defined at TGRR
APOGEE	Height of apogee
PERIGEE	Height of perigee
INCLINATION	Inclination of orbit plane
INS PHASE	Phase angle at insertion
DN TARGET	Descending node of target
BIAS	Time added to GMTLO* time to obtain a lift-off time
REF	Reference radius
T ANOMALY	True anomaly
ALTITUDE	Height of chaser, at insertion (MECO)
DH	Delta height between chaser and target, at insertion (MECO)
TIME	Greenwich mean time of lift-off

TABLE 4-IV.- Continued

(i) LWP state vector display

PROCESSOR LWP

```
                          VECTOR  AT  INSERTION

                                                                    COORDINATE SYSTEM: TEG
A, E, I, G, H, L, T                                       AEIMA
      .XXXXXXXXXE+XX     .XXXXXXXXXE+XX    .XXXXXXXXXE+XX    .XXXXXXXXXE+XX
     +.XXXXXXXXXE+XX    +.XXXXXXXXXE+XX   +.XXXXXXXXXE+XX

V, GAM, AZ, R, RASC, DECL                                 POLRDC
      .XXXXXXXXXE+XX    +.XXXXXXXXXE+XX   +.XXXXXXXXXE+XX
      .XXXXXXXXXE+XX    +.XXXXXXXXXE+XX   +.XXXXXXXXXE+XX

X, Y, Z, XD, YD, ZD                                       CARXYZ
     +.XXXXXXXXXE+XX    +.XXXXXXXXXE+XX   +.XXXXXXXXXE+XX
     +.XXXXXXXXXE+XX    +.XXXXXXXXXE+XX   +.XXXXXXXXXE+XX

HA, HA, INC, LONG, DECL, TA                               HSPLDC
      .XXXXXXXXXE+XX     .XXXXXXXXXE+XX   +.XXXXXXXXXE+XX
     +.XXXXXXXXXE+XX     .XXXXXXXXXE+XX   +.XXXXXXXXXE+XX

MA, N, PERIOD
      .XXXXXXXXXE+XX    +.XXXXXXXXXE+XX    .XXXXXXXXXE+XX

      .XXXXXXXXE+XX      .XXXXXXXXE+XX
```

TABLE 4-IV.- Concluded

(j) Display parameter definition table for the LWP state vector display

PROCESSOR LWP

ORBITER VECTOR OR TARGET VECTOR AT INSERTION	
Display parameter label	Parameter definition
A	Semimajor axis
E	Eccentricity
I	Inclination of orbit plane
G	Argument of perigee
H	Right ascension of orbit plane ascending node
L	Mean anomaly
V	Inertial velocity magnitude
GAM	Geocentric flight path angle
AZ	Geocentric flight azimuth
R	Radius magnitude
RASC	Right ascension
DECL	declination
X	
Y	Cartesian position components, TEG - in Y direction, TEG
Z	
XD	
YD	Cartesian velocity components, TEG - in Y direction, TEG
ZD	
HA	Apogee altitude above a sperical Earth
HP	Perigee altitude above a sperical Earth
INC	Inclination of orbit plane
LONG	Geographic longitude
DECL	Declination
TA	True anomaly
MA	Mean semimajor axis
N	Mean motion
PERIOD	Inertial period of orbit, min.
ALT	Height of vehicle above reference radius
TINS	Time of insertion (MECO)

TABLE 4-V.- PROCESSOR MESSAGE TABLE

PROCESSOR LWP

MSG no.	Message ID block	Message text block and explanation
1	*LWP*	*LWPIO* INPUT VECTOR IS NOT TEG/CARTESIAN Meaning: The user has input a vector that is not in the true equator and Greenwich meridian of epoch Cartesian coordinate system. Severity: The processor will terminate, and control will be returned to the FDS Executive. Action required by user: Transform the vector to TEG/Cartesian then resume execution.
2	*LWP*	*LWPIO* THE FOLLOWING CODE IS INVALID INPUT: "XX". Meaning: The user has supplied an invalid mnemonic. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to correct the erroneous code; then resume execution.
3	*LWP*	*LWT* IFAIL=1. ITERV TERMINATED. Meaning: ITERV did not converge on finding the inplane lift-off time based on minimum wedge angle at insertion. Severity: Warning only; processor will continue. Action required by user: None
4	*LWP*	*NPLAN* DID NOT CONVERGE. Meaning: NPLAN did not converge on theoretical inplane lift-off time. Severity: Warning only; processor will continue. Action required by user: None.
5	*LWP*	*GMTLS* STAR TABLE FILLED Meaning: User has requested more than the maximum ten star times allowed. Severity: Warning only; processor will continue. Action required by user: None.
6	*LWP*	*GMTLS* DID NOT CONVERGE. K25=NN. Meaning: GMTLS did not converge on a star time. Severity: Warning only; processor will continue. Action required by user: None.

TABLE 4-V.- Concluded

PROCESSOR LWP

MSG no.	Message ID block	Message text block and explanation
7	*LWP*	*RLOT* DESIRED INSERTION CONDITIONS NOT MET. Meaning: RLOT failed to converge on delta-V required to meet altitude requirement. Severity: Warning only; processor will continue. Action required by user: None.
8	*LWP*	*RLOT* NOT CONVERGING ON TYAW. Meaning: RLOT failed to converge on minimum yaw steering lift-off time. Severity: Warning only; processor will continue. Action required by user: None.
9	*LWP*	*LWDSP* INVALID DISPLAY CODE: "XX". Meaning: The user has requested an invalid display code. Severity: Warning only. Action required by user: Input a valid display code.
10	*LWP*	*LWPT* INVALID DISPLAY CODE: "XX". Meaning: The user has requested an invalid display code. Severity: Warning only. Action required by user: Input a valid display code.
11	*LWP*	*RLOTD* INVALID DISPLAY CODE: "XX". Meaning: The user has requested an invalid display code. Severity: Warning only. Action required by user: Input a valid display code.

TABLE 4-VI.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
LWP	The LWP finds the optimum launch lift-off time (minimum payload loss lift-off time) and then calculates the insertion conditions necessary for a chaser vehicle to effect rendezvous with a target vehicle already in orbit.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
GLOCON	Global constants array, normally defaulted to !!GLCN.
SESCON	Session constants array, normally defaulted to !SESCN.
PROCON	LWP constants and tolerances.
TRGVEC	Position/velocity input state vector for target vehicle, standard format.
CCD	Chaser vehicle coefficient of drag.
CAREA	Chaser vehicle reference area.
CWHT	Chaser vehicle weight at insertion (MECO).

TABLE 4-VI.- Continued

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
LW	Launch window/launch targeting options = "LW"; compute LW = "LT"; compute LT = "WT"; compute LW and LT
NS	In-plane launch window opening and closing times option = "OP"; compute opening = "CL"; compute closing = "OC"; compute opening and closing
DAY	Day on which launch window times are computed, relative to base date
LPT	Launch window parameter table options = "NO"; do not generate table = "OP"; around launch window opening = "CL"; around launch window closing = "OC"; around both opening and closing = "EN"; through entire window
TSTART	Delta time prior to in-plane time to start parameter table
TEND	Delta time after in-plane time to stop parameter table
TSTEP	Time step in parameter table

TABLE 4-VI.- Continued

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
STABLE	GMTLO* table flag = "NO"; do not compute table = "YE"; compute
STARS	Delta time prior to each in-plane launch point to start GMTLO* search
STARE	Delta time after each in-plane launch point to end GMTLO* search
LATLS	Geocentric latitude of launch site
LONGLS	Geographic longitude of launch site
PFT	Powered flight time
PFA	Powered flight arc
YSMAX	Yaw steering limit
DTOPT	Delta time to be subtracted from analytical in-plane launch time to obtain empirical in-plane launch time
DTGRR	Delta-t from lift-off, which defines the time of guidance reference release

TABLE 4-VI.- Continued

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
LAZCOE	Launch azimuth coefficients (A0, A1, A2, A3)
RINS	Radius of insertion
VINS	Velocity magnitude at insertion
GAMINS	Flightpath angle at insertion
LOT	Lift-off time options for launch targeting = "GMTLOR"; lift-off on input time = "OFFSET"; compute lift-off time to achieve a desired phase angle at insertion = "RSTAR" ; lift-off on GMTLO* + BIAS (THRESHOLD = GMTLOR) = "PSTAR" ; lift-off on GMTLO* + BIAS (THRESHOLD = TPLANE) = "TPLANE"; lift-off on TPLANE + TRANS = "TYAW" ; lift-off on in-plane launch time based on phase angle
GMTLOR	Recommended or threshold lift-off time
OFFSET	Phase angle desired at insertion
BIAS	Bias time that is added to GMTLO* to produce lift-off time
TPLANE	Time of in-plane lift-off

TABLE 4-VI.- Continued

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
TRANS	Delta time added to in-plane time to obtain lift-off
INSCO	Insertion cutoff conditions option flag = "VGAMR" ; input RINS, VINS, GAMINS = "DHGAMR"; input RINS, GAMINS, DHW, DU, ANOM = "HGAMR:"; input RINS, GAMINS, DHW, DU, ANOM
DHW	Desired height difference (INSCO = "DHGAMR") or altitude (INSCO = "HGAMR")
DU	Angle from insertion to obtain a given altitude or delta altitude
ANOM	Nominal semimajor axis at insertion
DELNOF	Differential nodal regression option flag = "IN"; input DELNO = "CO"; compute DELNO
DELNO	Angle that is added to the target descending node to account for differential nodal regression
NEGTIV	Initial phase angle control flag = "PO"; use positive value = "NE"; use negative value = "LO"; use lowest absolute value

TABLE 4-VI.- Concluded

PROCESSOR LWP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
WRAP	Flag to WRAP initial phase angle = "N"; add $2N\pi$ to phase angle (N is integer)
SVPROP	State vector propagation selection flag = "CON"; conic = "AEG"; analytic ephemeris generator
PVTABT	Target vehicle position/velocity phase table. Two vectors are generated per call to LWP: (1,1) initial time (1,2) insertion time
PVTABC	Chaser vehicle position/velocity phase table. Vector is defined at insertion.
SUMTAB	Output summary table
GMTLO	GMT of chaser vehicle lift-off. Not output when LW = "LW."

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - Main Program LWP

5.1.1 Purpose

The Launch Window Program (LWP) is the main program routine of the Launch Window Processor. As such, it directs the computational flow of the processor, dependent upon user requests.

5.1.2 Functional Description

The Launch Window Processor is too large to reside entirely in core at one time. Consequently, the function of LWP is to control the computational flow of the processor, and to load into core each segment of the processor as it is needed. It also sets up the initialization and finalization functions required of each processor.

5.1.3 Assumptions and Limitations

None.

5.1.4 Method

None.

5.1.5 Routine Input/Output Variables

The input/output variables are presented in Table 5.1-I.

5.1.6 Functional Logic Flow

The functional logic flow is presented in Figure 5.1-1.

5.1.7 Diagnostics and Debug

None.

5.1.8 Special Comments

None.

5.1.9 References

None.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKB	--	Real	I	--	C	--	Parameters array
BLKC	--	Real	I	--	C	--	Parameters array
CONST	--	Real	I	--	C	--	Earth constants array
IPRAM	--	Intg	I/O	--	A	--	System parameters
LW	--	Intg	I	--	C	LW	Launch window/launch targeting options
LPT	--	Intg	I	--	C	LPT	Launch window parameter table options
RT	$\bar{R}_T$	Real	I/O	ft	C	PVTAB	Target position vector
SESCON	--	Free	I	--	C	!SESCN	Session constants
TRGVEC	--	Real	I	--	C	TRGVEC	Position/velocity input state vector for target vehicle
TT	T_T	Real	I/O	sec	C	PVTAB	Time of target state vector
VT	$\bar{V}_T$	Real	I/O	fps	C	PVTAB	Target velocity vector

NOTES:

TYPE
Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.1-I.- Concluded

Routine LWP

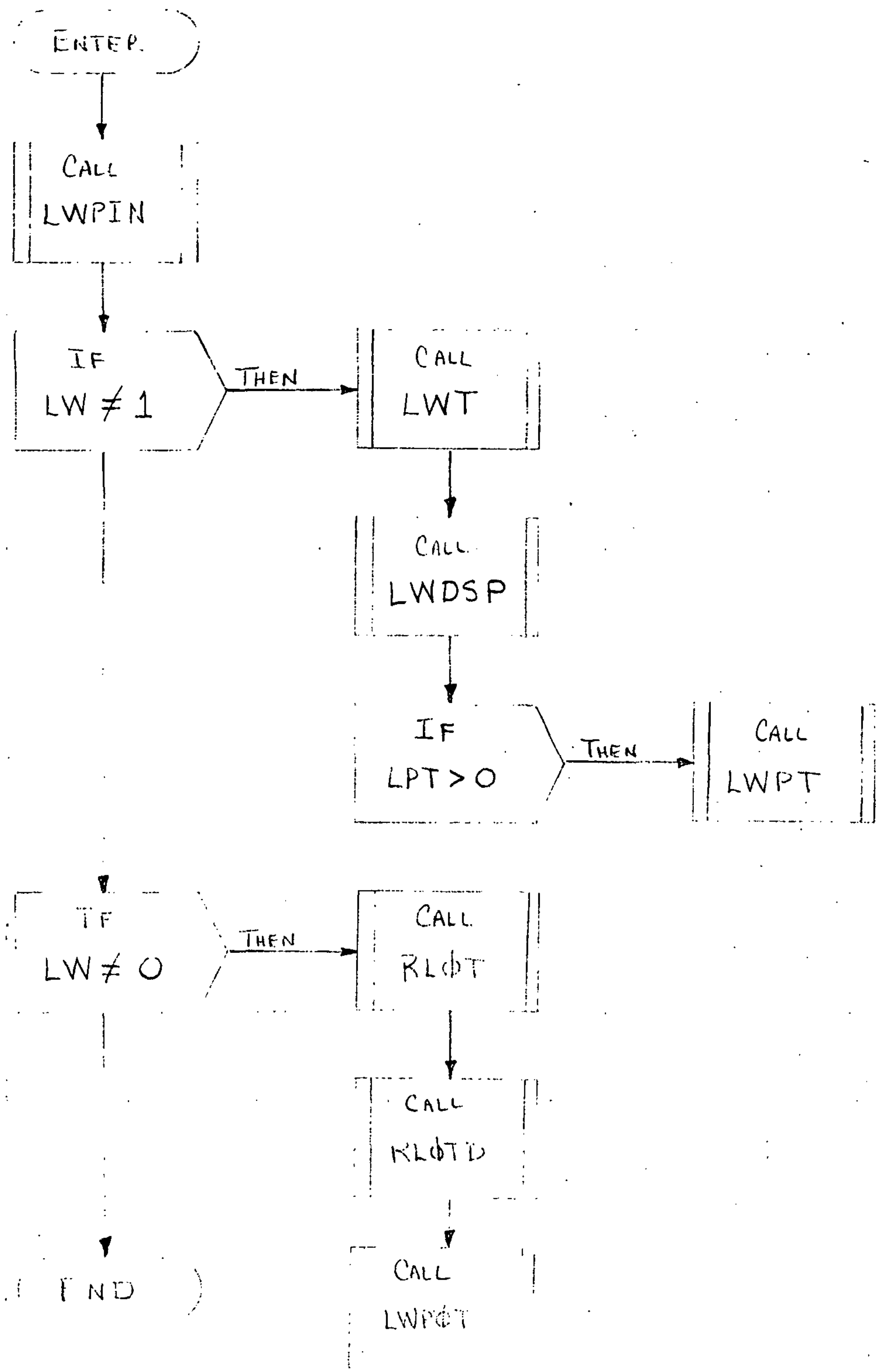
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LU	--	Intg	O	--	A	IPRAM(1)	User's logical unit number
RP	$\bar{R}_P$	Real	O	ft	C	PVTAB	Chaser position vector
TP	T_P	Real	O	sec	C	PVTAB	Time of chaser state vector
VP	$\bar{V}_P$	Real	O	fps	C	PVTAB	Chaser velocity vector
ERROR	--	Intg	I	--	C	--	Processor error return flag

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.1-1.- LWP functional logic flow.

5.2 ROUTINE NAME - Subroutine LWPIN

5.2.1 Purpose

LWPIN is the Launch Window Processor data input routine.

5.2.2 Functional Description

LWPIN calls the system routine XPGET to obtain the user supplied input quantities necessary to run LWP. Then, the input data is interpreted, converted to internal units, and assigned to the appropriate common locations for execution. If the input state vector is found not to be True-Equatorial-Greenwich (TEG), LWPIN sets the error return code to -32768 prior to exiting the routine.

5.2.3 Assumptions and Limitations

None.

5.2.4 Method

None.

5.2.5 Routine Input/Output Variables

The input/output variables are presented in Table 5.2.I.

5.2.6 Functional Logic Flow

The functional logic flow is presented in Figure 5.2-1.

5.2.7 Diagnostics and Debug

When an inappropriate parameter code for the state vector coordinate system is encountered, LWPIN sends a message to the user, sets the appropriate error return code, and exits the routine.

If the processor debug flag is set, LWPIN writes the launch window/launch targeting flag, the input target state vector, and various other input parameters on the user specified logical unit.

5.2.8 Special Comments

None.

5.2.9 References

None.

TABLE 5.2-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWPIN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LU	--	Intg	I	--	C	IPRAM(1)	Logical unit number
INTERFACE TABLE VARIABLES	--	--	I	--	IT	--	See table 4-I of this processor for definition of contents
A0, A1 A2, A3	A_0, A_1 A_2, A_3	Real	O	--	C	--	Launch azimuth coefficients
A5	A_5	Real	O	rad	C	--	Angle from insertion
AREA	--	Real	O	ft^2	C	--	Target vehicle reference area
CD	C_D	Real	O	--	C	--	Target vehicle coefficient of drag
DET	--	Real	O	sec	C	--	Time iteration tolerance
DELH	--	Real	O	ft	C	--	Radius iteration tolerance
DOS	--	Real	O	rad	C	--	Angular iteration tolerance
DVTOL	--	Real	O	fps	C	--	Delta velocity iteration tolerance

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.2-I.- Continued

Routine LWPIN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
ERROR	--	Intg	O	--	C	--	Input error return flag
GMTLOR	--	Real	O	sec	C	--	Recommended lift-off time
IBUG	--	Intg	O	--	C	--	Output unit on which to write print
INSCO	--	Intg	O	--	C	--	Insertion cutoff conditions flag
K70	--	Intg	O	--	C	--	Flag to wrap initial phase angle
LOT	--	Intg	O	--	C	--	Lift-off time option
NPRINT	--	Intg	O	--	C	--	Internal print flag
RAD	--	Real	O	deg/rad	C	--	Radians to degrees conversion factor
SPERT	--	Intg	O	--	C	--	State vector propagation flag
TLO	--	Real	O	sec	C	--	GMT of target vehicle lift-off
TPLANE	--	Real	O	sec	C	--	GMT of in-plane lift-off

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.2-I.- Concluded

Routine LWPIN

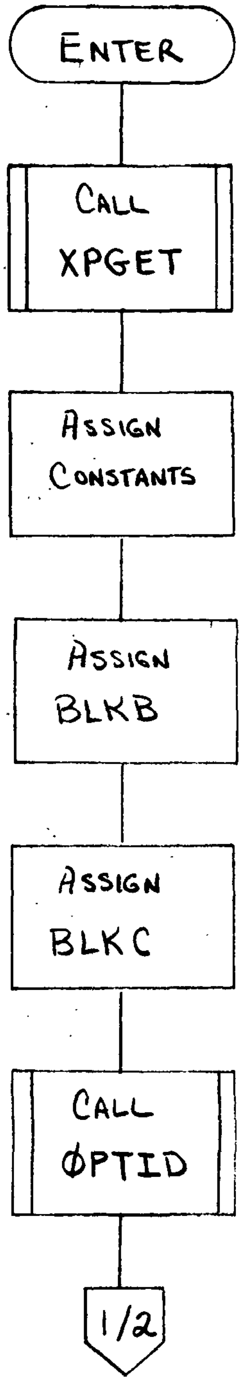
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RT	$\bar{R}_T$	Real	O	ft	C	--	Target vehicle position vector
VT	$\bar{V}_T$	Real	O	fps	C	--	Target vehicle velocity vector
TT	T_T	Real	O	sec	C	--	Target vehicle vector time
WHT	--	Real	O	lb	C	--	Target vehicle weight

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

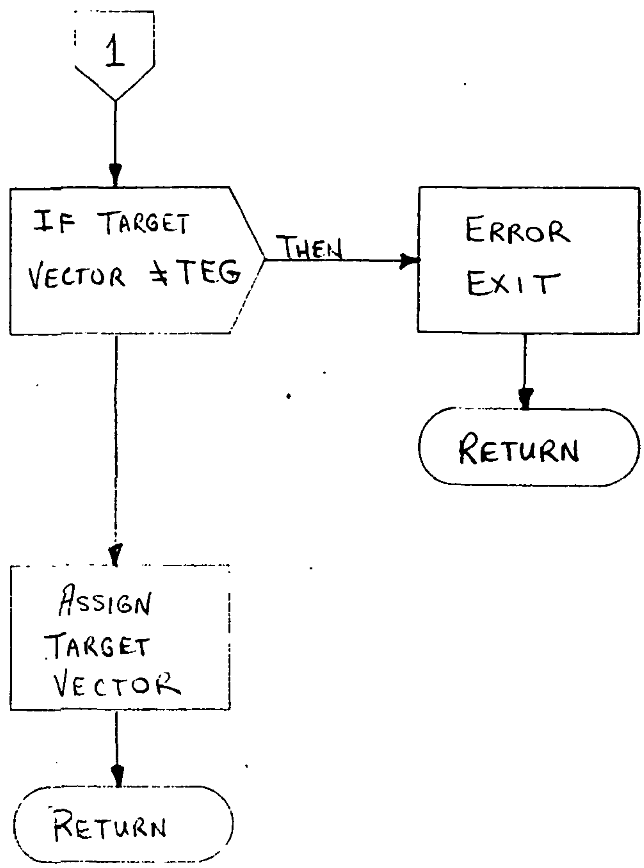
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 2

Figure 5.2-1.- LWPIN functional logic flow.



Page 2 of 2

Figure 5.2-1.- Concluded.

5.3 ROUTINE NAME - Subroutine OPTID

5.3.1 Purpose

The Options Identification (OPTID) subroutine interprets and assigns a numerical value to each parameter that requires ASCII input by the user.

5.3.2 Functional Description

There are nine parameters in the LWP interface table that require ASCII input by the user. OPTID is called sequentially by the routine LWPIN, using a routing flag, to have each parameter interpreted. When the input is recognized, a numerical value is assigned the parameter, which subsequently will be used by LWP for proper program routing.

5.3.3 Assumptions and Limitations

None.

5.3.4 Method

The correct user responses for each parameter are stored in data statements within OPTID. Each user response is compared, bit-by-bit, using the system routine XRCPR. When a match occurs, an appropriate numerical value is assigned to the parameter. If no match occurs, the user is informed of an input error, the error return code flag is set, and the processor LWP is terminated.

5.3.5 Routine Input/Output Variables

The OPTID input/output variables are presented in Table 5.3-I.

5.3.6 Functional Logic Flow

The functional logic flow is presented in Figure 5.3-1.

5.3.7 Diagnostics and Debug

None.

5.3.8 Special Comments

None.

5.3.9 References

None.

TABLE 5.3-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine OPTID

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
ROUTE	--	Intg	I	--	A	--	Routing flag
ID	--	2CH	I	--	A	--	Maneuver code
NUMBER	--	Intg	O	--	A	--	Numerical value for option code
ERROR	--	Intg	O	--	A	--	Error return code

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

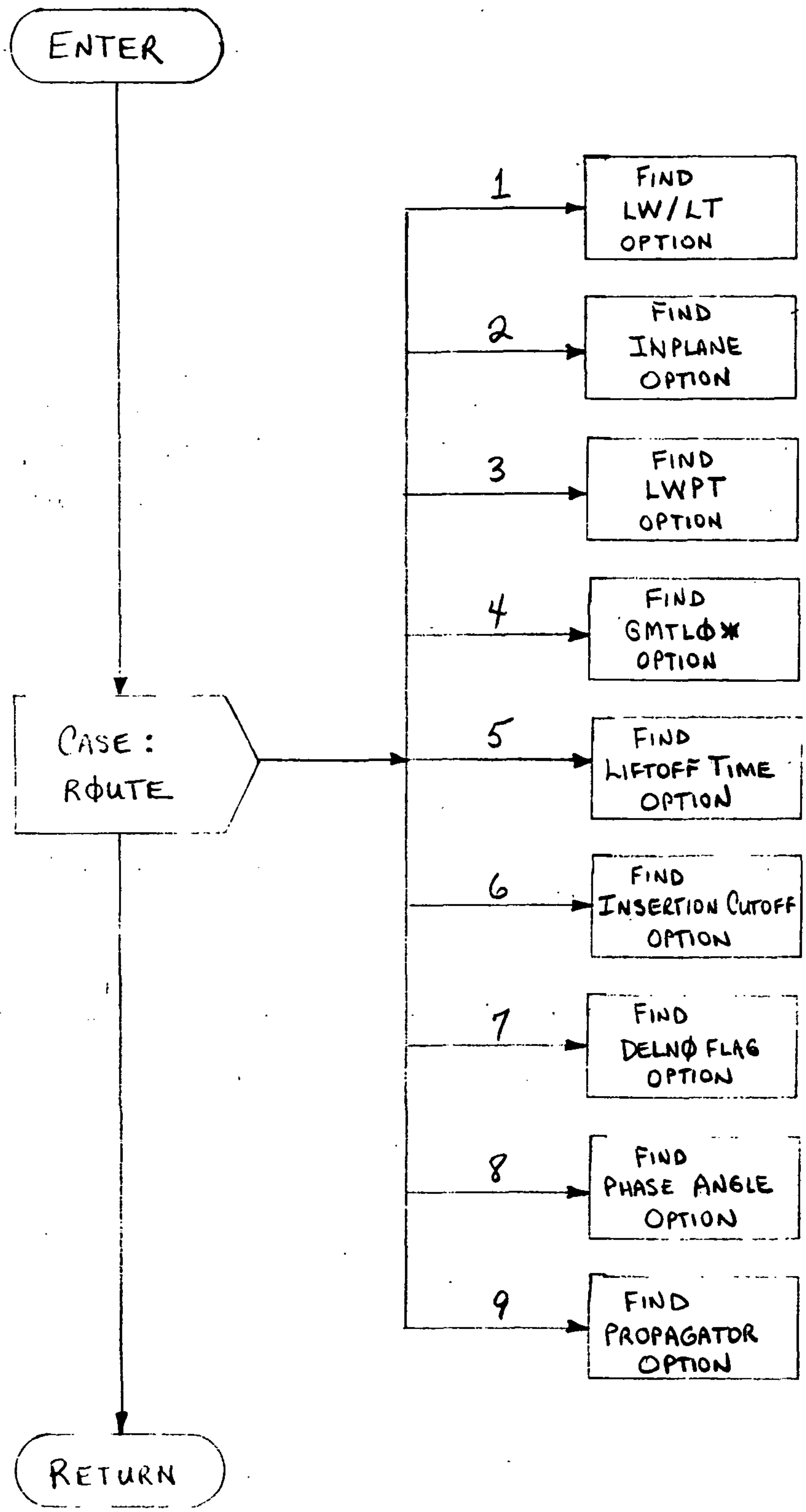


Figure 5.3-1.- OPTID functional logic flow.

5.4 ROUTINE NAME - Subroutine LWT

5.4.1 Purpose

The Launch Window Time (LWT) routine finds the analytical inplane launch time, then adjusts this time by an input value to determine the optimum launch lift-off time (minimum payload loss lift-off time).

5.4.2 Functional Description

Given an in-orbit target vehicle state vector, the LWT calculates the inplane time of the launch site with the target orbit plane. First, LWT finds the analytical inplane time. This time occurs when the wedge angle at insertion between the chaser and target vehicles' planes is zero. Then, this time is adjusted by an input value to determine the optimum launch lift-off time. LWT can, upon user request, find the northerly inplane time, the southerly inplane time, or both.

Within LWT, the GMTLO* time (i.e., phase match or zero phase angle at insertion point) table may be computed. This is accomplished by a call to the subroutine GMTLS.

5.4.3 Assumptions and Limitations

The LWT will compute the inplane times for one launch day at a time.

The input parameter DTOPT is needed to adjust the analytical solution for the inplane launch time to the integrated launch trajectory inplane launch time. DTOPT is used in computing the northerly and the southerly inplane launch points.

The value of the empirical inplane time (TPLANE) is dependent upon the value of the launch window flag (NS). If N=0 or NS=2, TPLANE is set to the opening inplane time. If NS=1, TPLANE is set to the closing inplane launch time.

5.4.4 Method

In order to find the inplane launch time, LWT first computes the wedge angle at insertion between the chaser and target planes for a given lift-off time. The delta velocity required for a plane change is then computed, using

$$\Delta V = 2V \sin(\delta/2),$$

where V = inertial velocity magnitude at insertion of chaser

δ = wedge angle.

If $|\Delta V|$ < tolerance, the analytical inplane point (TIP) is found. If not, the utility subroutine ITERV calculates a new lift-off time, and a new wedge angle is calculated. The empirical inplane time (TPLANE) is then given by TPLANE = TIP - DTOPT.

5.4.5 Routine Input/Output Variables

The LWT input/output variables are presented in Table 5.4-I.

5.4.6 Functional Logic Flow

The LWT functional logic flow is presented in Figure 5.4-1.

5.4.7 Diagnostics and Debug

If the internal print flag is set, the initial target state vector, the day of liftoff, and the first guess analytical inplane time are listed. Also, the wedge angle and plane change delta velocity are listed on each iteration through ITERV. And, the final inplane launch time and insertion time are listed.

5.4.8 Special Comments

None.

5.4.9 References

None.

TABLE 5.4-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKB	--	Real	I/O	--	C	--	Parameters array
BLKC	--	Real	I/O	--	C	--	Parameters array
RT	$\bar{R}_T$	Real	I/O	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I/O	fps	C	--	Target velocity vector
TT	T_T	Real	I/O	sec	C	--	Target vector time
CMAX	--	Intg	I	--	C	--	Maximum iteration tolerance
DX1	--	Real	I	--	C	--	First guess for independent variable stepsize
CLOSE	--	Real	O	sec	C	--	Time of lift-off for launch window closing
ERROR	--	Intg	O	--	C	--	Error return code
K25	--	Intg	O	--	C	--	Number of GMTLO* times computed
OPEN	--	Real	O	sec	C	--	Time of lift-off for launch window opening

NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.4-I.- Concluded

Routine LWT

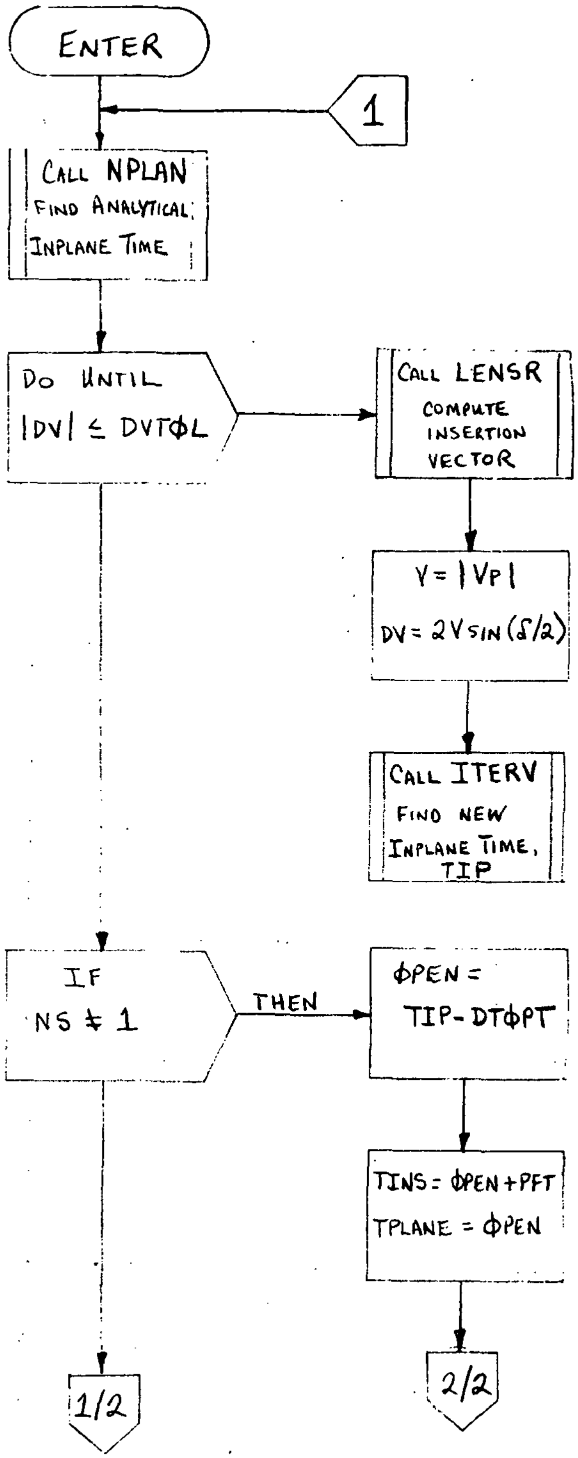
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
PAC	—	Real	O	rad	C	—	Phase angle at window closing
PAO	--	Real	O	rad	C	--	Phase angle at window opening
TINS	--	Real	O	sec	C	--	Time of insertion
TPLANE	--	Real	O	sec	C	--	Time of in-plane lift-off
WEDGE	δ	Real	O	rad	C	—	Wedge angle between planes

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

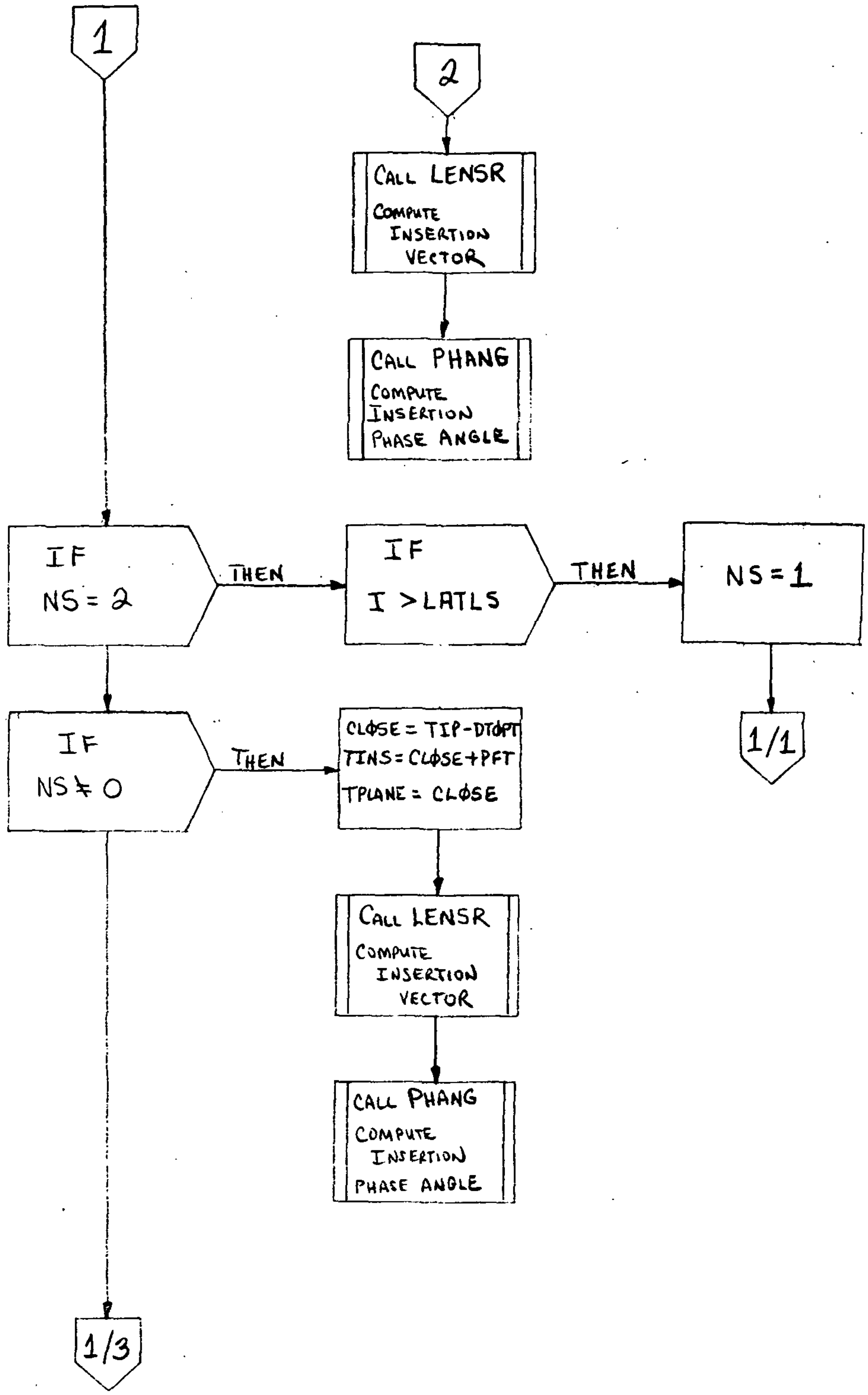
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



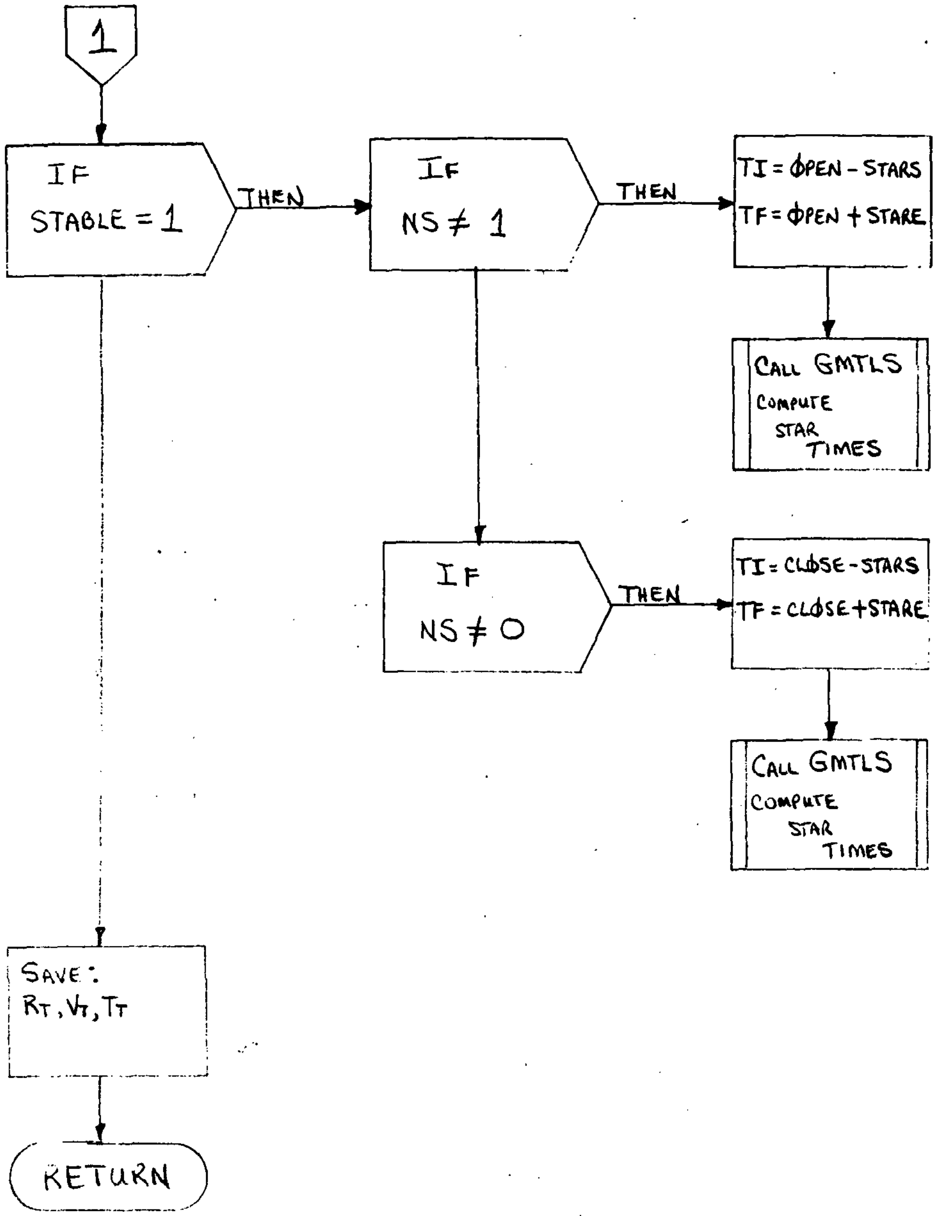
Page 1 of 3

Figure 5.4-1.- LWT functional logic flow.



Page 2 of 3

Figure 5.4-1.- Continued.



Page 3 of 3

Figure 5.4-1.- Concluded.

5.5 ROUTINE NAME - Subroutine NPLAN

5.5.1 Purpose

The inplane launch time routine (NPLAN) computes the analytical inplane launch time.

5.5.2 Functional Description

Given a first guess at the inplane time, NPLAN uses spherical trigonometry to find the time at which the launch site is in the plane of the target vehicle.

5.5.3 Assumptions and Limitations

None.

5.5.4 Method

The objective of NPLAN is to find the time at which the launch site is inplane with the target plane. This is accomplished by first finding the argument of latitude of the target vehicle at the launch site. Thus, using triangle BDC in figure 5.5-I,

$$\cos(90^\circ - \phi_{LS}) = \cos U_T \cos 90^\circ + \sin U_T \sin 90^\circ \cos(90^\circ - i_m)$$

or

$$U_T = \sin^{-1} (\sin \phi_{LS}/\sin i_m).$$

where ϕ_{LS} = geocentric latitude of launch site,

i_m = mean inclination.

Also, find the angle between the ascending node and the projection of the target vehicle position on the equatorial plane. Using triangle ABC in Figure 5.5-1,

$$\cos U_T = \cos \phi_{LS} \cos \eta + \sin \phi_{LS} \sin \eta \cos 90^\circ$$

or

$$\eta = \cos^{-1} (\cos U_T/\cos \phi_{LS}).$$

Finally, find the geographic longitude of the ascending node at the time T_{IP}, and then the geographic longitude of the target plane at the argument of latitude U_T. Thus,

$$\lambda_N = H - \omega_e \cdot T_{IP},$$

where H = right ascension of target

ω_e = Earth's rotation rate

And, $\lambda = \lambda_N + \eta$.

If $|\lambda - \lambda_{LS}| <$ tolerance, where λ_{LS} is the geographic longitude of the launch site, then the computations are complete, and T_{IP}, the time of analytical inplane launch, is found. If the tolerance has not been met, then set

$$T_{IP} = T_{IP} + (\lambda - \lambda_{LS})/(\omega_e - \dot{H}),$$

where $\dot{H}$ = the nodal precession rate, and repeat the calculations for the new T_{IP}.

Math symbol	Internal code symbol	Math symbol	Internal code symbol
U_T	UT	ϕ_{LS}	LATLS
i_m	MI	Π	PI
η	ETA	λ_N	LAMBDA
λ	LONG	T_{IP}	TIP
ω_e	WE	H	H
$\dot{H}$	HDOT	λ_{LS}	LONGLS

5.5.5 Routine Input/Output Variables

The NPLAN input/output variables are presented in Table 5.5-I.

5.5.6 Functional Flow Chart

The NPLAN functional flow chart is presented in Figure 5.5-2.

5.5.7 Diagnostics and Debug

When the internal print flag is set, the various spherical trigonometry parameters are listed at each iteration so that the condition of the convergence may be observed.

5.5.8 Special Comments

Note that T_{IP} is the time at which the launch site is inplane with the target orbit. This is an analytical inplane launch time. The routine LWT uses this time as its first guess, iterating on the plane change delta velocity required at insertion to find the final analytical time.

5.5.9 References

None.

TABLE 5.5-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine NPLAN

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TIP	T_{IP}	Real	I/O	sec	A	--	Analytical in-plane time
BLKC	--	Real	I/O	--	C	--	Parameters array
BLKB	--	Real	I	--	C	--	Parameters array
RT	$\bar{R}_T$	Real	I/O	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I/O	fps	C	--	Target velocity vector
TT	T_T	Real	I/O	sec	C	--	Target vector time
ERROR	--	Intg	O	--	C	--	Error return code

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

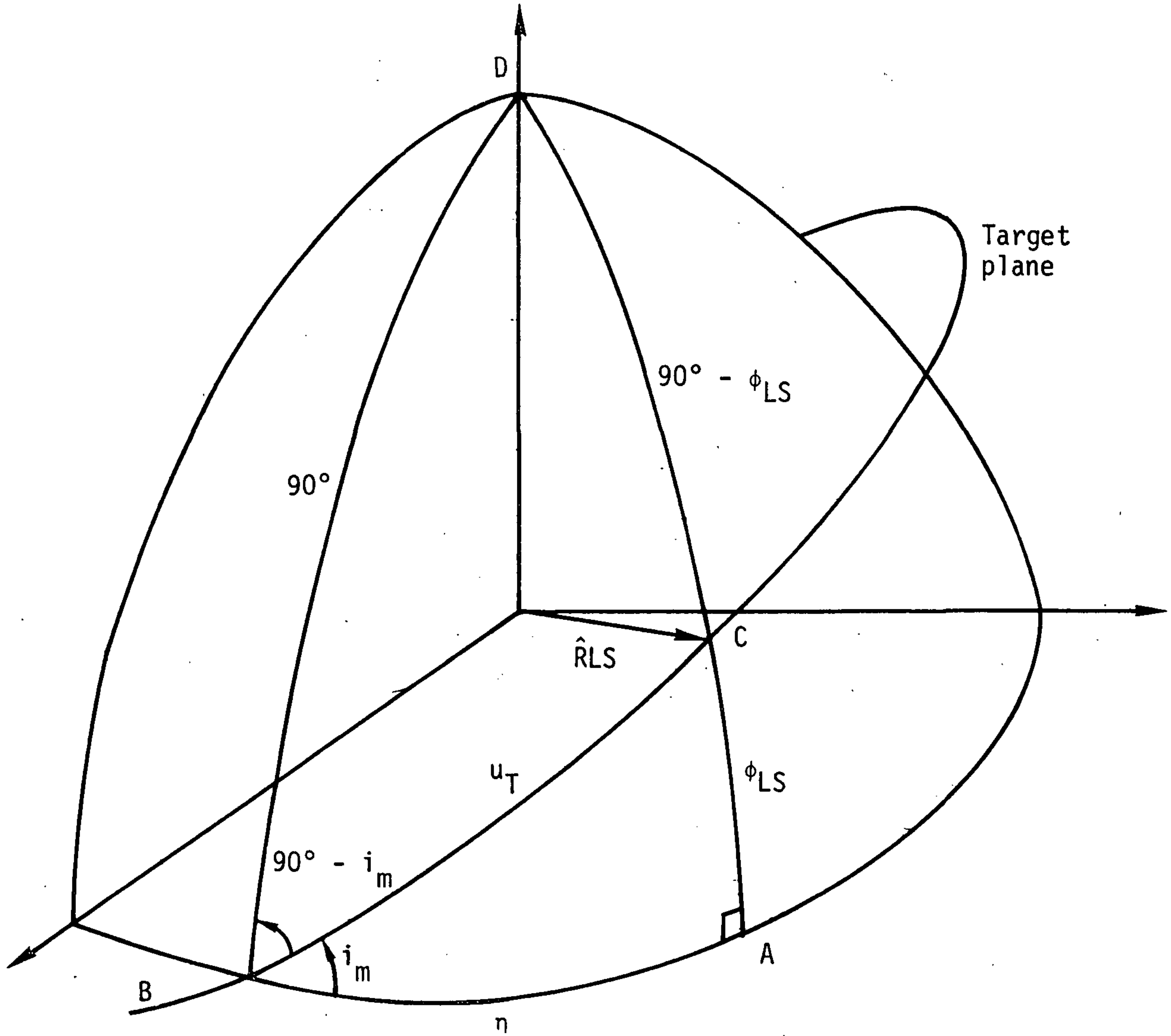
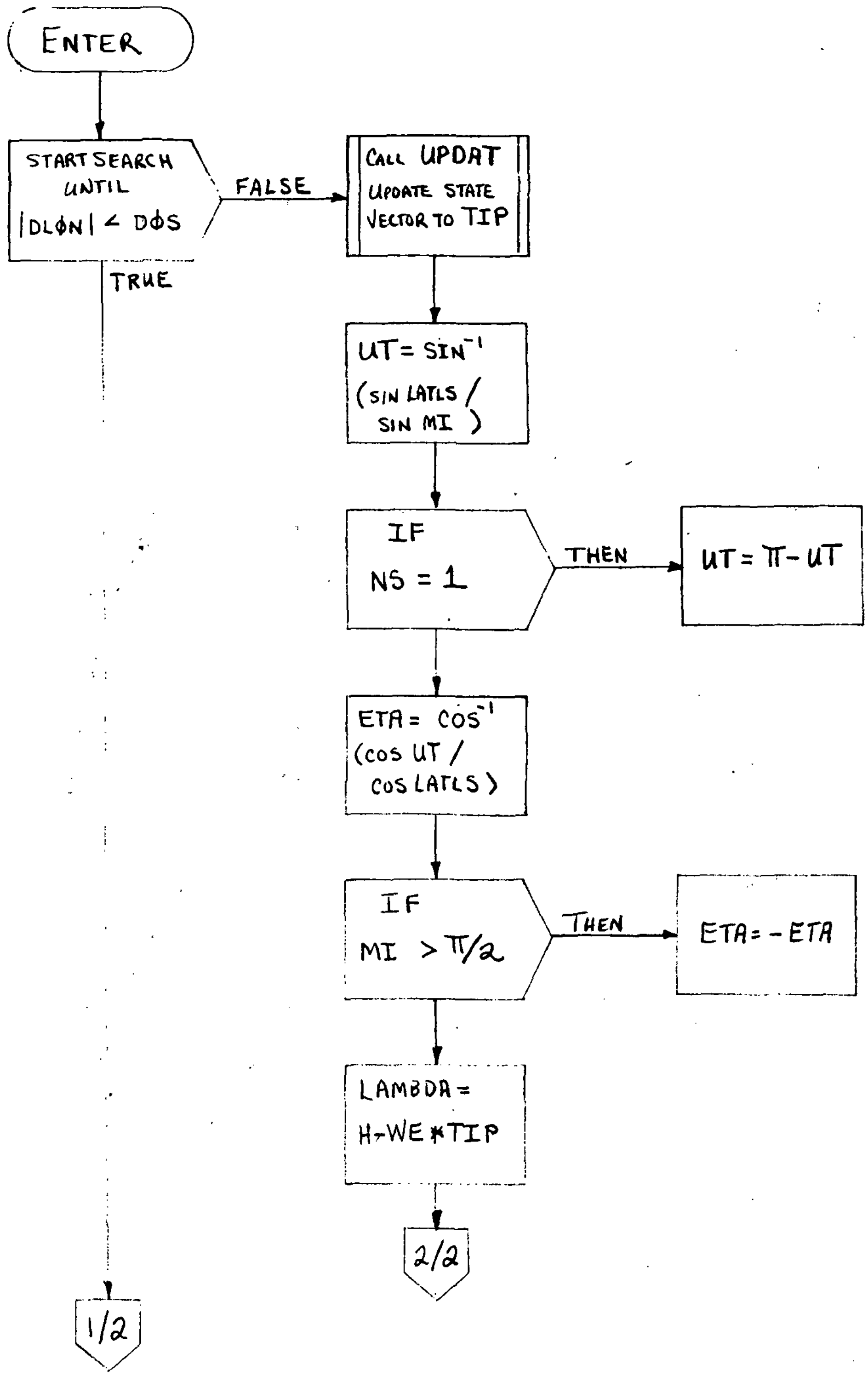


Figure 5.5-1.- NPLAN geometry.



Page 1 of 2

Figure 5.5-2.- NPLAN functional logic flow.

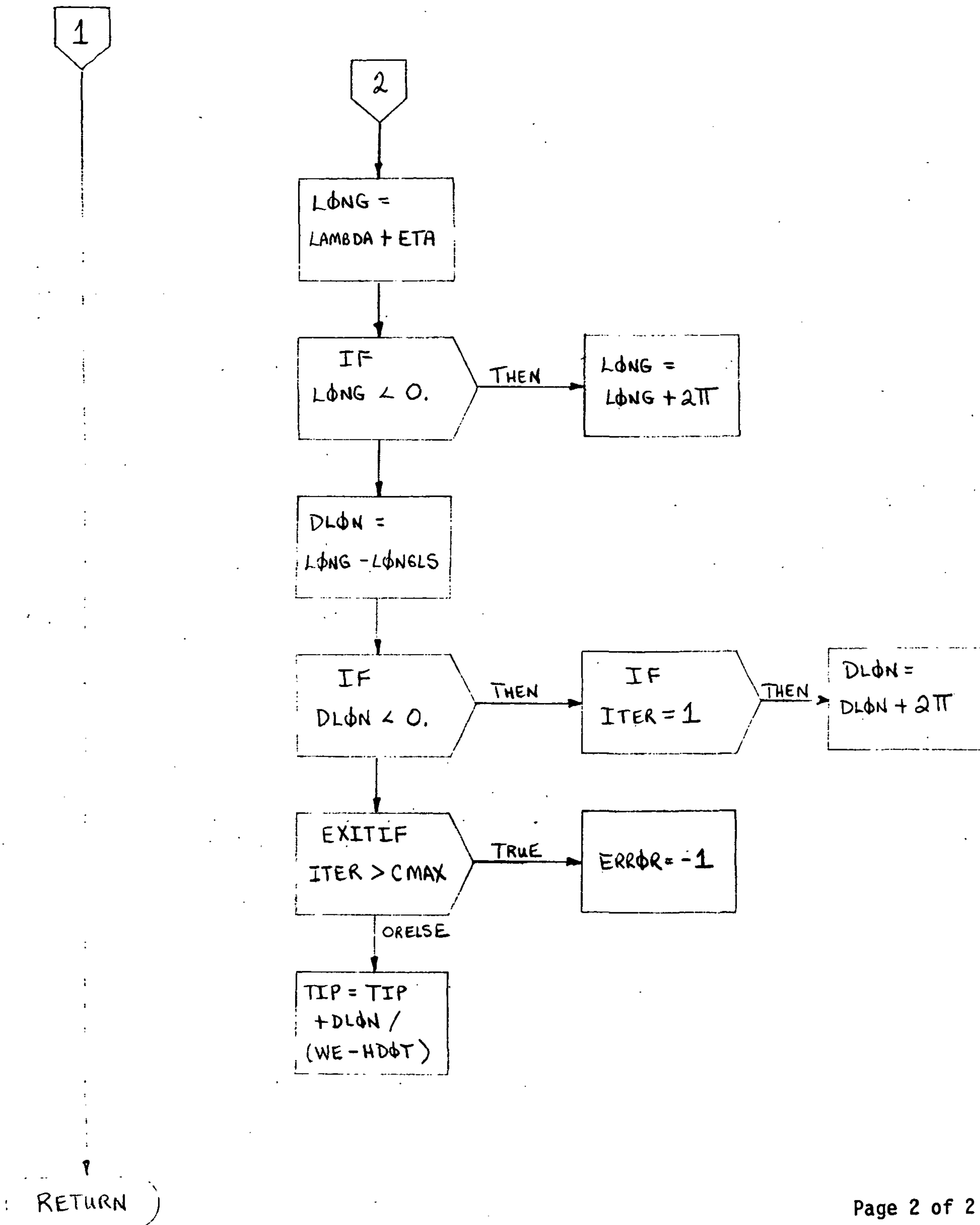


Figure 5.5-2.- Concluded.

5.6 ROUTINE NAME - Subroutine LENSR

5.6.1 Purpose

The purpose of LENSR is to analytically compute an insertion vector for the chaser vehicle using an in-orbit target vehicle.

5.6.2 Functional Description

Given an in-orbit target vehicle as input, LENSR uses analytical equations to compute a chaser vehicle insertion vector. Orbital geometry quantities such as launch site location, powered flight arc and time, velocity, flight path angle, and radius at insertion are used to simulate a parallel launch (if wedge angle is greater than the yaw steering capability) or an inplane launch (if yaw steering is greater than wedge angle). In addition, the wedge angle is given as an output quantity.

5.6.3 Assumptions and Limitations

LENSR will always place the chaser vehicle inplane if the launch vehicle has the yaw steering capability, or parallel to the target plane if the launch vehicle does not have the yaw steering capability. This is fine for use in LWT. But in RLOT, where the insertion phase angle may require an out-of-plane launch (because of differential nodal precession), the more sophisticated targeting scheme found in the subroutine TARGT is used.

5.6.4 Method

LENSR applies orbital geometry knowledge (i.e., the relationship between the target vehicle plane and the location of the launch site) to compute the chaser insertion vector. Needed as input are the following quantities:

$\bar{R}_T, \bar{V}_T, T_T$ - target vehicle state vector

GMTLO - GMT of chaser vehicle lift-off

λ_{LS} - geographic longitude of launch site

ϕ_{LS} - geocentric latitude of launch site

R_{INS} - radius magnitude at insertion

V_{INS} - inertial velocity magnitude at insertion

γ_{INS} - flight path angle at insertion

θ_{PF} - powered flight arc

T_{PF} - powered flight time

YS_{MAX} - maximum yaw steering angle

ω_e - Earth's rotation rate

First, define the unitized radius of launch site vector:

$$\bar{U}_{RLS}(1) = \cos \phi_{LS} \cos \lambda$$

$$\bar{U}_{RLS}(2) = \cos \phi_{LS} \sin \lambda$$

$$\bar{U}_{RLS}(3) = \sin \phi_{LS}$$

where $\lambda = \lambda_{LS} + \omega_e \cdot GMTLO$

Next, define a $\bar{J} - \bar{K} - \bar{H}$ coordinate system where the $\bar{J} - \bar{K}$ plane and the target orbit plane are co-planar.

$$\bar{H} = (\bar{R}_T \times \bar{V}_T)/(|\bar{R}_T \times \bar{V}_T|)$$

$$\bar{K} = (\bar{H} \times \bar{U}_{RLS})/|\bar{H} \times \bar{U}_{RLS}|)$$

$$\bar{J} = \bar{K} \times \bar{H}$$

Now, when a vehicle is launched toward the plane of a target vehicle already in orbit and is placed into a plane that is parallel to the target plane at the time of insertion (i.e., the cross velocity is zero), the chaser plane will intersect the target plane at a travel angle of 90 degrees from the insertion point (see reference 1). Assume a parallel launch, define the orbital geometry as depicted in Figure 5.6-1, and compute ψ (the angle between the launch site vector and the target plane), δ (the angle between the target and chaser planes at the time of insertion), and Γ (the projection of the powered flight are onto the target orbit plane). Using the Law of Cosines and Figures 5.6-2 and 5.6-3,

$$\psi = \sin^{-1}(\bar{H} \cdot \bar{U}_{RLS})$$

$$\delta = \sin^{-1}(\sin \psi / \cos \Theta_{PF})$$

and

$$\Gamma = \sin^{-1}(\sin \Theta_{PF} / \cos \psi)$$

Compute the difference between the wedge angle and the yaw steering maximum,

$$DYAW = |\delta| - YS_{MAX}$$

If $YS_{MAX} \leq 0.$, the parallel plane angles are defined.

If $YS_{MAX} > 0.$, the angles need to be redefined. Thus,

if $DYAW \leq 0.$, using Figures 5.6.4 and 5.6-5,

$$\Gamma = \cos^{-1}(\cos \Theta_{PF}/\cos \psi)$$

and the wedge angle is zero. Or, if DYAW > 0., using figures 5.6-6, 5.6-7, 5.6-8, and 5.6-9, respectively, compute

$$X = \cos^{-1}(\cos \Theta_{PF}/\cos YS_{MAX})$$

$$\delta = |\sin^{-1}(\sin \psi/\cos X)| - YS_{MAX}$$

$$\sin \Gamma = \sin X/\cos \psi$$

and

$$\Gamma = \cos^{-1}|(1. - \sin^2 \Gamma)^{1/2}|$$

Rotate the $\bar{J} - \bar{K} - \bar{H}$ coordinate system through Γ, δ and γ_{INS}, respectively. Thus, rotating through the projection Γ

$$\begin{bmatrix} \bar{J} \\ \bar{K} \\ \bar{H} \end{bmatrix} = \begin{bmatrix} \cos \Gamma & \sin \Gamma & 0 \\ -\sin \Gamma & \cos \Gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{J} \\ \bar{K} \\ \bar{H} \end{bmatrix}$$

If DYAW > 0., then rotating through the wedge angle δ

$$\begin{bmatrix} \bar{J} \\ \bar{K} \\ \bar{H} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 & \sin \phi & \bar{J} \\ 0 & 1 & 0 & \bar{K} \\ -\sin \phi & 0 & \cos \phi & \bar{H} \end{bmatrix}$$

where

$$\phi = (\psi/|\psi|)\delta$$

Finally, rotating through the flight path angle γ_{INS}

$$\begin{bmatrix} \bar{J}' \\ \bar{K}' \\ \bar{H}' \end{bmatrix} = \begin{bmatrix} \cos\gamma_{INS} & -\sin\gamma_{INS} & 0 \\ \sin\gamma_{INS} & \cos\gamma_{INS} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{J} \\ \bar{K} \\ \bar{H} \end{bmatrix}$$

Now, the chaser insertion position vector, $\bar{R}_P$, may be defined in the $\bar{J} - \bar{K} - \bar{H}$ coordinate system. Thus,

$$\bar{R}_P = R_{INS}\,\bar{J}$$

And, the chaser insertion velocity vector, $\bar{V}_P$, may be defined in the $\bar{J}' - \bar{K}' - \bar{H}'$ coordinate system. Thus,

$$\bar{V}_P = V_{INS}\,\bar{K}'$$

The vector time is

$$T_P = GMTLO + T_{PF}$$

Math Symbol	Internal Code Symbol	Math Symbol	Internal Code Symbol
$\bar{R}_T$	RT	$\bar{V}_T$	VT
T_T	TT	γ_{INS}	GAMINS
ϕ_{LS}	LATLS	λ_{LS}	LONGLS
θ_{PF}	PFA	T_{PF}	PFT
R_{INS}	RINS	V_{INS}	VINS
YS_{MAX}	YSMAX	λ	LONG
ω_e	WE	$\bar{U}_{RLS}$	URLS
$\bar{H}$	UNITH	$\bar{J}$	UNITJ
$\bar{K}$	UNITK	ψ	PSI
δ	WEDGE	Γ	BGAMM
DYAW	DELYAW	ϕ	PHI
$\bar{R}_P$	RP	$\bar{V}_P$	VP
$\bar{T}_P$	TP		

5.6.5 Routine Input/Output Variables

The LENSR input/output variables are presented in Table 5.6-I.

5.6.6 Functional Logic Flow

The LENSR functional logic flow is presented in Figure 5.6-10.

5.6.7 Diagnostics and Debug

If the internal print flag is set, the target position and velocity vectors, the chaser position and velocity vectors, and the wedge angle are printed out.

5.6.8 Special Comments

None.

5.6.9 References

1. Milstead, A. H.: Launch Windows for Orbital Missions. Aerospace Corporation Report No. TDR-269 (4550-10)-6, April 1, 1964.

2. Sullivan, W. A.: Logic and Equations for the Real-Time Computation of the Lunar Module Launch Window and Recommended Lift-Off Time. MSC IN 68-FM-5, January 5, 1968.

TABLE 5.6-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LENSR

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GMTLO	--	Real	I	sec	A	--	GMT of lift-off
BLKB	--	Real	I	—	C	--	Processor array
RT	$\bar{R}_T$	Real	I/O	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I/O	fps	C	--	Target velocity vector
TT	T_T	Real	I/O	sec	C	--	Target vector time
GAMINS	γ_{INS}	Real	I	rad	C	--	Flightpath angle at insertion
LATLS	ϕ_{LS}	Real	I	rad	C	--	Geocentric latitude of launch site
LONGLS	λ_{LS}	Real	I	rad	C	--	Geographic longitude of launch site
PFA	ϑ_{PF}	Real	I	rad	C	--	Powered flight arc
PFT	T_{PF}	Real	I	sec	C	--	Powered flight time
RINS	R_{INS}	Real	I	ft	C	--	Radius of insertion
VINS	V_{INS}	Real	I	fps	C	--	Velocity magnitude at insertion

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.6-I.- Concluded

Routine LENSR

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
YSMAX	YS_{MAX}	Real	I	rad	C	--	Yaw steering maximum
RP	$\bar{R}_P$	Real	O	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	O	fps	C	--	Chaser velocity vector
TP	T_P	Real	O	sec	C	--	Chaser vector time

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

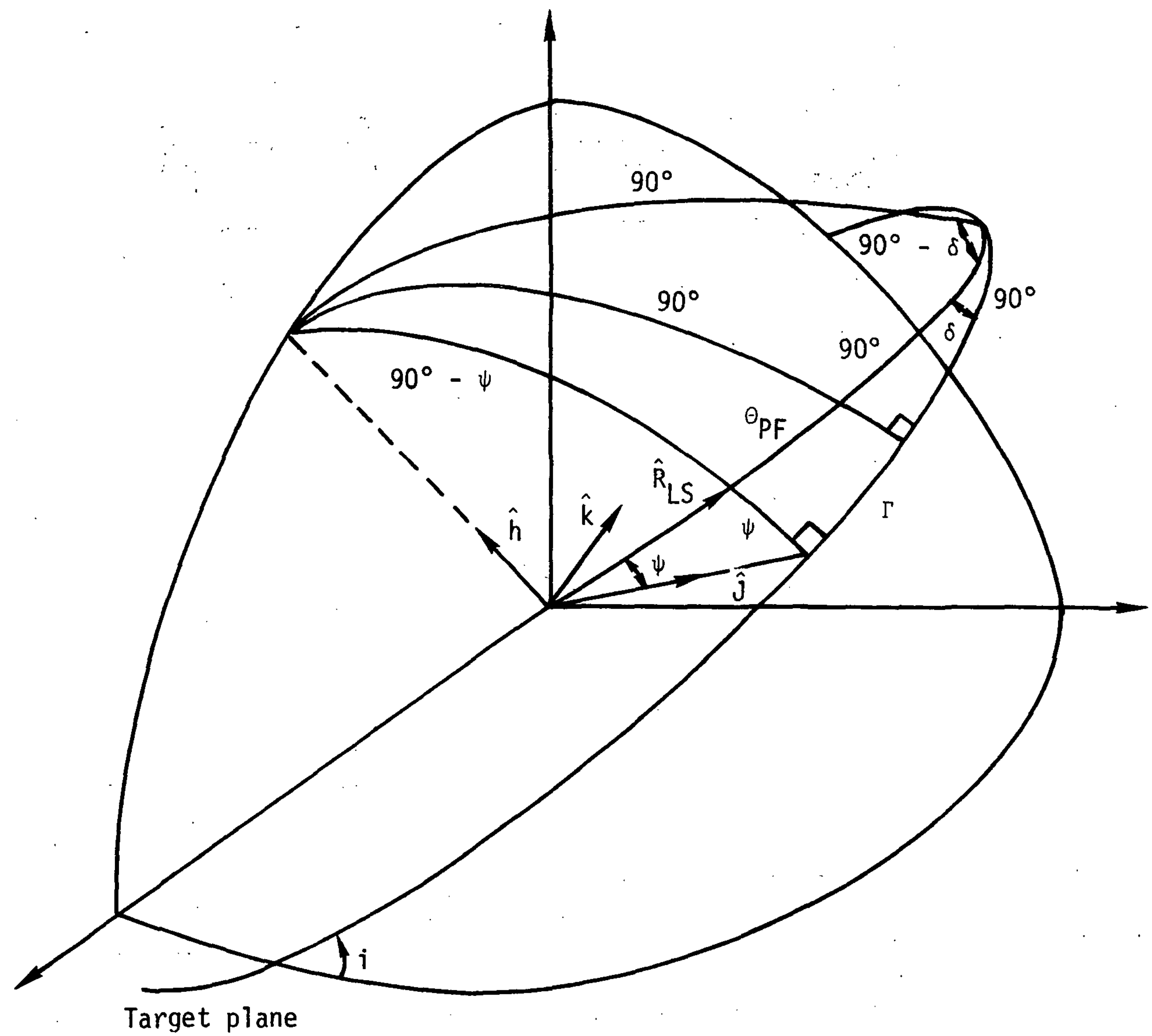


Figure 5.6-1.- Parallel launch geometry (no yaw steering).

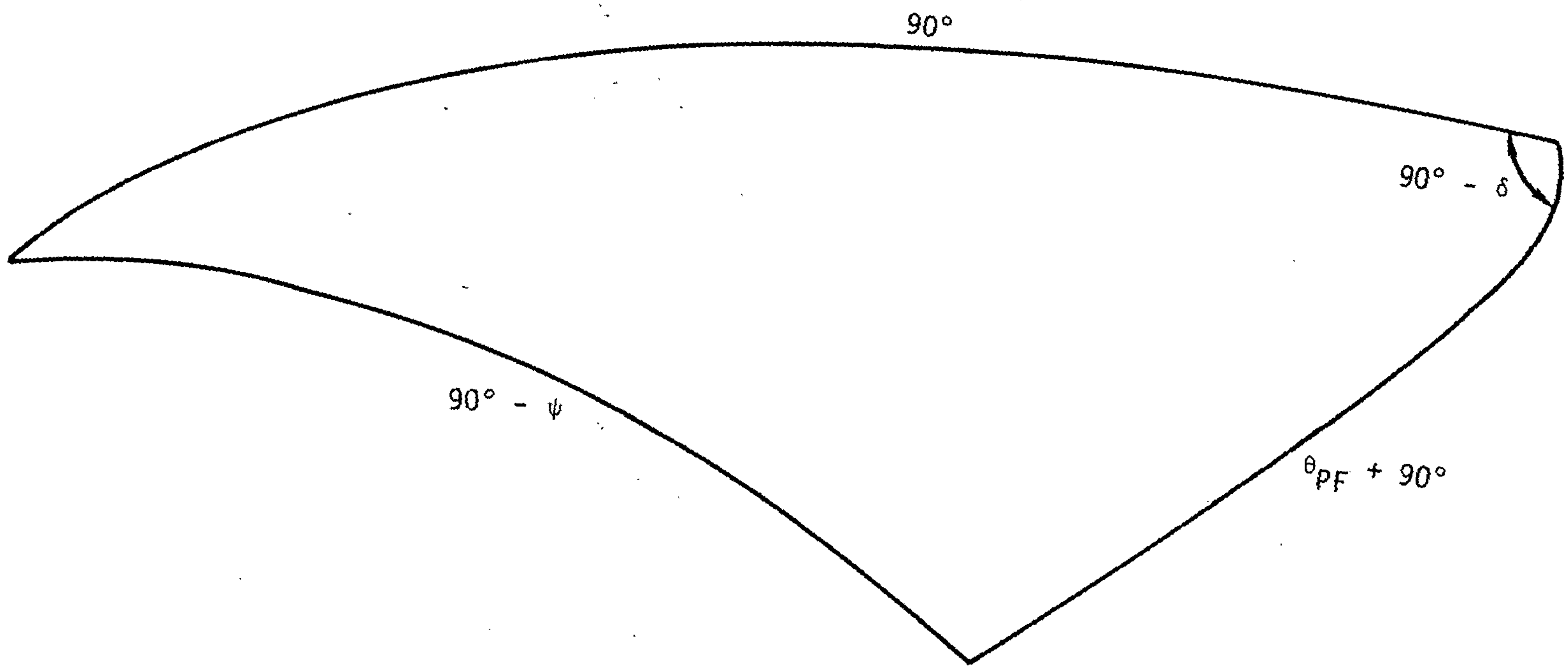


Figure 5.6-2.- Find δ : parallel launch; no yaw steering case.

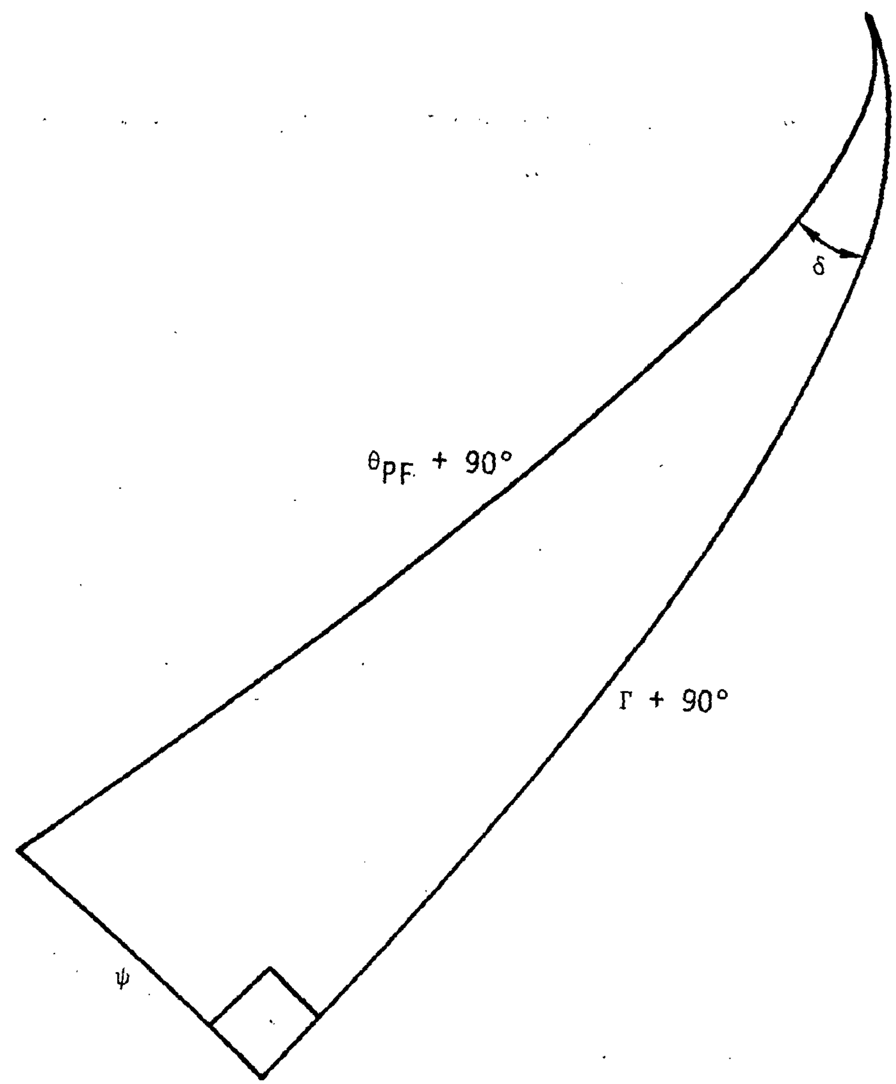


Figure 5.6-3.- Find Γ : parallel launch; no yaw steering case.

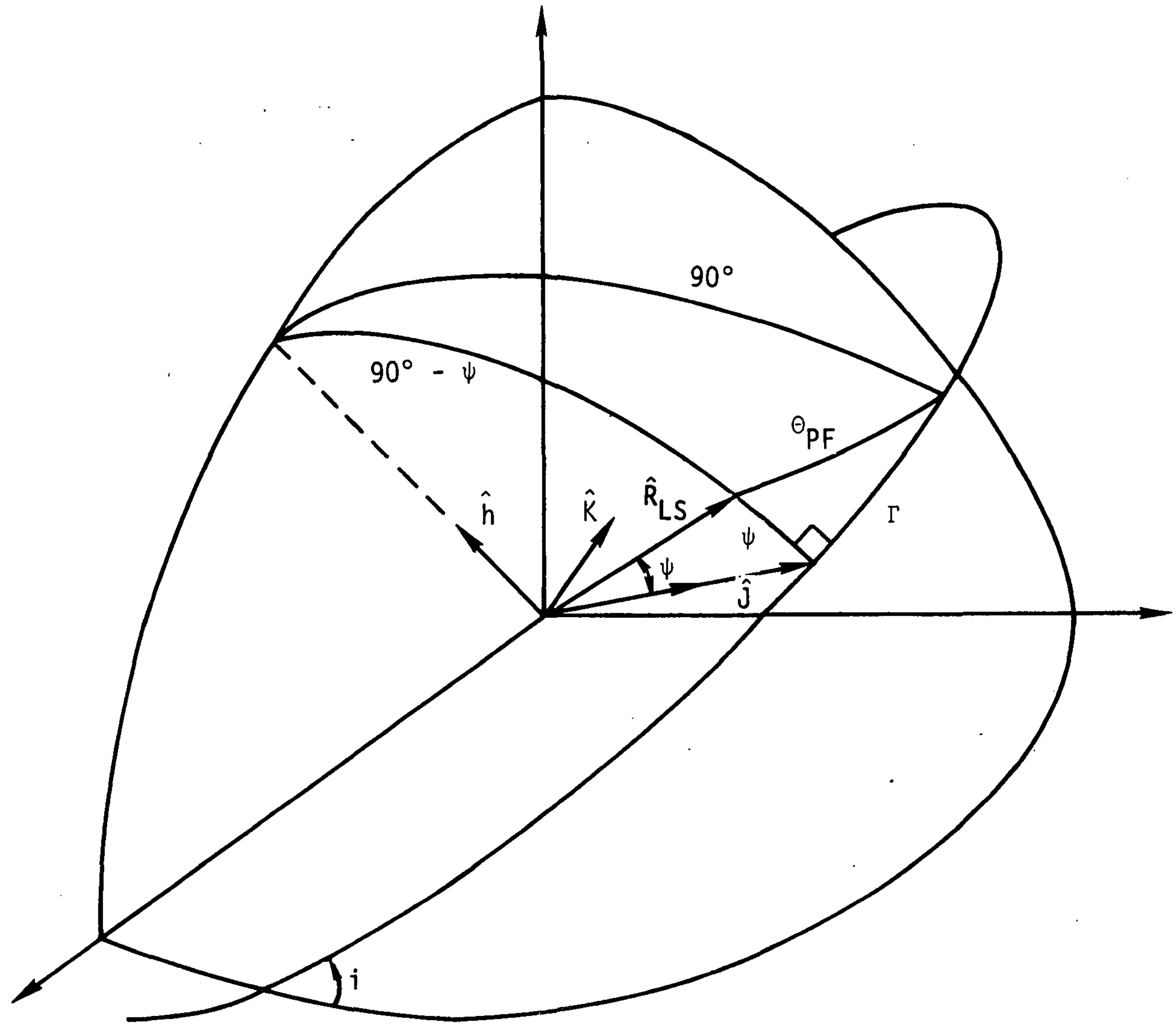


Figure 5.6-4.- Nominal insertion geometry (yaw steering greater than wedge angle).

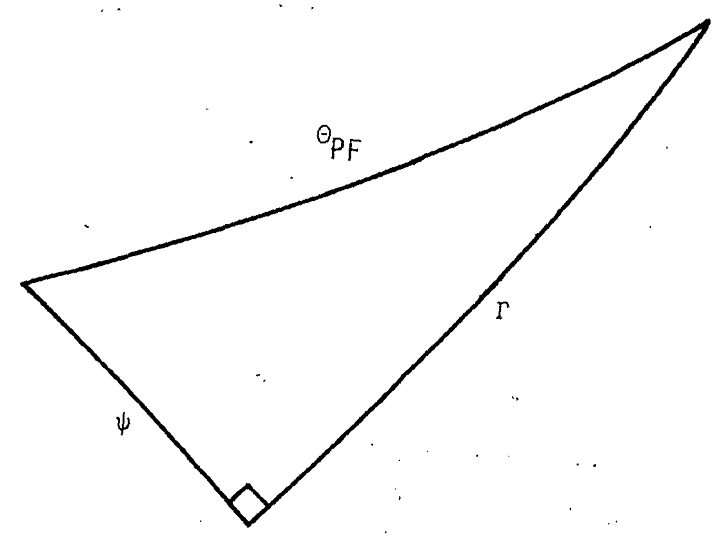


Figure 5.6-5.- Find Γ : nominal launch (yaw steering greater than wedge angle).

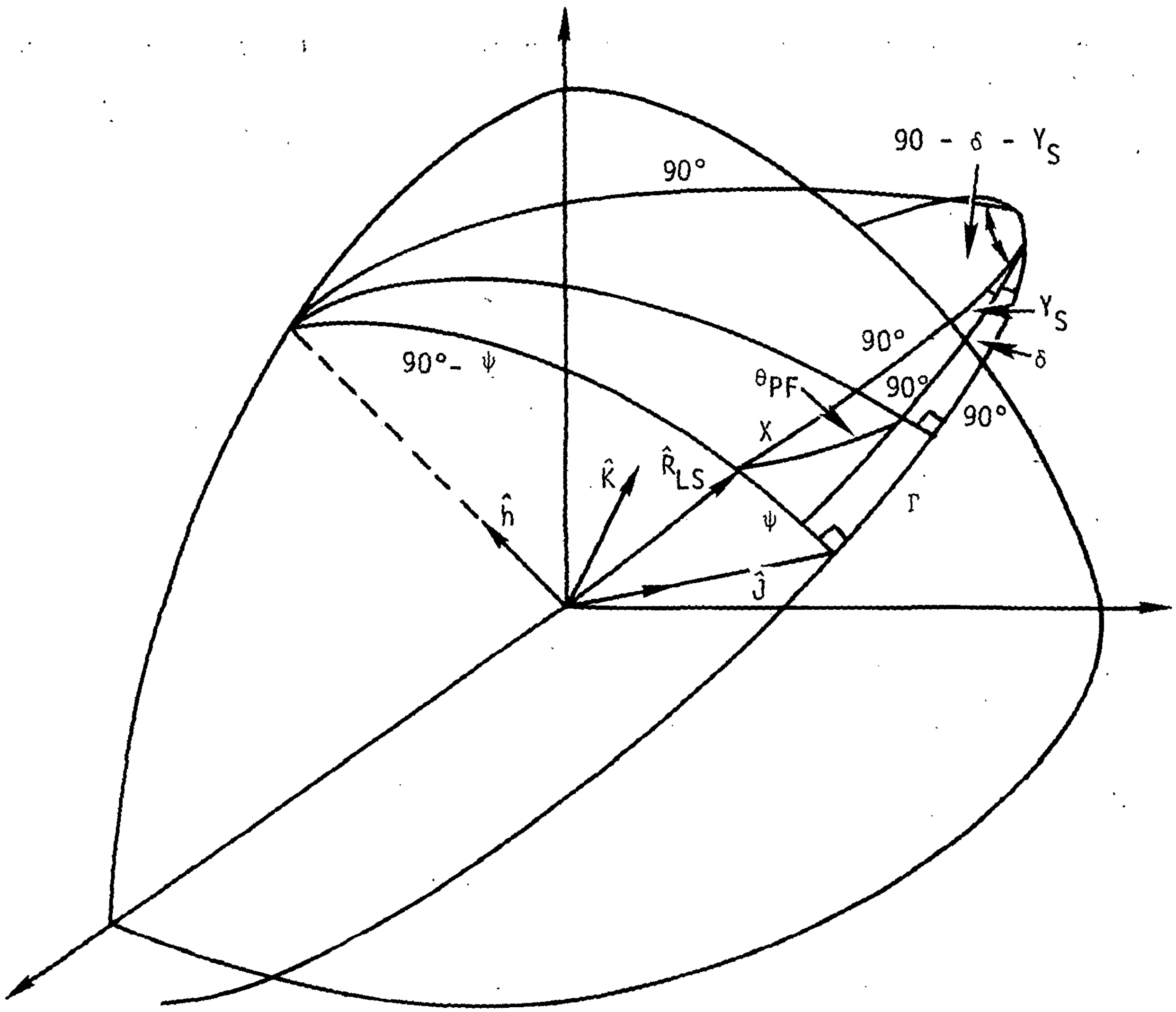


Figure 5.6-6.- Parallel launch geometry (wedge angle greater than yaw steering).

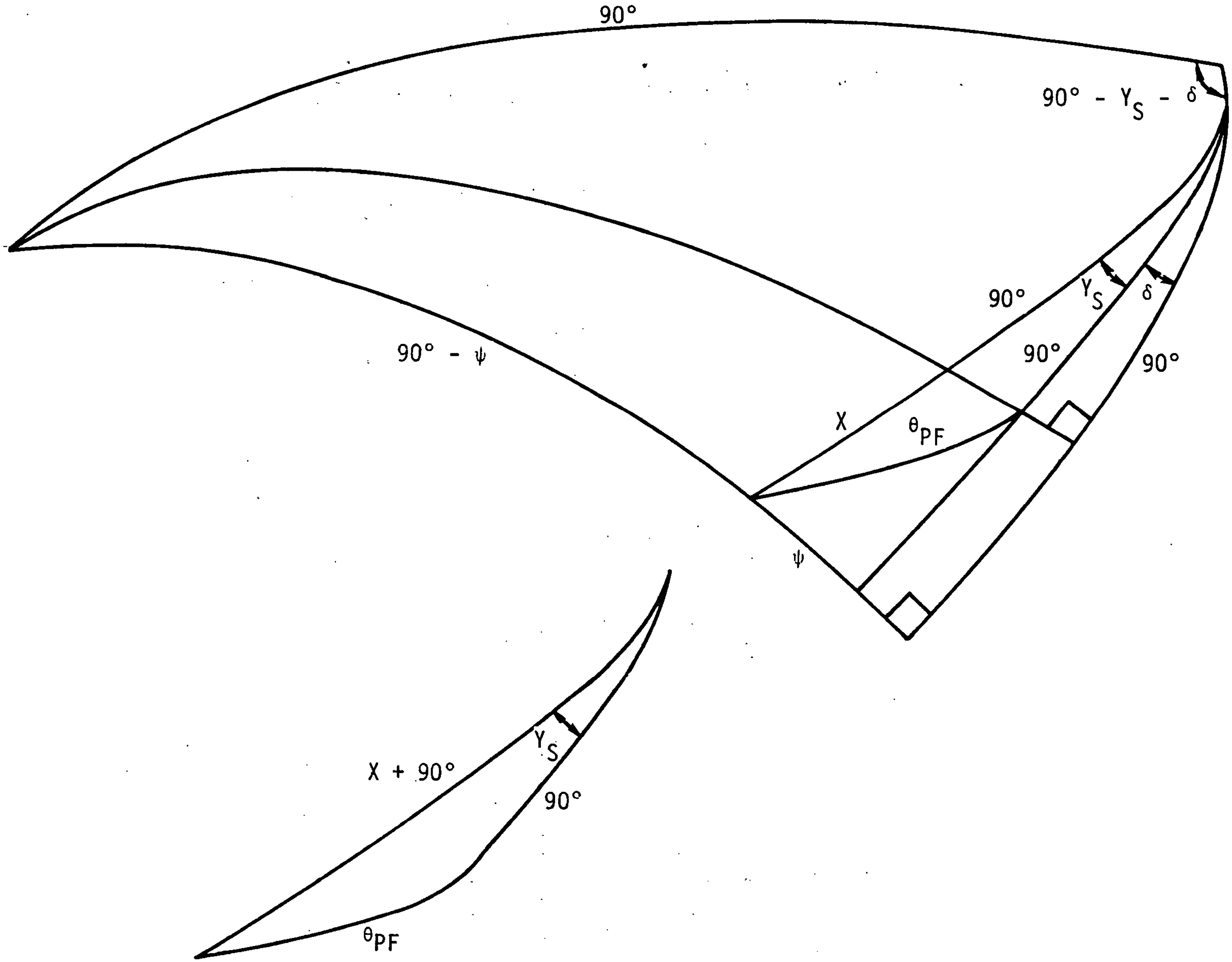


Figure 5.6-7.- Parallel launch geometry (wedge angle greater than yaw steering).

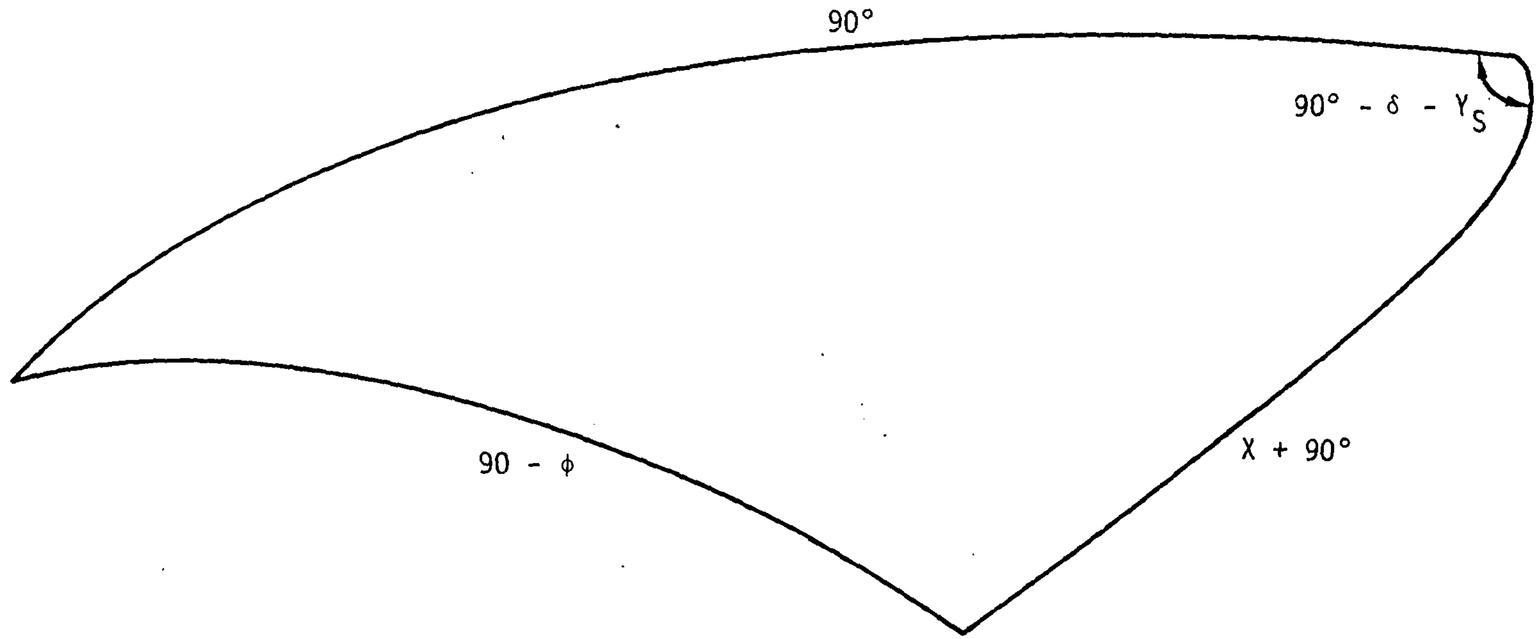


Figure 5.6-8.- Find δ : parallel launch (wedge angle greater than yaw steering).

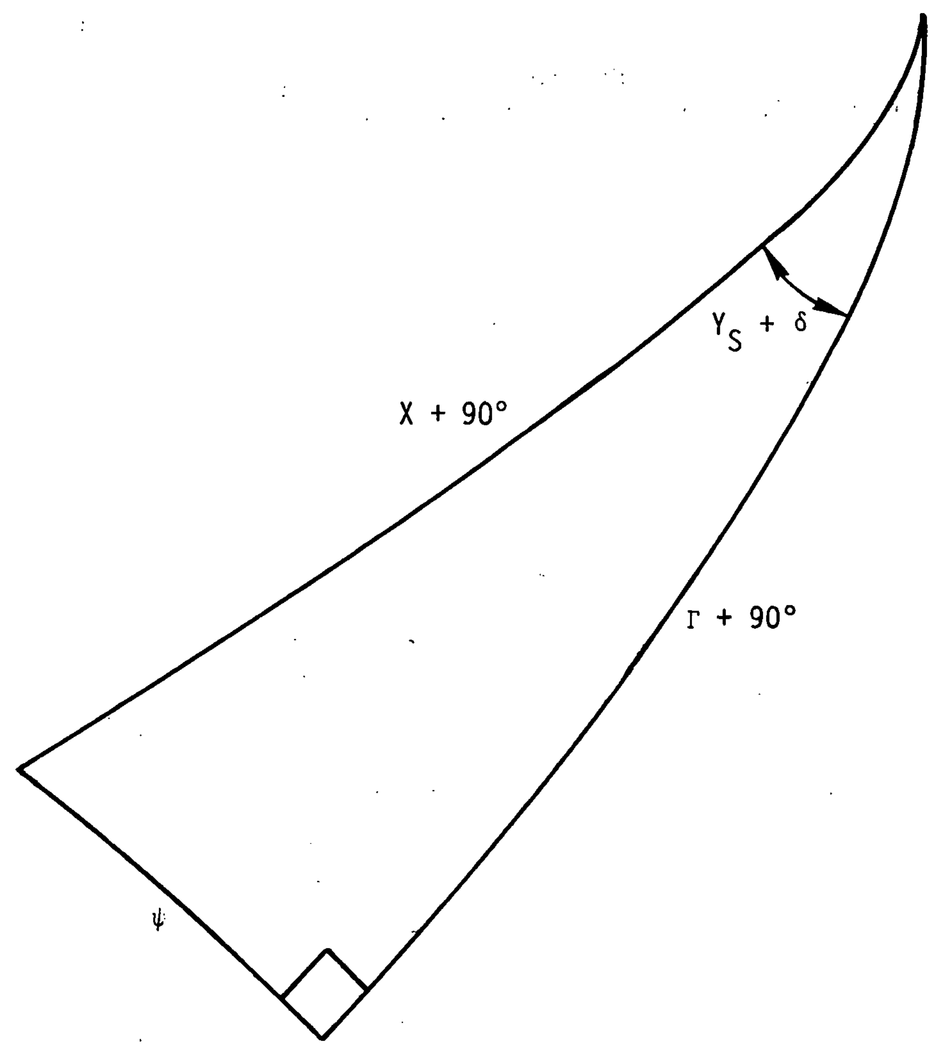
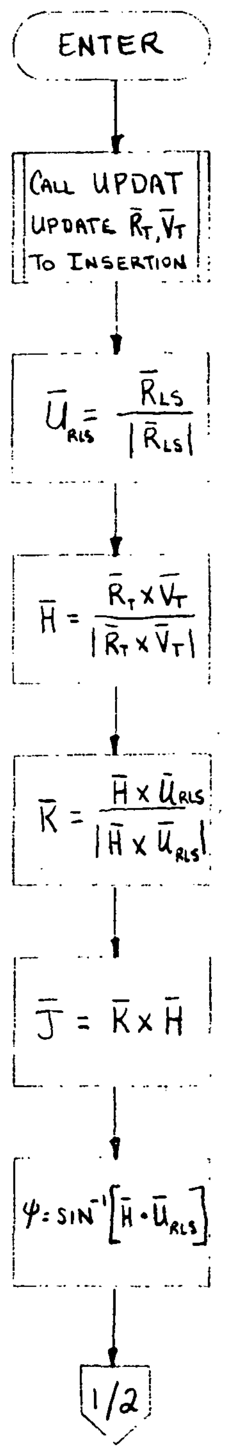
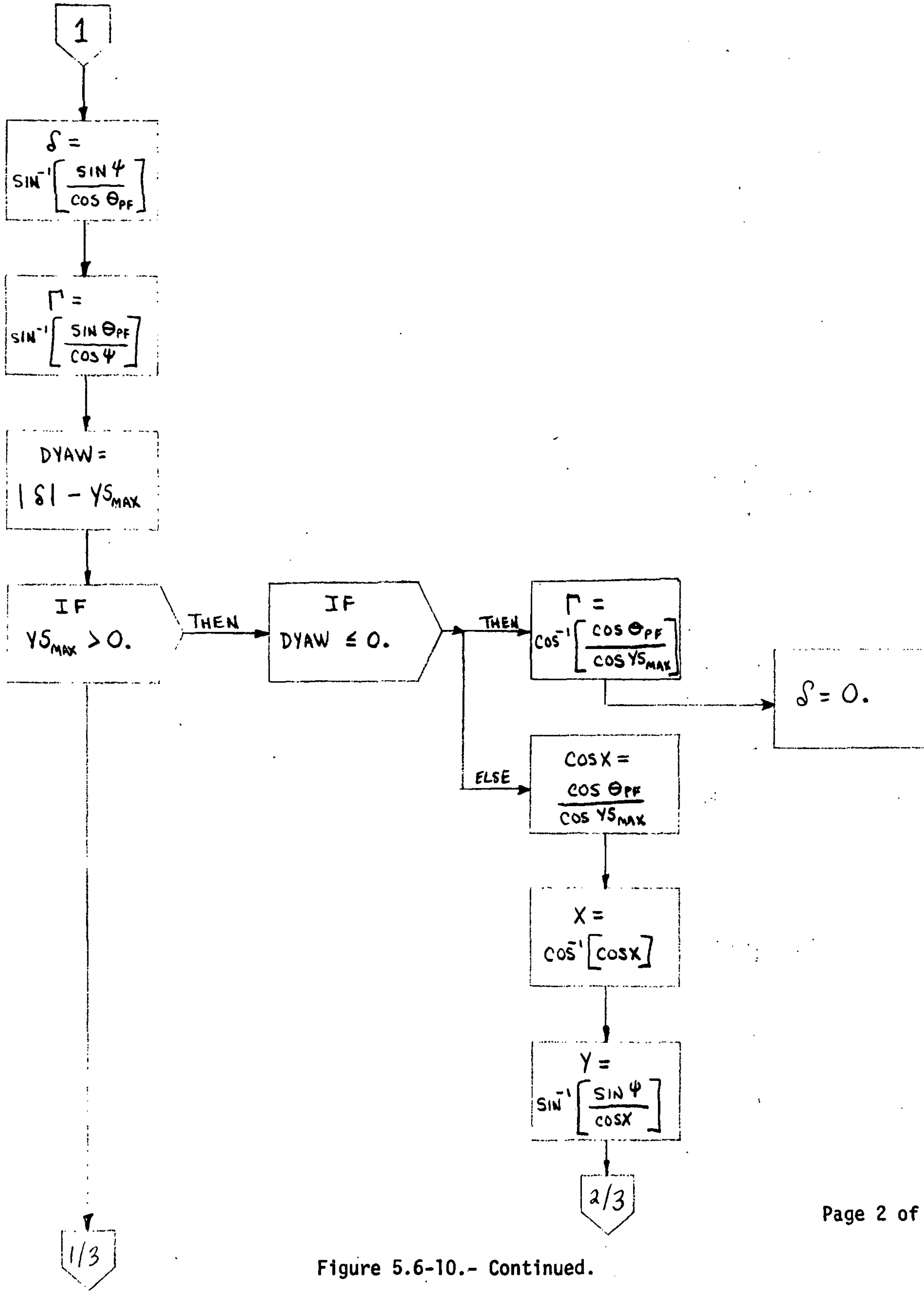


Figure 5.6-9.- Find Γ : parallel launch (wedge angle greater than yaw steering).



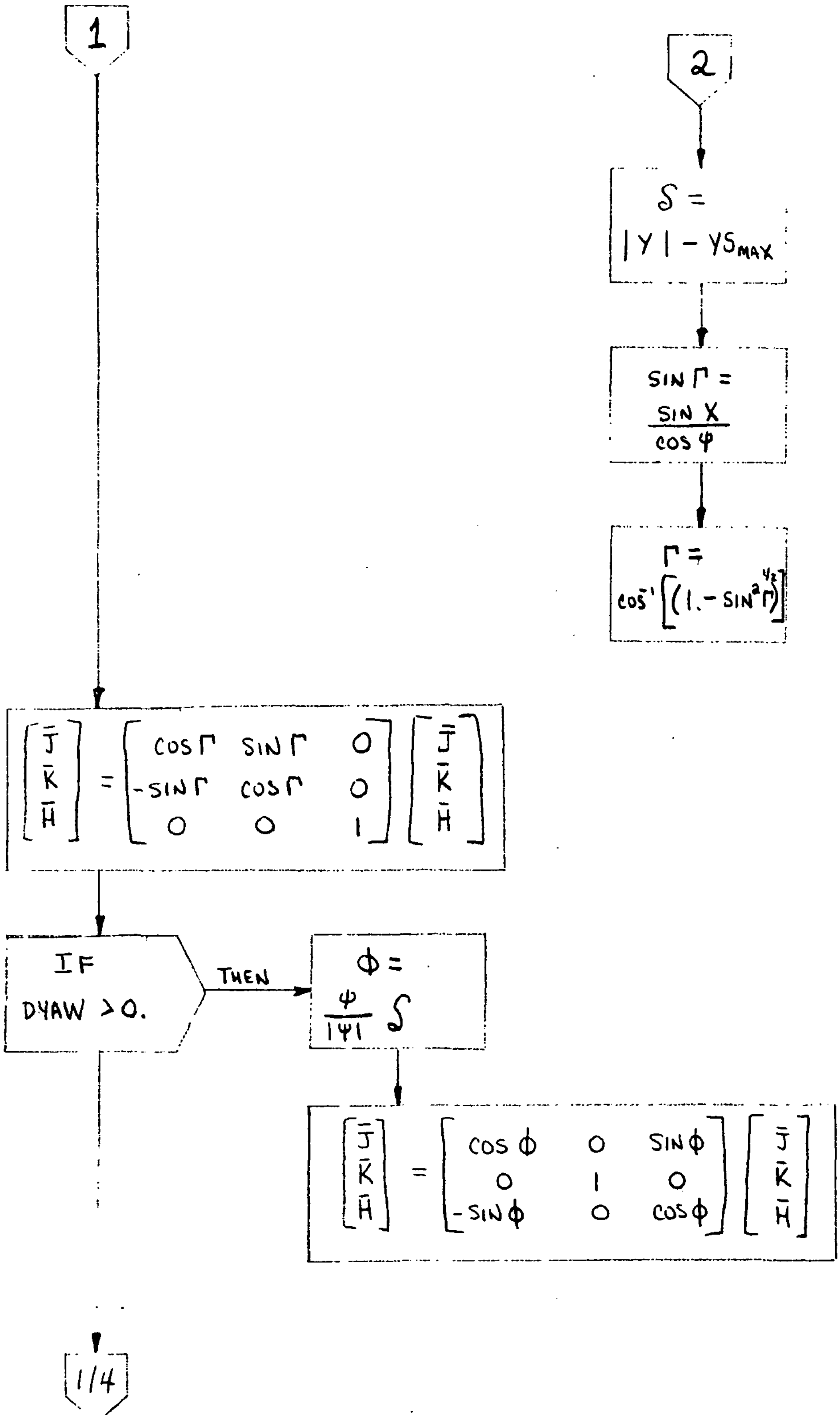
Page 1 of 4

Figure 5.6-10.- LENSR functional logic flow.



Page 2 of 4

Figure 5.6-10.- Continued.



Page 3 of 4

Figure 5.6-10.- Continued.

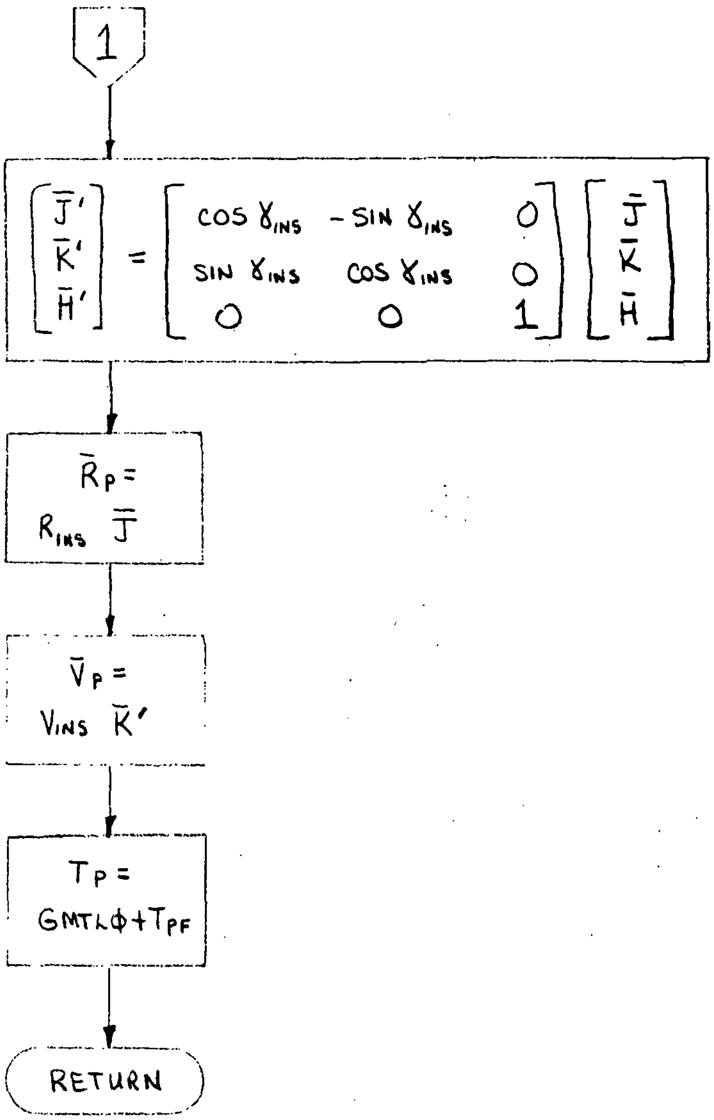


Figure 5.6-10.- Concluded.

5.7 ROUTINE NAME - GMTLS

5.7.1 Purpose

The Greenwich mean time of lift-off for a star time routine (GMTLS) finds the lift-off time required for the chaser vehicle to insert with the phase angle between the chaser and target equal to zero degrees.

5.7.2 Functional Description

Given a target vehicle state vector and a time frame over which to look, GMTLS iterates on the phase angle to find the times at which the chaser vehicle may have a phase match (i.e., zero degree phase angle) at insertion. For each lift-off time computed, GMTLS calls the LENSR routine to compute a chaser insertion vector, computes the phase angle, and checks to see if the phase angle is within a tolerance of equaling zero. If not, the lift-off time is incremented for another iteration. One star time occurs each orbit, and GMTLS finds each one there is within the input bounds.

5.7.3 Assumptions and Limitations

The GMTLO* table can compute a maximum of 10 star times. This number is determined by sizing restrictions and practicality.

5.7.4 Method

The method is presented in the reference.

5.7.5 Routine Input/Output Variables

The GMTLO* input/output variables are presented in table 5.7-I.

5.7.6 Functional Logic Flow

The functional logic flow is presented in figure 5.7-1.

5.7.7 Diagnostics and Debug

When the internal print flag is set, GMTLS prints out various quantities so that the flow of the iteration may be monitored. Those quantities that are output include the lift-off time, the phase angle, and the time to the next star time.

5.7.8 Special Comments

None.

5.7.9 References

Kenyon, E. J.: Project Gemini, Logic for Computation of GMTLO* and GMTLOR. MSC IN 64-FM-70, December 28, 1964.

TABLE 5.7-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine GMTLS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TI	--	Real	I	sec	A	--	Initial search time
TF	--	Real	I	sec	A	--	Final search time
RT	$\bar{R}_T$	Real	I	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I	fps	C	--	Target velocity vector
TT	T_T	Real	I	sec	C	--	Target vector time
BLKC	--	Real	I	--	C	--	Parameter array
BLKB	--	Real	I	--	C	--	Parameter array
DET	--	Real	I	sec	C	--	Time iteration tolerance
STABLE	--	Intg	I	--	C	--	GMTLO* table flag
ERROR	--	Intg	O	--	C	--	Error return code
GSTAR	--	Real	O	sec	C	--	Last GMTLO* time computed
K25	--	Intg	O	--	C	--	Number of GMTLO* times computed

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

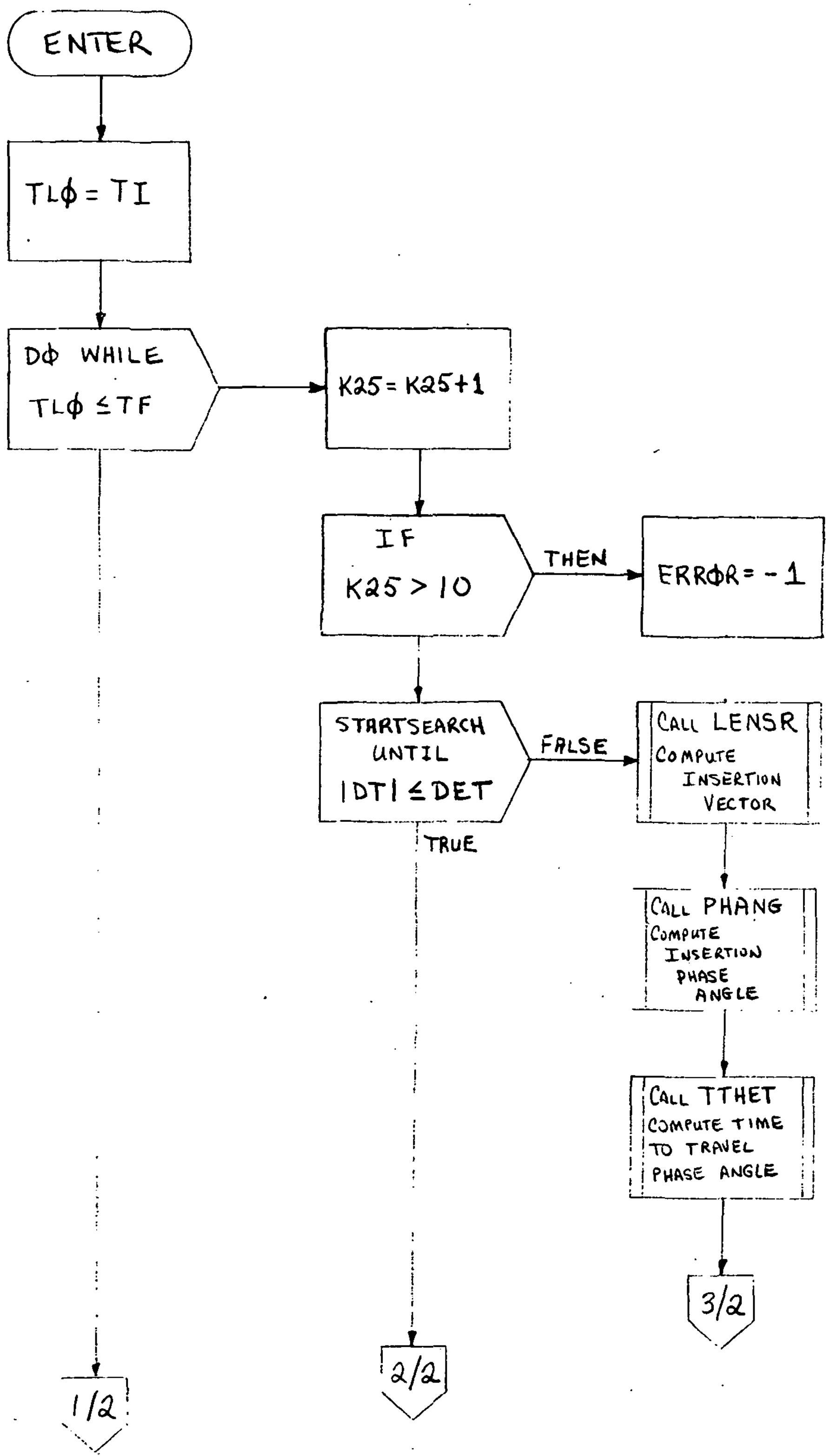
TABLE 5.7-I.- Concluded

Routine GMTLS

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
STAR	—	Real	O	sec	C	—	GMTLO* time array

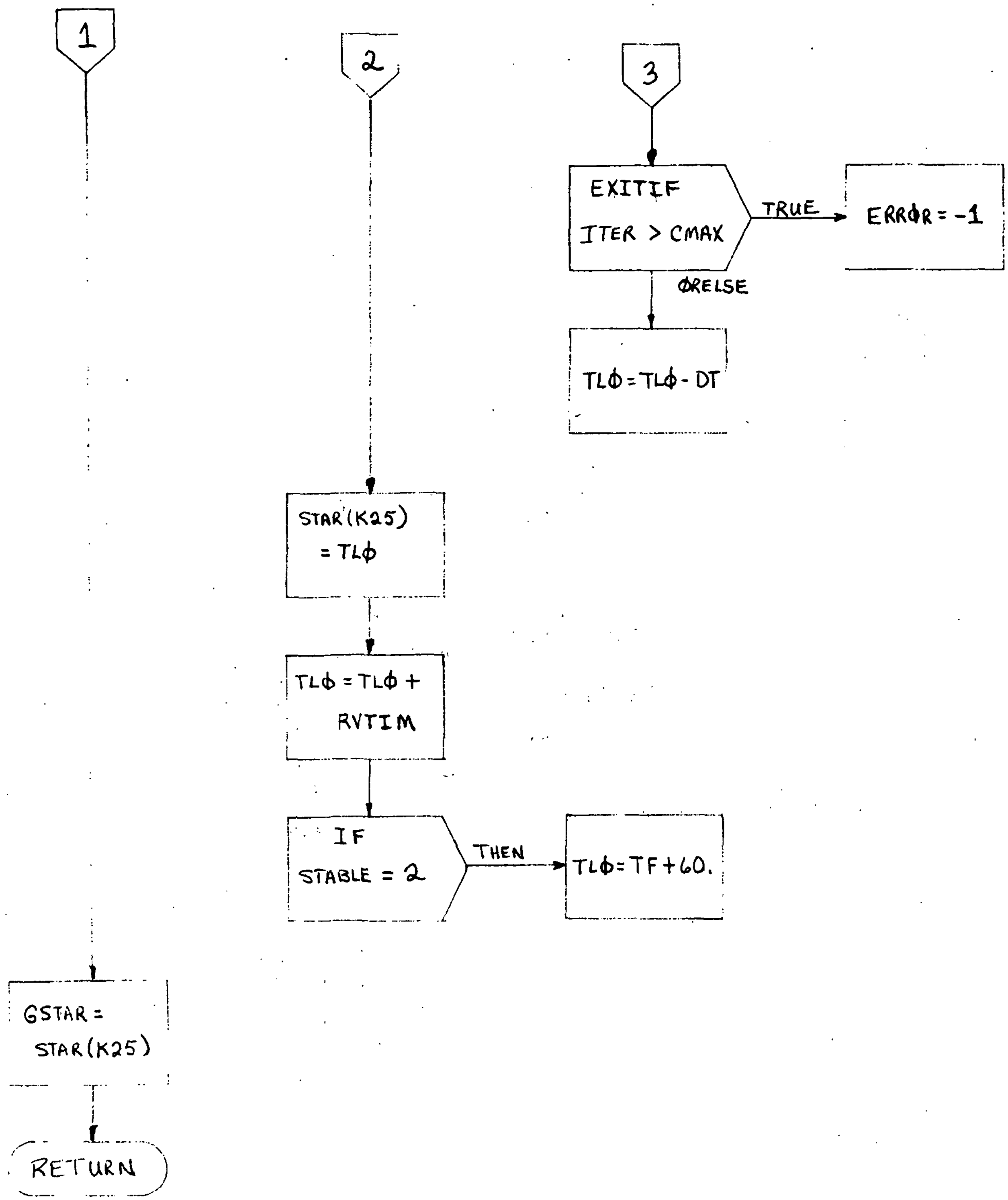
NOTES:

TYPE				USE	SOURCE
Free	Dubl	18CH	Mix	I = Input	IT = Interface Table
Intg	2CH	36CH	Char	O = Output	T = Terminal
Real	6CH	72CH	Bin	I/O = Input/Output	A = Calling Argument
					C = Common
					F = Disk File
					SAM = System Available Memory



Page 1 of 2

Figure 5.7-1.- GMTLS functional logic flow.



Page 2 of 2

Figure 5.7-1.- Concluded.

5.8 ROUTINE NAME - Subroutine LWDSP

5.8.1 Purpose

LWDSP is the Launch Window Display routine.

5.8.2 Functional Description

Given the launch window data computed in LWT and the GMTLO* times computed in GMTLS, LWDSP arranges and writes the launch window display and the GMTLO* table.

5.8.3 Assumptions and Limitations

None.

5.8.4 Method

None.

5.8.5 Routine Input/Output Variables

The LWDSP input/output variables are presented in Table 5.8-I.

5.8.6 Functional Logic Flow

None.

5.8.7 Diagnostics and Debug

None.

5.8.8 Special Comments

None.

5.8.9 References

None.

TABLE 5.8-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWDSP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RT	$\bar{R}_T$	Real	I/O	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I/O	fps	C	--	Target velocity vector
TT	T_T	Real	I/O	sec	C	--	Target vector time
SESCON	--	Free	I	--	C	--	Session constants
BLKC	--	Real	I	--	C	--	Parameter array
BLKB	--	Real	I	--	C	--	Parameter array
LATLS	ϕ_{LS}	Real	I	rad	C	--	Geocentric latitude of launch site
LONGLS	λ_{LS}	Real	I	rad	C	--	Geographic longitude of launch site
FTNM	--	Real	I	ft/n.mi.	C	--	Conversion factor (feet per n. mi.)
OPEN	--	Real	I	sec	C	--	In-plane opening time
CLOSE	--	Real	I	sec	C	--	In-plane closing time
PAC	--	Real	I	rad	C	--	Phase angle at closing
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.8-I.- Concluded

Routine LWDSP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
PAO	--	Real	I	rad	C	--	Phase angle at opening
NS	--	Intg	I	--	C	--	In-plane launch window opening and closing times
RAD	--	Real	I	deg/rad	C	--	Conversion factor (deg/rad)
RREF	--	Real	I	ft	C	--	Reference radius
ERROR	--	Intg	O	--	C	--	Error return code
LWT DISPLAY	--	--	O	--	--	--	See table 4-IV(b) of this processor for definition of contents.
GMTLO* TABLE	--	--	O	--	--	--	See table 4-IV(d) of this processor for definition of contents.
STATE VECTOR DISPLAY	--	--	O	--	--	--	See table 4-IV(j) of this processor for definition of contents.

NOTES:

TYPE
Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.9 ROUTINE NAME - Subroutine LWPT

5.9.1 Purpose

The purpose of the Launch Window Parameter Table (LWPT) routine is to compute and display selected launch window parameters over a given time frame around the inplane time.

5.9.2 Functional Description

Given a start time, the LWPT computes selected launch parameters associated with that lift-off time and displays them in a table. The lift-off time is incremented and new launch parameters are calculated and displayed. The calculations continue until the stop time is exceeded.

5.9.3 Assumptions and Limitations

The launch window parameter table is an auxiliary table to the launch window time table. As such, the user can only call it when the launch window and launch targeting option flag calls LWT (LW=0 or LW=2).

The inplane launch window opening and closing flag (NS) determines which inplane nodes will be computed and is used concurrently with the launch parameter table flag (LPT). Thus, NS should be set as follows,

if LPT = 1 , NS = 0 or 2

LPT = 2 , NS = 1 or 2

LPT = 3 , NS = 2

LPT = 4 , NS = 2

5.9.4 Method

None.

5.9.5 Routine Input/Output Variables

The LWPT input/output variables are presented in Table 5.9-I.

5.9.6 Functional Logic Flow

None.

5.9.7 Diagnostics and Debug

None.

5.9.8 Special Comments

Note that the wedge angle (YP) is zero at the inplane time (i.e., the chaser is inplane with the target orbit plane at insertion). YP is computed analytically in LENSR. However, empirically, the inplane time is approximately 300 seconds earlier than the analytical inplane time. See the LWT parameters DTOPT, TIP, and TPLANE.

5.9.9 References

None.

TABLE 5.9-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
RT	$\bar{R}_T$	Real	I	ft	C	—	Target position vector
VT	$\bar{V}_T$	Real	I	fps	C	--	Target velocity vector
TT	T_T	Real	I	sec	C	—	Target vector time
SESCON	--	Free	I	—	C	--	Session constants
LATLS	ϕ_{LS}	Real	I	rad	C	—	Geocentric latitude of launch site
LONGLS	λ_{LS}	Real	I	rad	C	--	Geographic longitude of launch site
PFT	T_{PF}	Real	I	sec	C	—	Powered flight time
RAD	--	Real	I	deg/rad	C	--	Conversion factor (deg/rad)
CLOSE	--	Real	I	sec	C	—	In-plane closing time
DELNO	--	Real	I	rad	C	--	Angle between descending nodes
LPT	—	Intg	I	--	C	--	LWPT options flag
NS	--	Intg	I	—	C	--	LW in-plane opening or closing flag
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.9-I.- Concluded

Routine LWPT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
OPEN	--	Real	I	sec	C	--	In-plane opening time
TEND	--	Real	I	sec	C	--	Delta time after inplane to stop LWPT
TSTART	--	Real	I	sec	C	--	Delta time prior to inplane to stop LWPT
TSTEP	--	Real	I	sec	C	--	Delta time step in LWPT
DISPLAY PARAMETERS	--	--	O	--	--	--	See table 4-IV(f) of this processor for definition of contents
ERROR	--	Intg	O	--	C	--	Error return code

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE: I = Input; O = Output; I/O = Input/Output

SOURCE: IT = Interface Table; T = Terminal; A = Calling Argument; C = Common; F = Disk File; SAM = System Available Memory

5.10 ROUTINE NAME - Subroutine RLOT

5.10.1 Purpose

The purpose of the Recommended Lift-off Time (RLOT) routine is to compute the user specified lift-off time and insertion orbit conditions so that the routine TARGT may generate an appropriate insertion vector.

5.10.2 Functional Description

RLOT, in conjunction with the targeting routine TARGT, uses analytical equations to determine the insertion vector required for a chaser vehicle to rendezvous at some time later with a target vehicle already in orbit. RLOT is designed to calculate and to present as output six targeting quantities, the time of guidance reference release, and a corresponding chaser insertion orbit state vector.

The user has a choice of six lift-off time options and three insertion conditions options. The six lift-off time options are, (1) input time; (2) lift-off to achieve a desired phase angle at insertion; (3) lift-off on GMTLO* plus a bias time, using GMTLOR as a threshold time; (4) lift-off on GMTLO* plus a bias, using TPLANE as a threshold time; (5) lift-off based on inplane time (TPLANE); and (6) iterate to lift-off on inplane time (TYAW) based on target orbit phase angle. The three insertion cutoff conditions are, (1) input the velocity magnitude, flight path angle and radius magnitude at insertion; (2) input the flight path angle and radius magnitude at insertion, and the height difference between the target and chaser desired at an input angle from insertion; and (3) input the flight path angle and radius magnitude at insertion, and the altitude desired at an input angle from insertion.

5.10.3 Assumptions and Limitations

None.

5.10.4 Method

Several of the options require mathematical formulation. The second lift-off time option uses an input threshold time for the first guess at the proper time to lift-off to achieve the desired insertion phase angle (Θ_D). The phase angle (Θ_A) is computed. Then, $\phi = \Theta_A - \Theta_D$. If $\phi \leq$ TOLERANCE, then the lift-off time (GMTLO) is found. If $\phi >$ TOLERANCE, then GMTLO = $t_0 - (\phi/N)$, where t_0 is the previous value of GMTLO and N is the target vehicle mean motion.

The time of inplane lift-off (T_{IN}) is either input or calculated in the routine LWT. Analytically, the equation for the inplane at insertion liftoff time is

$$T_{IN} = T_{IP} + (\dot{\Omega} \cdot t_{PF})/\omega_e + \Delta t_{OPT} ,$$

where T_{IP} = analytical inplane at launch time (computed by NPLAN)

$\dot{\Omega}$ = nodal precession rate of target

t_{PF} = powered flight time

ω_e = Earth's rotation rate

Δt_{OPT} = time between theoretical inplane at insertion point and the empirical inplane point (accounting for vehicle drift into target plane following lift-off)

In LWT, T_{IN} is found by iterating on the wedge angle at insertion and terminating when the wedge angle is zero.

However, in a rendezvous situation when rendezvous does not occur at insertion (i.e., the phase angle is not equal to zero and the chaser insertion orbit is not equal to the target orbit), the chaser vehicle must be placed out-of-plane by delta node angle (DELNO) in order to be inplane at the rendezvous point. If lift-off occurs at T_{IN} and the chaser must be out-of-plane at insertion, yaw steering by the launch vehicle must occur. Thus, if the launch vehicle yaw steering is to be a minimum, lift-off must occur some delta time (Δt) from T_{IN}, leaving the chaser out-of-plane at insertion. The equation for this is

$$T_{YAW} = T_{IN} + \Delta t ,$$

or

$$T_{YAW} = T_{IN} + DELNO/\omega_e .$$

Since the value of DELNO is dependent upon the lift-off time, T_{YAW} must be found by iteration. Thus, at each iteration step, the previous value of T_{YAW} is compared with the present value. When the difference is minimized, T_{YAW} is found.

The second and third insertion conditions options also require iteration. An initial velocity magnitude is used to calculate the altitude desired at the input argument of latitude. A delta altitude (Δh) between the actual and the desired altitudes is calculated, and a new velocity magnitude is calculated by the equation

$$V_{INS} = V_O + \Delta V$$

where

$$\Delta V = \Delta h/3420.$$

This equation for ΔV is obtained from the reference, using

$$\Delta h = C\ \Delta V$$

where

$$C = 0.56 \text{ n. mi./fps}$$

Math symbol	Internal code symbol	Math symbol	Internal code symbol
Θ_A	PA	Θ_D	OFFSET
N	N	T_{IN}	TPLANE
T_{IP}	TIP	t_{PF}	PFT
ω_e	WE	Δt_{OPT}	DTOPT
T_{YAW}	TYAW	V_{INS}	VINS
Δh	DALT		

5.10.5 Routine Input/Output Variables

The RLOT input/output variables are presented in Table 5.10-I.

5.10.6 Functional Logic Flow

The functional logic flow for RLOT is presented in Figure 5.10-1.

5.10.7 Diagnostics and Debug

If the internal print flag is set, RLOT prints out for reference the input target state vector.

5.10.8 Special Comments

None.

5.10.9 References

McDonough, R. K.: Altitude Change Resulting from Impulsive, Horizontal Maneuvers in the Nominal GTA-6 Agena Orbit. MSC Memorandum 65-FM-5, May 5, 1965.

TABLE 5.10-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine RLOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKC	—	Real	I	--	C	—	Parameters array
BLKB	--	Real	I	—	C	--	Parameters array
RT	$\bar{R}_T$	Real	I/O	ft	C	—	Target position vector
VT	$\bar{V}_T$	Real	I/O	fps	C	--	Target velocity vector
TT	T_T	Real	I/O	sec	C	—	Target vector time
ANOM	--	Real	I	ft	C	ANOM	Nominal insertion semimajor axis
A5	—	Real	I	rad	C	DU	Angle from insertion
BIAS	--	Real	I/O	sec	C	BIAS	Bias that is added to GMTLO*
CMAX	—	Intg	I	--	C	—	Maximum iterations
DHW	--	Real	I	ft	C	DHW	Delta height or altitude desired
GMTLOR	—	Real	I	sec	C	GMTLOR	Recommended lift-off time
INSCO	--	Intg	I	--	C	INSCO	Insertion cutoff conditions flag
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.10-I.- Continued

Routine RLOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
K70	--	Intg	I	--	C	WRAP	Flag to wrap initial phase angle
LOT	--	Intg	I	--	C	LOT	Lift-off times option routing flag
OFFSET	--	Real	I	rad	C	OFFSET	Phase angle desired at insertion
TPLANE	T_{IN}	Real	I	sec	C	TPLANE	Time of in-plane lift-off
TRANS	--	Real	I	sec	C	TRANS	Delta-T added to TPLANE
MU	μ	Real	I	ft^3/sec^2	C	--	Earth gravitational constant
RINS	RINS	Real	I	ft	C	--	Radius of insertion
VINS	VINS	Real	I/O	fps	C	--	Velocity magnitude at insertion
RP	$\bar{R}_P$	Real	O	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	O	fps	C	--	Chaser velocity vector
TP	T_P	Real	O	sec	C	--	Chaser vector time
ERROR	--	Intg	O	--	C	--	Error return code
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.10-I.- Concluded

Routine RLOT

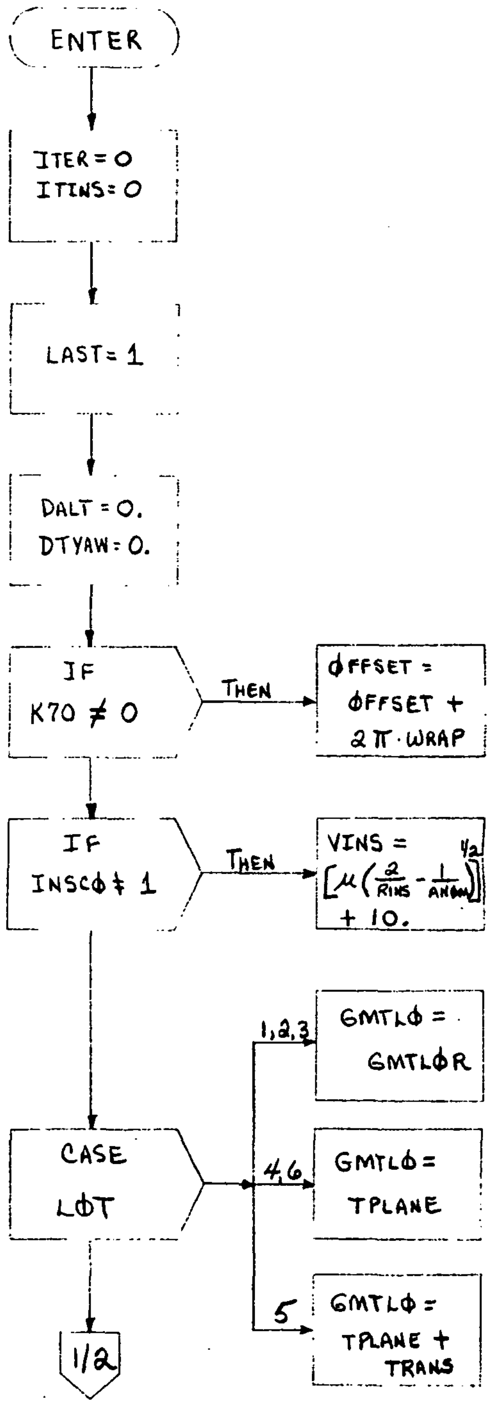
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GMTLO	--	Real	O	sec	C	--	GMT of lift-off
PA	θ_A	Real	O	rad	C	--	Phase angle at insertion
STABLE	--	Intg	O	--	C	STABLE	GMTLO* table flag
TYAW	T_{YAW}	Real	O	sec	C	--	Time of minimum yaw steering lift-off

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

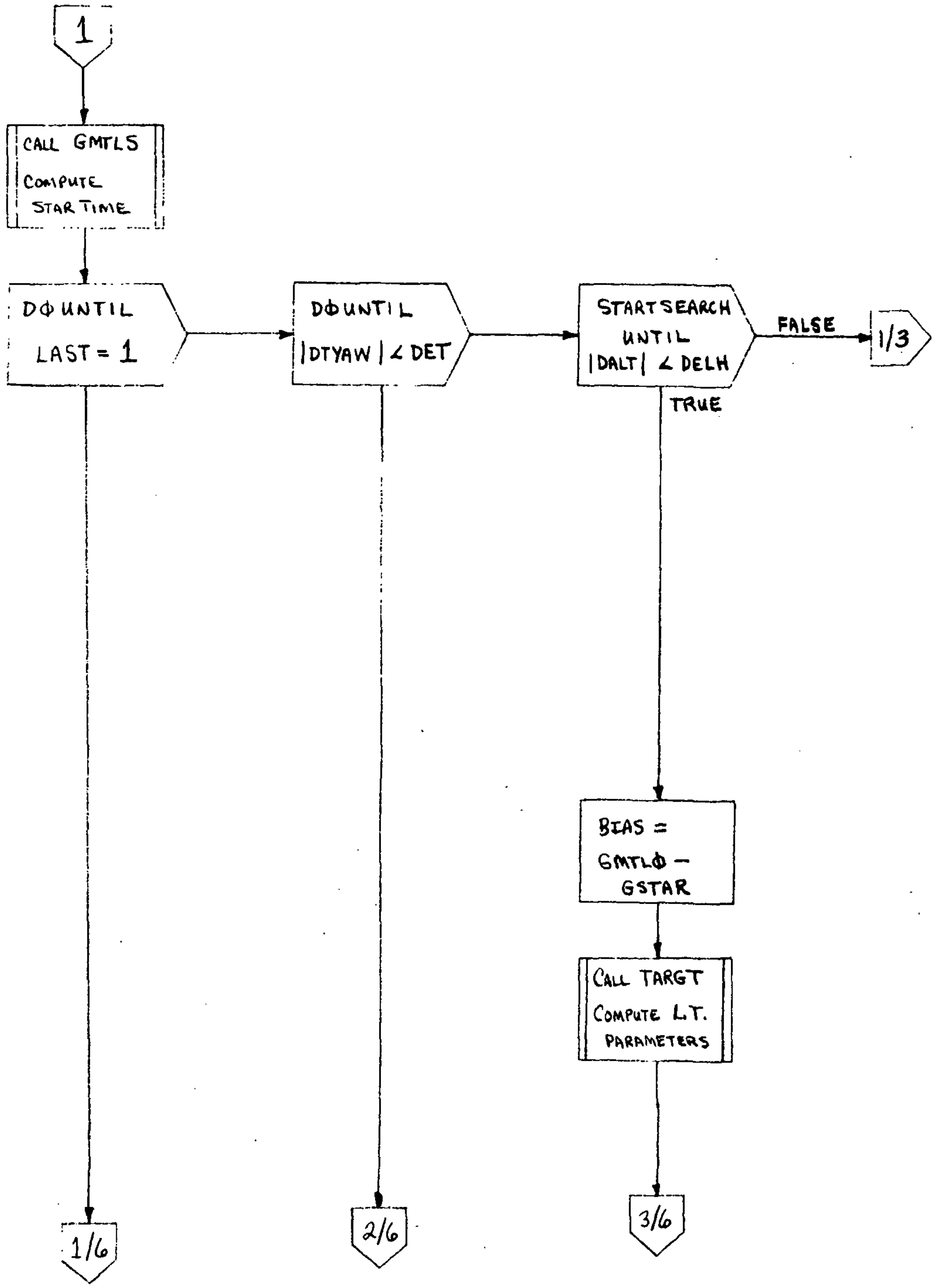
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



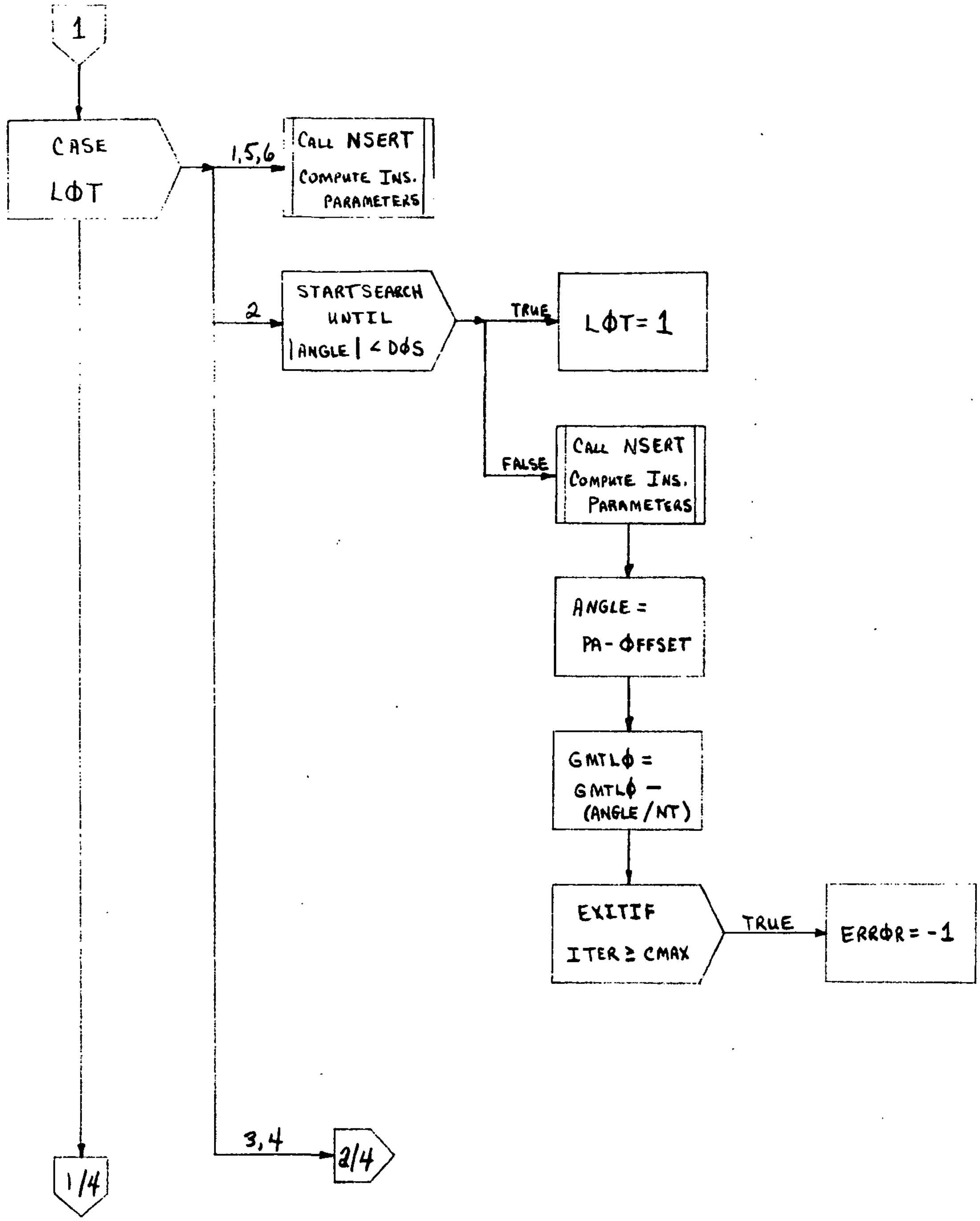
Page 1 of 6

Figure 5.10-1.- RLOT functional logic flow.



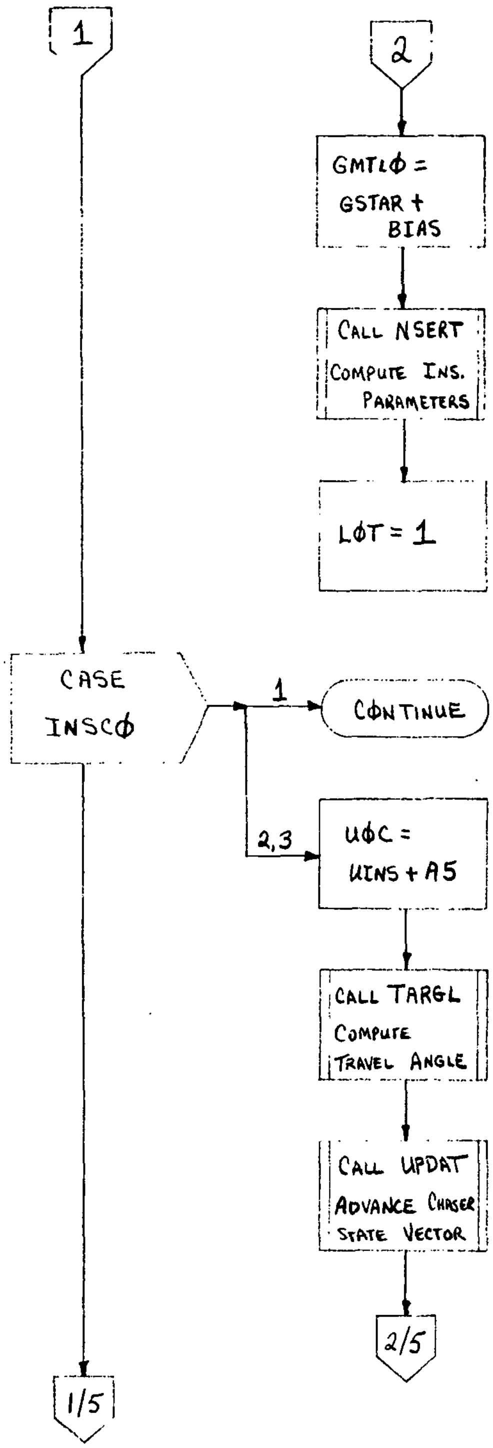
Page 2 of 6

Figure 5.10-1.- Continued.



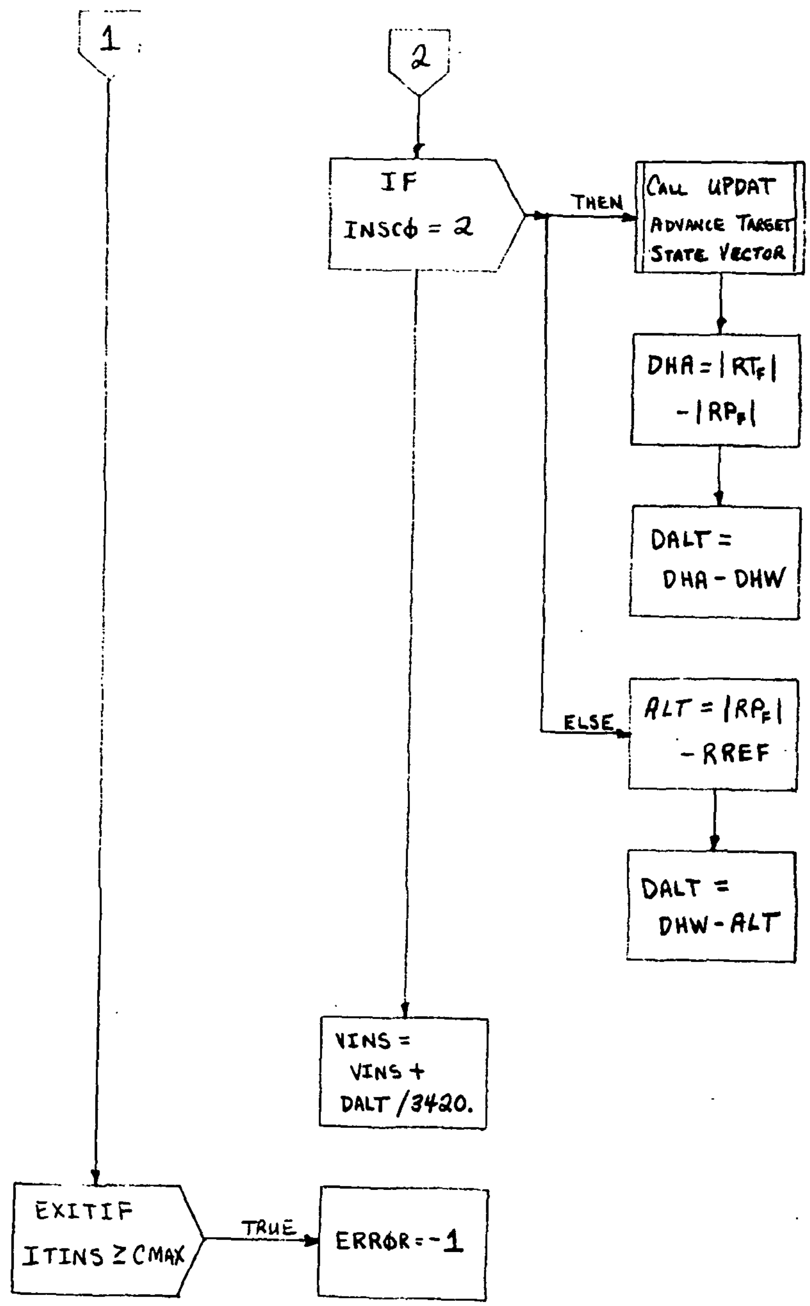
Page 3 of 6

Figure 5.10-1.- Continued.



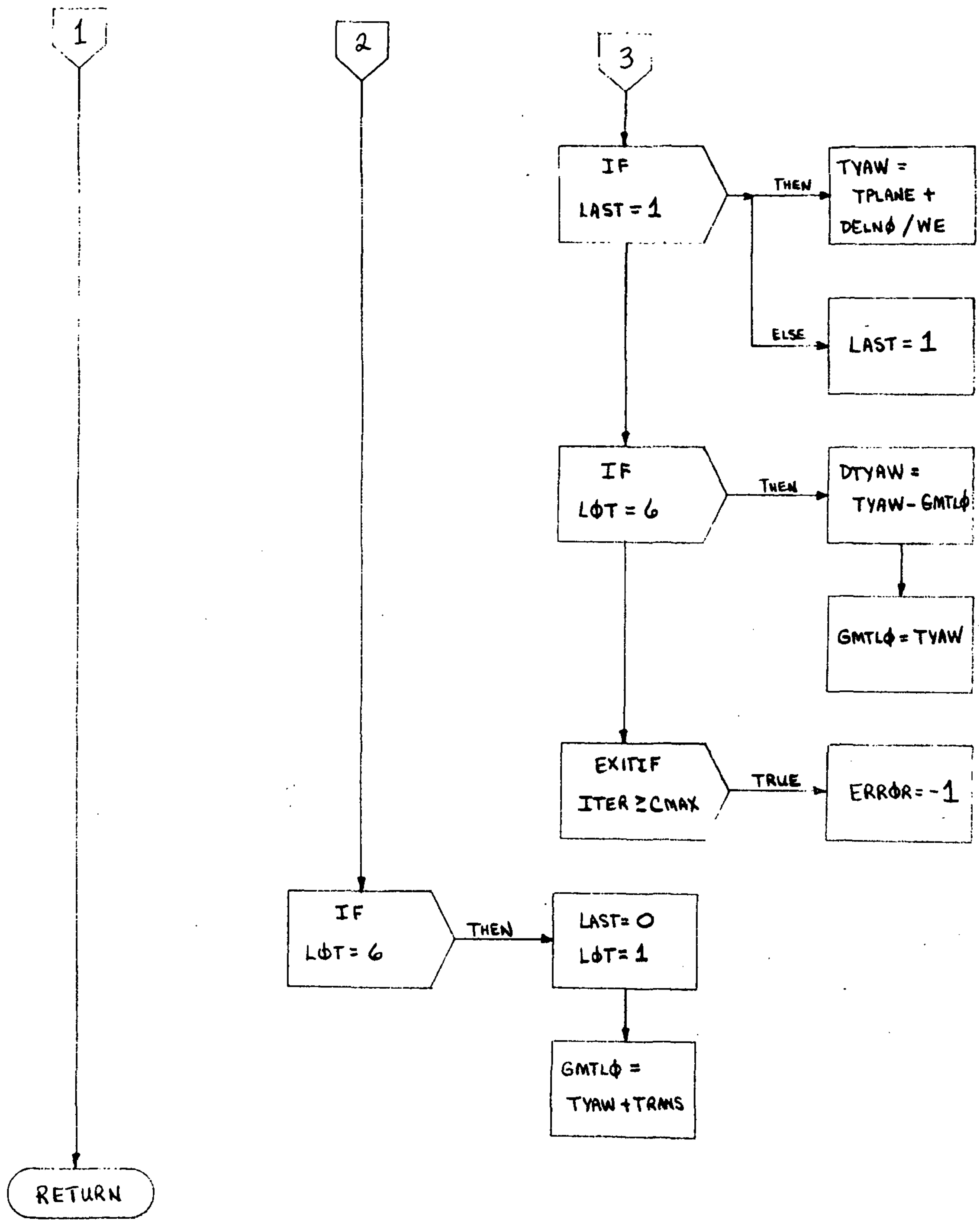
Page 4 of 6

Figure 5.10-1.- Continued.



Page 5 of 6

Figure 5.10-1.- Continued.



Page 6 of 6

Figure 5.10-1.- Concluded.

5.11 ROUTINE NAME - Subroutine NSERT

5.11.1 Purpose

The purpose of NSERT is to compute particular insertion parameters that are needed by the RLOT routine.

5.11.2 Functional Description

NSERT updates and saves the target vehicle state vector at the time of insertion. NSERT also generates, through LENSR, a chaser vehicle insertion vector and saves it. Then, the phase angle at insertion, and the phase angle, time lag, delta altitude, and argument of latitude at the counter line are all computed.

5.11.3 Assumptions and Limitations

None.

5.11.4 Method

None.

5.11.5 Routine Input/Output Variables

The NSERT input/output variables are presented in Table 5.11-I.

5.11.6 Functional Logic Flow

None.

5.11.7 Diagnostics and Debug

If the internal print flag is turned on, NSERT prints the radius magnitude, flight path angle, and velocity magnitude at insertion, the G.M.T. of lift-off, and the phase angle, time lag, delta altitude, and argument of latitude at the counter line.

5.11.8 Special Comments

None.

5.11.9 References

None.

TABLE 5.11-1.- ROUTINE INPUT/OUTPUT VARIABLES

Routine NSERT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BVEC	--	Real	I	--	A	--	Base vector for target
BLKC	--	Real	I	--	C	--	Parameter array
BLKB	--	Real	I	--	C	--	Parameter array
GMTLO	--	Real	I	sec	C	--	GMT of lift-off
NEGTIV	--	Intg	I	--	C	NEGTIV	Initial phase angle control flag
K70	--	Intg	I	--	C	WRAP	Flag to wrap initial phase angle
PFT	T_{PF}	Real	I	sec	C	--	Powered flight time
UINS	--	Real	O	rad	A	--	Argument of latitude at insertion
RP	$\bar{R}_P$	Real	O	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	O	fps	C	--	Chaser velocity vector
TP	T_P	Real	O	sec	C	--	Chaser vector time
RT	$\bar{R}_T$	Real	O	ft	C	--	Target position vector

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.11-1.- Concluded

Routine NSERT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
VT	$\bar{V}_T$	Real	O	fps	C	--	Target velocity vector
TT	T_T	Real	O	sec	C	--	Target vector time
DH	--	Real	O	ft	C	--	Delta height at insertion
PA	--	Real	O	rad	C	--	Phase angle at insertion
TINS	--	Real	O	sec	C	--	GMT of insertion

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.12 ROUTINE NAME - Subroutine TARGT

5.12.1 Purpose

TARGT computes the shuttle launch targeting quantities and produces the chaser insertion vector.

5.12.2 Functional Description

TARGT is designed to generate, through analytical equations, a chaser vehicle insertion vector so that it may effect rendezvous with a target vehicle already in orbit at some time later in the mission. Given the target vehicle defined at the time of chaser insertion and a chaser parallel launch insertion vector generated by the routine LENSR, TARGT modifies the chaser vector to compensate for differential nodal precession during the rendezvous. In this way, TARGT is able to minimize the in-orbit plane change delta velocity requirements. In the process of generating the new chaser vector, TARGT also calculates the shuttle launch targeting quantities that are used for on-board guidance equations.

5.12.3 Assumptions and Limitations

None.

5.12.4 Method

The mathematical formulation that TARGT uses is presented in reference 1. Figure 5.12-1 presents a pictorial representation of (a) an inplane launch and (b) an out-of-plane launch.

The parallel launch azimuth (GPAZ) is also computed in TARGT. The equation for this is presented in reference 2. Note that GPAZ is defined at the analytical inplane time.

5.12.5 Routine Input/Output Variables

The TARGT input/output variables are presented in Table 5.12-I.

5.12.6 Functional Logic Flow

The TARGT functional logic flow is presented in Figure 5.12-2.

5.12.7 Diagnostics and Debug

None.

5.12.8 Special Comments

None.

5.12.9 References

1. Kahanek, J. W.: Logic for Real-Time Computation of Skylab Launch Targeting Parameters and Recommended Lift-Off Time. MSC IN 71-FM-21, February 10, 1971.

2. Kahanek, J. W.: Logic and Equations for the Real-Time Computation of the LM Launch Targeting and Display. MSC IN 68-FM-67, March 8, 1968.

TABLE 5.12-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine TARGT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKC	--	Real	I	--	C	--	Parameter array
BLKB	--	Real	I	--	C	--	Parameter array
RP	$\bar{R}_P$	Real	I/O	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	I/O	fps	C	--	Chaser velocity vector
TP	T_P	Real	I/O	sec	C	--	Chaser vector time
RT	$\bar{R}_T$	Real	I	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I	fps	C	--	Target velocity vector
TT	T_T	Real	I	sec	C	--	Target vector time
A0 A1 A2 A3	A_0 A_1 A_2 A_3	Real	I	--	C	A0 A1 A2 A3	Optimum Launch Azimuth Coefficients
DELNOF	--	Intg	I	--	C	--	DELNO computation flag
LATLS	ϕ_{LS}	Real	I	rad	C	LATLS	Geocentric latitude of launch site
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.12-I.- Continued

Routine TARGT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LONGLS	λ_{LS}	Real	I	rad	C	LONGLS	Geographic longitude of launch site
PA	--	Real	I	rad	C	--	Phase angle at insertion
PFT	T_{PF}	Real	I	sec	C	--	Powered flight time
WE	ω_e	Real	I	rad/sec	C	--	Rotation rate of Earth
AZL	--	Real	O	rad	C	--	Optimum launch azimuth
DELNO	--	Real	O	rad	C	--	Angle between target and chaser descending nodes
DELNOD	--	Real	O	rad/sec	C	--	Differential nodal precession
DN	--	Real	O	rad	C	--	Target vector Earth-fixed descending node
GPAZ	--	Real	O	rad	C	--	Gemini parallel launch azimuth
IIGM	--	Real	O	rad	C	--	Inclination of chaser at insertion
TIGM	θ_{IGM}	Real	O	rad	C	--	Angle measured from launch site meridian to chaser descending node
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.12-I.- Concluded

Routine TARGT

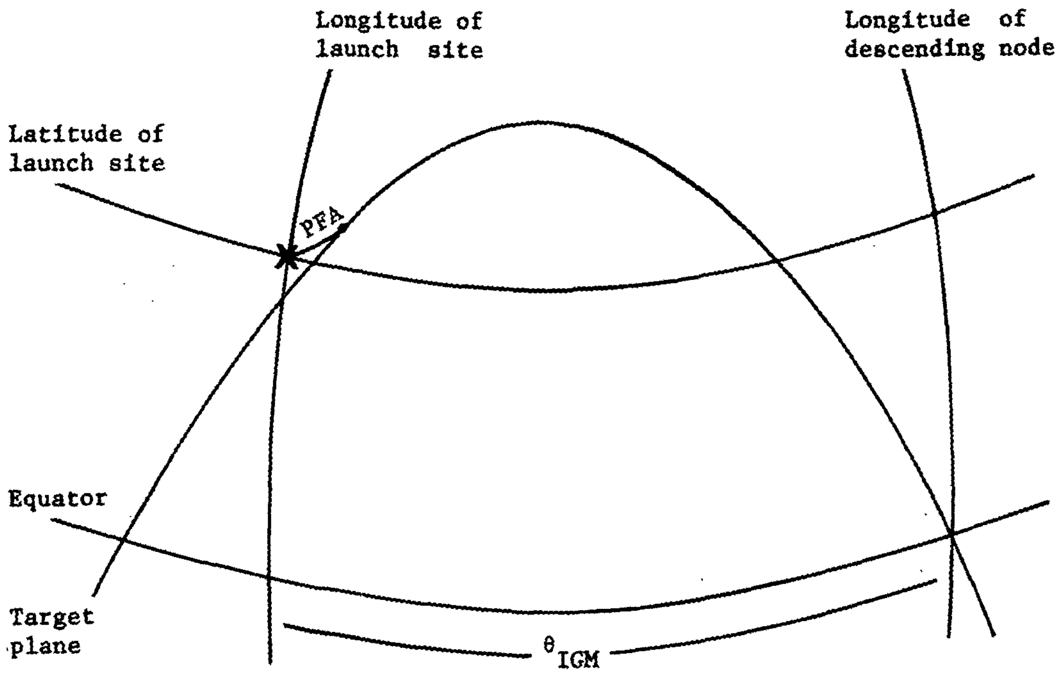
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
TDIGM	$\dot{\theta}_{IGM}$	Real	O	rad/sec	C	--	Rate of change of TIGM
YP	--	Real	O	rad	C	--	Gemini parallel launch wedge angle

NOTES:

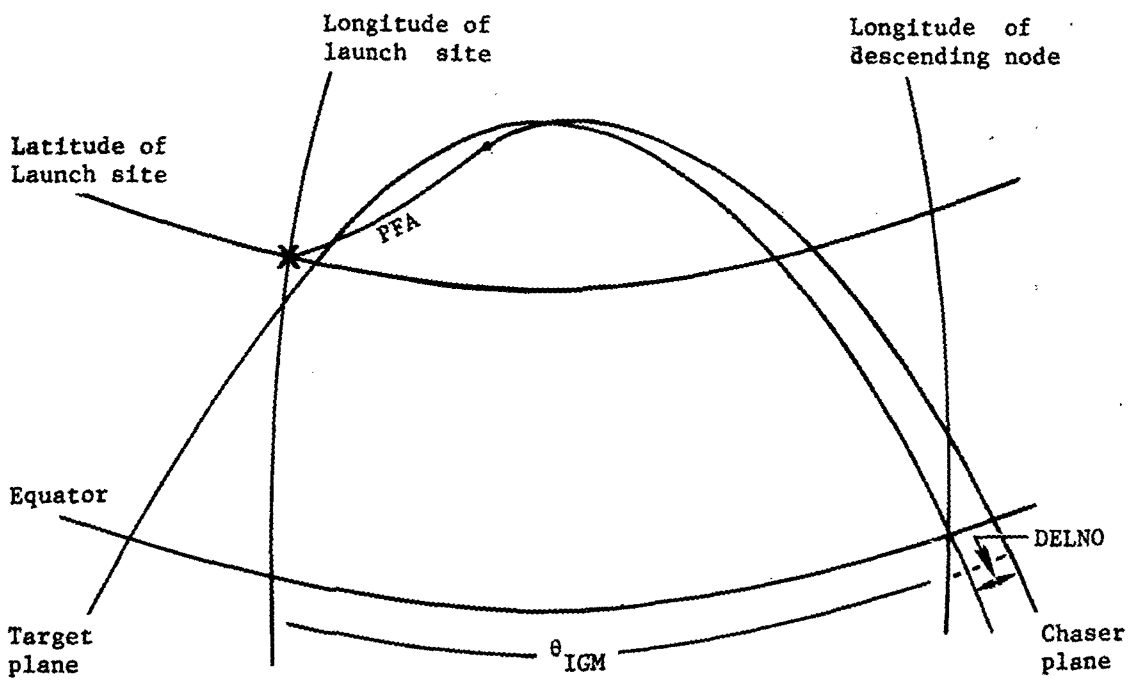
TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



(a) Inplane launch.



(b) Out-of-plane launch.

Figure 5.12-1.- Launch targeting for rendezvous.

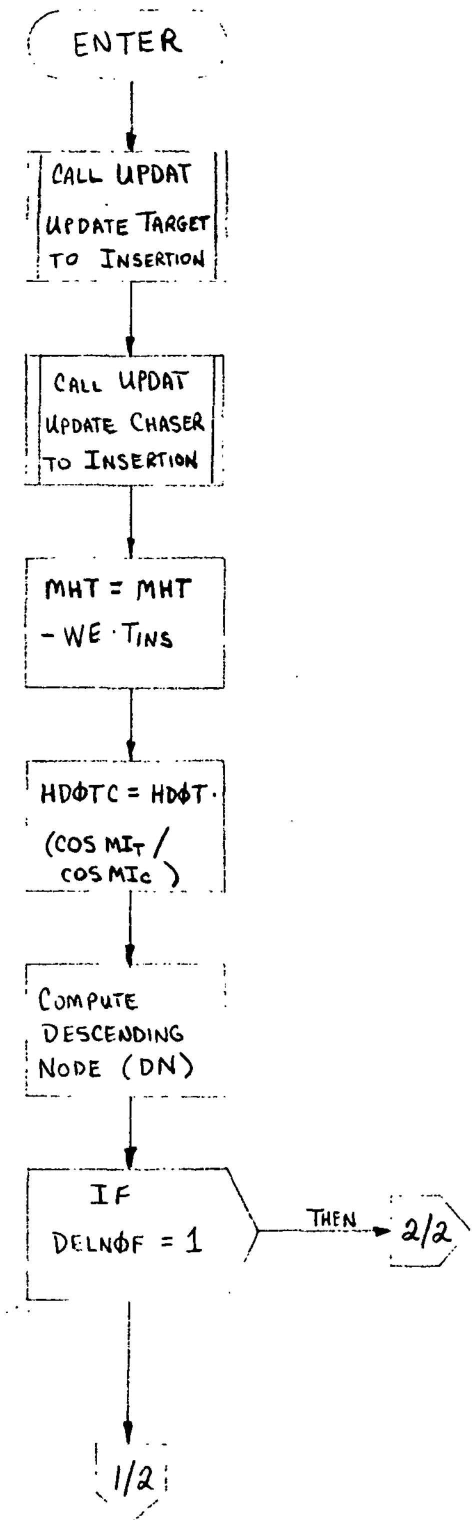


Figure 5.12-2.- TARGT functional logic flow.

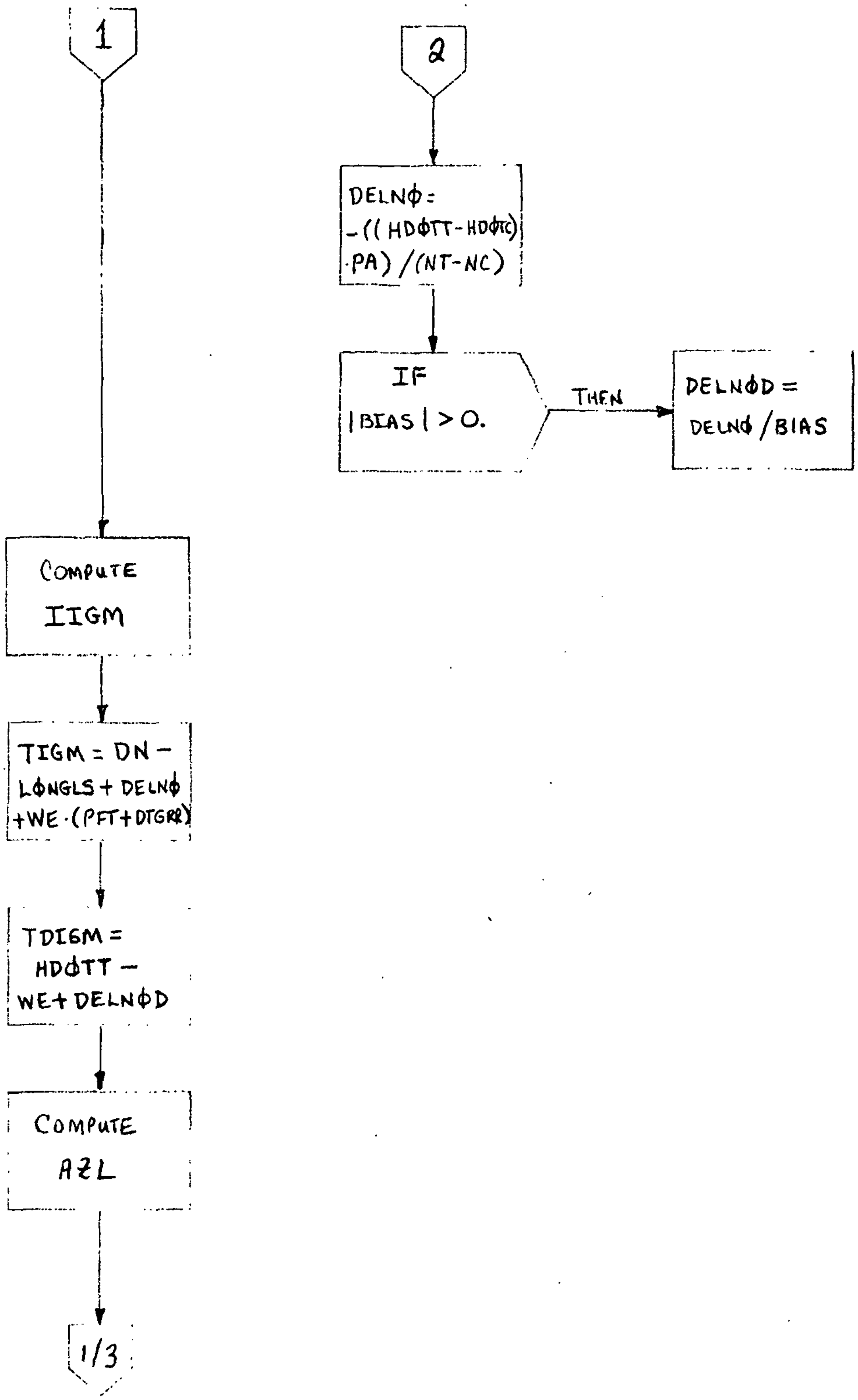
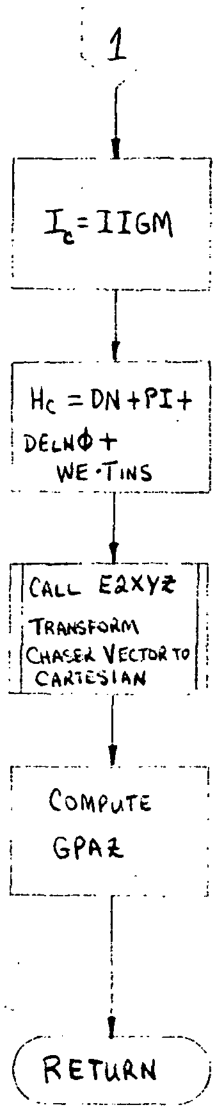


Figure 5.12-2.- Continued.



Page 3 of 3

Figure 5.12-2.- Concluded.

5.13 ROUTINE NAME - Subroutine RLOTD

5.13.1 Purpose

RLOTD is the Recommended Lift-off Time (RLOT) display routine.

5.13.2 Functional Description

The Recommended Lift-off Time Display (RLOTD) routine writes the launch targeting display and the chaser and target vehicle state vectors display. See the reference for format and contents of each.

5.13.3 Assumptions and Limitations

None.

5.13.4 Method

None.

5.13.5 Routine Input/Output Variables

The RLOTD input/output variables are presented in table 5.13-I.

5.13.6 Functional Logic Flow

None.

5.13.7 Diagnostics and Debug

None.

5.13.8 Special Comments

None.

5.13.9 References

None.

TABLE 5.13-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine BLOTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKC	--	Real	I	--	C	--	Parameter array
RP	$\bar{R}_P$	Real	I	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Chaser velocity vector
TP	T_P	Real	I	sec	C	--	Chaser vector time
BLKB	--	Real	I	--	C	--	Parameter array
RT	$\bar{R}_T$	Real	I	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I	fps	C	--	Target velocity vector
TT	T_T	Real	I	sec	C	--	Target vector time
SESCON	--	Intg	I	--	C	ISESCN	Session constants
IBUG	--	Intg	I	--	C	--	Logical unit
FTNM	--	Real	I	ft/n.mi.	C	--	Conversion factor (ft/n. mi.)
RREF	--	Real	I	ft	C	--	Reference radius

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.13-I.- Continued

Routine RLOTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LATLS	ϕ_{LS}	Real	I	rad	C	LATLS	Geocentric latitude of launch site
LONGLS	λ_{LS}	Real	I	rad	C	LONGLS	Geographic longitude of launch site
RINS	R_{INS}	Real	I	ft	C	RINS	Radius at insertion
VINS	V_{INS}	Real	I	fps	C	VINS	Velocity magnitude at insertion
GAMINS	γ_{INS}	Real	I	rad	C	GAMINS	Flightpath angle at insertion
PFA	θ_{PF}	Real	I	rad	C	PFA	Powered fight arc
PFT	T_{PF}	Real	I	sec	C	PFT	Powered flight time
ERROR	--	Intg	I/O	--	C	--	Error return code
BDATE	--	Real	I	Y,M,D	C	--	Base date array (year, month, day)
GMTLO	--	Real	I	sec	C	--	GMT of chaser lift-off
TLO	--	Real	I	sec	C	--	GMT of target lift-off
TPLANE	--	Real	I	sec	C	--	GMT of in-plane lift-off

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.13-I.- Concluded

Routine RLOTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
YP	—	Real	I	rad	C	—	Wedge angle between planes
CONT	—	2CH	I	—	C	"RLOT DISPLAY:"	Display routing flag
RLOT DISPLAY	—	MIX	O	—	T	—	See table 4-IV(h) of this processor for definition of contents.

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.14 ROUTINE NAME - Subroutine LWPOT

5.14.1 Purpose

LWPOT is the Launch Window Processor Output Tables routine.

5.14.2 Functional Description

The Launch Window Processor Output Tables (LWPOT) routine assigns and writes the appropriate data required for the target and chaser phase tables (see reference 1) and the LWP summary table (see table 4-II of this processor).

5.14.3 Assumptions and Limitations

None.

5.14.4 Method

None.

5.14.5 Routine Input/Output Variables

The LWPOT input/output variables are presented in Table 5.14-I.

5.14.6 Functional Logic Flow

None.

5.14.7 Diagnostics and Debug

None.

5.14.8 Special Comments

None.

5.14.9 Reference

1. Flight Design System-1, System Design, Standards. Vol. VI, JSC IN 77-FM-18, rev. 1, January 1978.

TABLE 5.14-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine LWPOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
BLKC	--	Real	I	--	C	--	Parameter array
RP	$\bar{R}_P$	Real	I	ft	C	--	Chaser position vector
VP	$\bar{V}_P$	Real	I	fps	C	--	Chaser velocity vector
TP	T_P	Real	I	sec	C	--	Chaser vector time
BLKB	--	Real	I	--	C	--	Parameter array
RT	$\bar{R}_T$	Real	I	ft	C	--	Target position vector
VT	$\bar{V}_T$	Real	I	fps	C	--	Target velocity vector
TT	T_T	Real	I	sec	C	--	Target vector time
SESCON	--	Intg	I	--	C	--	Session constants
TRGVEC	--	Real	I	--	C	--	Target input state vector
INTBUF	--	Intg	I	--	C	--	Interface table header
LATLS	ϕ_{LS}	Real	I	rad	C	LATLS	Geocentric latitude of launch site

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.14-I.- Continued

Routine LWPOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
LONGLS	λ_{LS}	Real	I	rad	C	LONGLS	Geographic longitude of launch site
RINS	R_{INS}	Real	I	ft	C	RINS	Radius at insertion
VINS	V_{INS}	Real	I	fps	C	VINS	Velocity magnitude at insertion
GAMINS	γ_{INS}	Real	I	rad	C	GAMINS	Flightpath angle at insertion
PFA	θ_{PF}	Real	I	rad	C	PFA	Powered flight arc
PFT	T_{PF}	Real	I	sec	C	PFT	Powered flight time
ERROR	—	Intg	I/O	—	C	—	Error return code
GMTLO	—	Real	I/O	hr	C	—	GMT of chaser lift-off
SPERT	—	Intg	I	—	C	—	State vector propagation flag
TLO	—	Real	I	sec	C	—	GMT of target lift-off
TPLANE	—	Real	I	sec	C	—	GMT of in-plane lift-off

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE:
I = Input
O = Output
I/O = Input/Output

SOURCE:
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.14-I.- Concluded

Routine LWPOT

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
PVTABC	—	Mix	O	—	F	—	Chaser position/velocity phase table. See SSD volume VI, section 1.2.3.4 for definition of contents.
PVTABT	—	Mix	O	—	F	—	Target position/velocity phase table. See SDD volume VI, section 1.2.3.4 for definition of contents.
ISUM	—	Mix	O	—	F	—	Output summary table. See table 4-II of this processor for definition of contents.

NOTES:

TYPE			
Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

5.15 ROUTINE NAME - Subroutine SVDSP

5.15.1 Purpose

SVDSP is the state vector display routine for LWP.

5.15.2 Functional Description

Given the cartesian XYZ state vector, SVDSP computes the transformation from XYZ to spherical and to classical orbital elements and then displays them on the user specified logical unit.

5.15.3 Assumptions and Limitations

None.

5.15.4 Method

None.

5.15.5 Routine Input/Output Variables

The SVDSP input/output variables are presented in Table 5.15-I.

5.15.6 Functional Logic Flow

None.

5.15.7 Diagnostics and Debug

None.

5.15.8 Special Comments

None.

5.15.9 References

None.

TABLE 5.15-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine SVDSP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
X	—	Real	I	ft	A	—	Position vector
XD	--	Real	I	fps	A	--	Velocity vector
T	—	Real	I	sec	A	—	Vector time
CD	--	Real	I	—	A	--	Coefficient of drag
AREA	—	Real	I	ft^2	A	—	Vehicle reference array
WHT	--	Real	I	lb	A	--	Vehicle weight
SESCON	--	Intg	I	—	C	ISESCON	Session constants
IBUG	--	Intg	I	—	C	--	Logical unit
MA	—	Real	I	ft	C	—	Mean semimajor axis
N	--	Real	I	rad/sec	C	--	Mean motion
RREF	—	Real	I	ft	C	--	Reference radius
TINS	--	Real	I	sec	C	--	GMT of insertion

NOTES:

TYPE
Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.15-I.- Concluded

Routine SVDSP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
WE	ω_e	Real	I	rad/sec	C	--	Rotation rate of Earth
DISPLAY PARAMETERS	—	Mix	O	—	T	--	See ~~SDD section 5.2.1,~~ table ~~5.2.1 XIV~~ for definition of contents.
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

MANEUVER ITERATOR PROCESSOR (MANIT)

TO BE SUPPLIED

MATRIX AND ATTITUDE SUPPORT TABLE PROCESSOR (MAST)

1.0 PURPOSE

The matrix option of MAST computes and outputs the transformation matrix from the mean of 1950 (M50) coordinate system to either the IMU coordinate system (REFSMMAT) or the ADI coordinate system (RELMAT). The capability also exists to compute a LVLH bias matrix.

The attitude option of MAST converts the attitude with respect to a specified matrix to an attitude with respect to a desired matrix.

The target identification option of MAST finds the first three celestial targets which are within a specified instruments field-of-view.

2.0 FUNCTIONAL DESCRIPTION

The source or input attitude may be specified by manual input of the attitude or through an attitude timeline phase table (ATL). The MODE parameter defines either the type of initial attitude specified, or a special type of matrix computation. The options for the MODE parameter are as follows:

AT - The initial attitude is from the attitude timeline phase table.

SI - The initial attitude is specified by solar inertial input.

LV - The initial attitude is specified by LVLH input.

IM - The initial attitude is specified by angles with respect to an IMU.

AD - The initial attitude is specified by angles with respect to the ADI.

OO - A new REFSMMAT is to be formed by applying offset orientation angles to the specified REFSMMAT.

DL - A REFSMMAT is to be computed by the dual line-of-sight method. That is, the REFMMAT is defined from two star sightings.

LR - A launch inertial RELMAT is to be computed.

MI - The matrix is to be computed based on an input set of Euler angles.

If the source attitude is provided through the ATL, the processor uses the last entry prior to the time of the matrix computation to define the source attitude. Since the ATL contains only the name of the source matrix, a matrix locker must also be provided, which contains the appropriate matrix.

If the source attitude is defined be solar inertial or by LVLH input, trajectory information is required. In this case, the trajectory is provided through a phase table. The processor uses the last entry prior to the time of the matrix computation to define the initial state vector. The processor then propagates the state vector to the time of the matrix computation. The TYPE parameter defines the type of matrix to be computed, or the attitude or target identification computation. The TYPE options are as follows:

RL - RELMAT matrix computation.

RF - REFSMMAT matrix computation.

LB - LVLH bias matrix computation.

AT - Desired attitude computation.

TI - Target identification computation.

For all MAST modes and types with the exception of the OO, DL, and LR modes and LB types, the initial or source attitude must be specified. For the RELMAT and REFSMMAT matrix computations the source attitude along with the desired IMU cluster angles or ADI attitude angles define the desired orientation of the IMU or ADI in space. The desired attitude computation uses the source attitude and a specified matrix to compute attitudes with respect to the specified matrix. The target identification computation uses the source attitude along with a specified instrument in the Instrument Definition Table to find the first three celestial targets which are within the instruments field-of-view.

The offset orientation (OO) mode re-orients the input REFSMMAT by the input offset angles.

The dual line-of-sight (DL) mode computes the orientation of the IMU from two star sightings.

The launch RELMAT (LR) mode computes a launch inertial alinement. The ADI X-axis is aligned downrange along the launch azimuth, and the ADI Z-axis is directed downward along the local gravity direction (plumbline). The Y-axis completes the right-handed coordinate system.

For the manual input mode (MI) a REFSMMAT, RELMAT, or LVLH bias matrix is computed based on an input set of Euler angles. The rotation sequence for RELMAT and LVLH bias matrices is pitch/yaw/roll. The rotation sequence for the REFMMAT matrix computation is yaw/pitch/roll.

For additional information, refer to JSC IN No. 76-FM-63.

3.0 ASSUMPTIONS AND LIMITATIONS

a. Only elliptical orbit propagation with the AEG is provided.

b. The DL mode and TI type are not supported on FDS-1.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the MAST processor interface parameters is provided in table 4-I. The MODE parameter defines the manner of specifying the input attitudes, or a special type of matrix computation. The required set of inputs for each MODE parameter is as follows:

AT - ATLFIL, LOCKER, GETM

SI - GETM, PTABLE, SCATT, SCSEN, SCOEF

LV - GETM, PTABLE, SCATT, SCSEN

IM - LOCKER, SCATT, SCNAM, IMUDOD

AD - LOCKER, SCATT, SCSEN, SCNAM

OO - LOCKER, SCNAM, DESATT

DL - CELOI, INSOI, IAOI, CAOI, CELOZ, INSOZ, IAOZ, CAOZ, INSDEF, CELTAB

LR - AZIL, LATL, LONL (The GMT of launch is obtained through the SESCON array)

The TYPE parameter identifies whether a matrix is to be generated (RL, RF, of LB), a desired attitude is to be computed (AT), or the target identification (TI) computation is desired.

For all matrix types (RL, R, or LB), the parameters DESATT, DESSEN, and MATNAM must also be provided (for the LB type these are the only required inputs).

The IMUBOD parameter must be provided whenever the type is RF.

For the attitude computation (AT) the parameters LOCKER, DESSEN, and MATNAM must be provided.

If the target identification (TI) mode is requested, the starting star number is provided by CELOI and instrument number is provided by INSOI. The parameters INSDEF and CELTAB must also be provided.

Table 4-II provides a summary of the required inputs for each valid MODE and TYPE combination.

b. Interface table data array definitions - The definition of the input/output data arrays appearing in the MAST interface table is provided in table 4-III.

c. Interface table data file definitions - The format and definitions of the input attitude time-line phase table is provided in table 4-IV.

d. Processor solicited (prompted) inputs - None.

e. Processor displays and display parameter definition table - The format and content of the MAST display is shown in table 4-V and a definition of the display parameters is provided in table 4-VI.

f. Processor message table - The MAST processor error messages are provided in table 4-VII.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VIII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR MAST

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TYPE	AWA	2CH	I	1			Type of matrix for matrix computation, attitude computation, or target identification. "RF" = REFSMMAT "RL" = RELMAT "LB" = LVLH bias "AT" = Attitude computation "TI" = Target identification
MODE	AWA	2CH	I	1	1		Mode of inputting attitude, or indication of special matrix computation. AT - ATL input SI - Solar inertial attitude input LV - LVLH attitude input IM - IMU cluster angles input AD - ADI attitude angles input OO - REFSMMAT offset orientation computation DL - REFSMAT dual line-of-right computation LR - Launch RELMAT computation MI - Manual input Euler angles sequence
ATLFIL	Disk	Free	I				Name of ATL required when MODE = "AT"
LOCKER	AWA	Free	I	25	25		Source matrix locker column
GETM	AWA	Real	I	6	3		Elapsed time from reference time for matrix computation (hrs, min, sec)

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR MAST

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
PTABLE	AWA	Real	I	60	30		Input position/velocity phase table column
SCATT	AWA	Real	I	6	3		Source attitude angles (roll, pitch, yaw)
SCSEN	AWA	2CH	I	1	1	"+X"	Source attitude sense. Indicater direction of ADI zero setting (+X, -X or - Z)
SCNAM	AWA	6CH	I	3	1		Name of source REFSMMAT or RELMAT
DESATT	AWA	Real	I	6	3		Desired IMU cluster angles, IMU offset angles, or ADI attitude angles. (roll, pitch, yaw)
DESSEN	AWA	2CH	I	1	1	"+X"	Desired attitude sense. Indicater direction of ADI zero setting (+X, -X or -Z)
MATNAM	AWA	6CH	I	3	1		Name of output matrix or name of matrix for desired attitude computation
Note: The following 10 parameters are required for REFSMMAT mode "DL"							
CEL01	AWA	Intg	I	1	1	DUMMY	Celestial target identifier of first target

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR MAST

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
INS01	AWA	Intg	I	3	1	DUMMY	Instrument identifier for first target sighting
IA01	AWA	Real	I	4	2	DUMMY	Instrument angles for first sighting
CA01	AWA	Real	I	4	2	DUMMY	IMU cluster angles for first sighting
CEL02	AWA	Intg	I	1	1	DUMMY	Celestial target identifier for second target
INS02	AWA	6CH	I	3	1	DUMMY	Instrument identifier for second target sighting
IA02	AWA	Real	I	4	2	DUMMY	Instrument angles for second sighting
CA02	AWA	Real	I	6	3	DUMMY	IMU cluster angles for second sighting
INSDEF	AWA	Free	I	250	(25,10)	DUMMY	Instrument definition table
CELTAB	AWA	Real	I	1200	(3,200)	DUMMY	Celestial target table
AZIL	AWA	Real	I	2	1		Launch azimuth required when MODE = "LR"
IMUBOD	AWA	REAL	I	18	9		IMU to case transformation matrix

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR MAST

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array
SESCON	AWA	Free	I	90	90	!!SESCN	Session constants array
PROCON	AWA	Free	I	5	5		Processor constants array
SCOEF	AWA	Real	I	44	22	!SCOEF	Sun coefficients for analytic ephemeris model
OUTMAT	AWA	Free	O	25	25		Output matrix

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- SUMMARY OF REQUIRED INPUTS

TYPE	RF	RF	RF	RF	RF	RF	RF	RL	RL	RL	RL	RL	RL
MODE	AT	SI	LV	IM	AD	OO	DL	AT	SI	LV	IM	AD	LR
ATLFIL	X							X					
LOCKER	X			X	X	X		X			X	X	
GETM	X	X	X					X	X	X			
PTABLE	(a)	X	X					(a)	X	X			
SCATT		X	X	X	X	X			X	X	X	X	
SCSEN		X	X		X				X	X		X	
SCNAM				X	X	X					X	X	
DESATT				X	X						X	X	
DESSEN					X							X	
MATNAM	X	X	X	X	X	X	X	X	X	X	X	X	X
CEL01							X						
INS01							X						
IA01/CA01 CEL02/INS02 IA02/CA02							X						

(a) Required when ATL entry is "LV" or "SI"

TABLE 4-II.- Continued

TYPE	RF	RF	RF	RF	RF	RF	RF	RL	RL	RL	RL	RL	RL
MODE	AT	SI	LV	IM	AD	OO	DL	AT	SI	LV	IM	AD	LR
INSDEF							X						
CELTAB							X						
AZIL													X
IMUBOD	X	X	X	X	X	X	X	(c)			X		
SCOEF	(b)	x						(b)	X				

(b) Required when ATL entry is "SI"
(c) Required when ATL matrix entry is "RL"

TABLE 4-II.- Continued

TYPE	AT	AT	AT	AT	AT	TI	TI	TI	TI	TI	MI	MI	MI
MODE	AT	SI	LV	IM	AD	AT	SI	LV	IM	AD	RL	RF	LB
ATLFIL	X					X							
LOCKER	X					X							
GETM	X					X							
PTABLE	(a)	X	X			(a)	X	X					
SCATT		X	X	X	X		X	X	X	X	X	X	X
SCSEN		X	X		X		X	X		X	X		X
SCNAM				X	X				X	X			
DESATT													
DESSEN													
MATNAM	X	X	X	X	X						X	X	X
CEL01						X	X	X	X	X			
INS01						X	X	X	X	X			
IA01/CA01 CEL02/INS02 IA02/CA02													

(a) Required when ATL entry is "LV" or "SI"

TABLE 4-II.- Concluded

TYPE	AT	AT	AT	AT	AT	TI	TI	TI	TI	TI	MI	MI	MI
MODE	AT	SI	LV	IM	AD	AT	SI	LV	IM	AD	RL	RF	LB
INSDEF						X	X	X	X	X			
CELTAB						X	X	X	X	X			
AZIL													
IMUBOD				X					X				
SCOEF	(b)	X				(b)	X						

(b) Required when ATL entry is "SI"

TABLE 4-III.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR MAST

Array name	Index location	Default value	Definition
LOCKER	1-18		Elements (by column) of matrix (real)
	19-21		Matrix name
	22-24		Unused
	25		Matrix code ("RF", "RL", or "LB")
PTABLE	1-30		Position/Velocity phase table column
GETM	1		Time of matrix relative to reference time, hours
	2		Time of matrix relative to reference time, minutes
	3		Time of matrix relative to reference time, seconds
SCATT	1		Source attitude roll angle
	2		Source attitude pitch angle
	3		Source attitude yaw angle
DESATT	1		Desired IMU or ADI roll angle
	2		Desired IMU or ADI pitch angle
	3		Desired IMU or ADI yaw angle
IA01	1		First instrument angle; first sighting
	2		Second instrument angle; first sighting
CA01	1		IMU roll cluster angle; first sighting
	2		IMU pitch cluster angle; first sighting
	3		IMU yaw cluster angle; first sighting
IA02	1		First instrument angle; second sighting
	2		Second instrument angle; second sighting
CA02	1		IMU roll cluster angle; second sighting
	2		IMU pitch cluster angle; second sighting
	3		IMU yaw cluster angle; second sighting
IMUBOD	1-9		Elements (by column) of IMU to body matrix

TABLE 4-III.- Concluded

PROCESSOR MAST

Array name	Index location	Default value	Definition
GLOCON	1-180		Global constants array
SESCON	1-190		Session constants array
PROCON	1	0	Debug print flag 0 = no debug print 1 = debug print
	2	0	Output logical unit 0 = same as input unit Positive number = logical unit of output
	3-4	S.E-6	Computation tolerance, radians
	5	20	Cartridge reference number for data files
OUTMAT	1-18		Elements (by column) of output matrix
	19-21		Name of output matrix
	22-24		Not used
	25		Matrix code of the output matrix

TABLE 4-IV.- INTERFACE TABLE DATA FILE DEFINITIONS

PROCESSOR MAST

DRDE DATA FILE ATLFIL

Record number	Integer word allocations	Content and definition
1	1-3	Processor creating file.
	4-6	Interface table variable creating file.
	7-9	Processor last changing file.
	10-12	Interface table last changing file.
2-N	1-2	Greenwich mean time of entry.
	3-20	Elements (by column) of transformation matrix from M50 to the body axes for IH and ROTR modes. On from LVLH or SI to the body axes for LVLH and SI modes.
	21-26	Roll, pitch, yaw attitude angles.
	27-29	Source matrix name (6 characters).
	30-37	Comments used to identify entry (16 characters).
	38-40	Attitude hold mode (6 characters).
	41-44	Vehicle pitch and yaw angles from the vehicle X-axis to define the Eigen axis
	45-46	Eigen axis rate.
	47-50	Not used.

TABLE 4-V.- PROCESSOR DISPLAY TABLE

PROCESSOR MAST

```
                        MATRIX  AND  ATTITUDE  SUPPORT  TABLE

          MODE αα     ATL αααα    GMTR XXX XX XX XX   GMT XXX XX XX XX
          TYPE αα  LOCKER αααααα                LOS1 DEF          LOS1 CLUSTER
          SOURCE MATRIX αααααα                  TGT ID    XXX      ANGLES
                    R XXX.XX                    INST ID   ααα      R XXX.XX
          SENSE αα  P XXX.XX                    INST AND ±XXX.XX   P XXX.XX
                    Y XXX.XX                    INST AND ±XXX.XX   Y XXX.XX

          PTABLE αααααα                         LOS2 DEF          LOS2 CLUSTER
          LVLH.PET XXX XX XX XX                 TGT ID    XXX      ANGLES
               GMT XXX XX XX XX                 INST ID   ααα      R XXX.XX
          LIR   AZ XXX.XX                       INST AND ±XXX.XX   P XXX.XX
                                                INST AND ±XXX.XX   Y XXX.XX
          DESIRED MATRIX αααααα
                     R XXX.XX
          SENSE αα   P XXX.XX                   TGT IDENT
                     Y XXX.XX                    INST ID αα
                                                 STAR ID XXX
          OUTPUT MATRIX  αααααα                          HORIZ     VERT
          ±X.XXXXXX ±X.XXXXXX ±X.XXXXXX          STAR1 ±XXX.X   ±XXX.X
          ±X.XXXXXX ±X.XXXXXX ±X.XXXXXX          STAR2 ±XXX.X   ±XXX.X
          ±X.XXXXXX ±X.XXXXXX ±X.XXXXXX          STAR3 ±XXX.X   ±XXX.X
```

TABLE 4-VI.- DISPLAY PARAMETER DEFINITION TABLE FOR THE MATRIX AND ATTITUDE SUPPORT TABLE

PROCESSOR MAST

MATRIX AND ATTITUDE SUPPORT TABLE

Display parameter label	Parameter definition
MODE	Input mode parameter
TYPE	Input type parameter
ATL	Name of attitude timeline DRDE
LOCKER	Name of input matrix locker
SOURCE MATRIX	Name of source attitude matrix
SENSE	Source attitude sense
R	Source attitude roll
P	Source attitude pitch
Y	Source attitude yaw
PTABLE	Name of input phase table
LVLH PET	Time from reference time of LVLH coordinate system (days, hrs, min, sec)
GMT	GMT of LVLH coordinate system (days, hrs, min, sec)
LIR AZI	Launch inertial RELMAT azimuth
DESIRED MATRIX	Name of desired matrix
SENSE	Desired attitude sense
R	Desired attitude roll
P	Desired attitude pitch
Y	Desired attitude yaw
OUTPUT MATRIX	Elements of output matrix
GMTR	GMT of reference time (days, hrs, min, sec)
GMT	GMT of matrix computation (days, hrs, min, sec)
LOS1 DEF	
TGT ID	Target identifier } Required for first sighting in the dual line-of-sight mode.
INST ID	Instrument identifier } Required for first sighting in the dual line-of-sight mode.
INST ANG	Instrument angles } Required for first sighting in the dual line-of-sight mode.
LOS1 CLUSTER ANGLES	
R	Roll } Cluster angles associated with first sighting in the dual line-of-sight mode.
P	Pitch } Cluster angles associated with first sighting in the dual line-of-sight mode.
Y	Yaw } Cluster angles associated with first sighting in the dual line-of-sight mode.
LOS2 DEF	
TGT ID	Target identifier } Required for second sighting in the dual line-of-sight mode.
INST ID	Instrument identifier } Required for second sighting in the dual line-of-sight mode.
INST ANG	Instrument angles } Required for second sighting in the dual line-of-sight mode.

TABLE 4-VI.- Concluded

PROCESSOR MAST

MATRIX AND ATTITUDE SUPPORT TABLE

Display parameter label	Parameter definition
LOS2 CLUSTER ANGLES	
R	Roll } Cluster angles associated with second sighting in the dual line-of-sight mode.
P	Pitch
Y	Yaw
TGT IDENT	Target identification mode.
INST ID	Instrument identifier.
STAR ID	Starting star identifier.
HORIZ VERT	
STAR 1	Horizonal and vertical displacement angles for the first three stars that are within the instrument field-of-view for the target identification mode
STAR 2	
STAR 3	

TABLE 4-VII.- PROCESSOR MESSAGE TABLE

PROCESSOR MAST

MSG no.	Message ID block	Message text block and explanation
1	MAST	INVALID MODE OR TYPE; MODE = αα TYPE = αα Meaning: Illegal mode or type specified. Severity: Processor terminates. Action required by user: Specify valid mode or type.
2	MAST	OPEN FILE ERROR ON ATL FILE; NAME = αααα Meaning: File open error on ATL. Severity: Processor terminates. Action required by user: Check validity of ATL name.
3	MAST	FILE SPECIFIED NOT ATL; NAME = αααα Meaning: Specified file is not on ATL. Severity: Processor terminates. Action required by user: Check validity of ATL name.
4	MAST	READ FILE ERROR ON ATL FILE; NAME = αααα Meaning: Read file error on ATL. Severity: Processor terminates. Action required by user: Seek aid from SDB.
5	MAST	NO ENTRIES IN ATL; NAME = αααα Meaning: No valid entries in ATL. Severity: Processor terminates . Action required by user: Check validity of ATL.
6	MAST	PHASE TABLE VECTOR CODE = XXXX; NOT TEG Meaning: Position/velocity vector code is not TEG Severity: Processor terminates. Action required by user: Specify TEG vector in phase table.

TABLE 4-VII.- Concluded

PROCESSOR MAST

MSG no.	Message ID block	Message text block and explanation
7	MAST	SOURCE MATRIX αααααα NOT FOUND IN αααααα Meaning: Source matrix not found in specified matrix locker. Severity: Processor terminates. Action required by user: Check names specified for source matrix and matrix locker.
8	MAST	SESCON PARAMETERS ARE NOT CONSISTENT WITH PHASE TABLE Meaning: SESCON initialization data are not consistent with data in phase table. Severity: Processor terminates. Action required by user: Initialize data for SESCON and phase table with the same data.

TABLE 4-VIII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR MAST

Processor name	Processor abstract prompt (maximum 256 characters)
MAST	The matrix and attitude support table computer REFSMMAT, RELMAT, and LVLH bias matrices. MAST also converts an input attitude to an attitude with respect to a specified matrix.
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
	The processor extended prompts are the same as the definitions provided in table 4-II, with the following exceptions:
MODE	Mode of inputting the source altitude, or an indication of a special matrix computation. The options are AT, SI, LU, IM, AD, OO, DL, and LR.
PROCON	Processor constants: (1) Debug print flag (0 = no print, 1 = debug print) (2) Output LU (3-4) Computation tolerance (5) Cartridge reference for data files

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

MASTER DATA TEMPORARY PRINT PROCESSOR (MDTP)

1.0 PURPOSE

The purpose of MDTP is to provide a means for obtaining a formatted listing of the global constants array (GLOCON) and the session constants array (SESCON).

2.0 FUNCTIONAL DESCRIPTION

The processor interface table for MDTP contains only three parameters: GLOCON, SESCON, and PRINT. The first parameter (GLOCON) is an input through which the name of the global constants array is supplied; this input is defaulted to the name !!GLCN, and unless the user has renamed the global constants array within his AWA, this input should not be changed. The second parameter (SESCON) is an input through which the name of the session constants array is supplied; this input is defaulted to the name !SESCN, and unless the user has renamed the session constants array within his AWA, this input should not be changed. The third parameter (PRINT) is a position independent Hollerith input array through which the user defines which master data element(s) (GLOCON, SESCON) are to be listed. This is done by setting PRINT(1) and PRINT(2) to the values "GLOCON" and/or "SESCON", or "SESCON" and/or "GLOCON", the order (position) makes no difference. For example, the following inputs:

```
\PRINT = "GLOCON", "SESCON"
or
\PRINT = "SESCON", "GLOCON"
```

will both produce the same result; i.e, in both cases, the global constants array and the session constants array will be listed. Only the mnemonics "GLOCON" and "SESCON" are recognized by the processor; use of any other mnemonic will not cause the data element to be listed. For example, if the user wanted to list only the session constants array, then the input could be any of the following:

```
\PRINT = "SESCON", "b̸"
or
\PRINT = "XXX", "SESCON"
or
\PRINT = "SESCON", "AAA"
```

3.0 ASSUMPTIONS AND LIMITATIONS

a. If the PRINT flag value is set to cause print of a master data element (i.e., either !!GLCN, or !SESCN), then that master data element must exist in the AWA at the time of processor execution. If the PRINT flag value indicates that a master data element is not to be listed, then that array does not have to exist in the AWA at processor execution time.

b. A request to print any data element other than "GLOCON" or "SESCON" will not produce any output.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the default interface table for the MDTP processor is provided in table 4-I.

b. Interface table data array definitions - A definition of the input and output data arrays appearing in the MDTP interface table is provided in table 4-II.

c. Interface table data file definitions - None.

d. Processor solicited (prompted) input - None.

e. Processor displays and display parameter definition tables - The display that presents the "GLOCON" array is defined in table 4-III(a), and the display parameter definitions for "GLOCON" are presented in table 4-III(b). The display that presents the "SESCON" array is defined in table 4-III(c), and the display parameter definitions for "SESCON" are presented in table 4-III(d). The parameter labeling and definitions are consistent with the definitions and usage in the data standards specified by FDS-1 System Design Document, Volume VI - Processor Interface Standards.

f. Processor message table - None.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-IV.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR MDTP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array; master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
PRINT	AWA	6CH	I	6	2		Position independent Hollerith print flag; two names may be entered, the options are: "GLOCON", causes global constant array to be listed "SESCON", causes session constants array to be listed

NOTES

CLASS	TYPE			USE
AWA	Free	2CH	72CH	I = Input
Disk	Intg	6CH	Mix	O = Output
	Real	18CH	Symb	I/O = Input/Output
	Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR MDTP

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array, master data base element; see JSC IN 78-FM-60, volume I, table 7.2-III for definition of array locations.
SESCON	(1) . . . (90)	!SESCN	Session constants array; see JSC IN 78-FM-60, volume I, table 7.2-II for definition of array locations.
PRINT	(1)		Print flag array position 1, a value of "GLOCON" or "SESCON" will cause the corresponding data element to be listed; any other values will not be recognized by the processor, and no data element will be listed.
	(2)		Print flag array position 2; a value of "GLOCON" or "SESCON" will cause the corresponding data element to be listed; any other values will not be recognized by the processor, and no data element will be listed.

TABLE 4-III.- PROCESSOR DISPLAYS AND DISPLAY DEFINITION TABLES

(a) Global constants array GLOCON

PROCESSOR MDTP

```
 1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75
 GLOBAL CONSTANTS ARRAY GLOCON
 GLCN(  1)   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX
 GLCN(  9)         .                .                .                .
 GLCN( 17)         .                .                .                .
 GLCN( 25)         .                .                .                .
 GLCN( 33)         .                .                .                .
 GLCN( 41)         .                .                .                .
 GLCN( 49)         .                .                .                .
 GLCN( 57)   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX
 GLCN( 65)   ±.XXXXXXXXXXXXXD±XX   ±.XXXXXXXXXXXXXD±XX
 GLCN( 71)   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX
 GLCN( 87)         .                .                .                .
 GLCN( 95)         .                .                .                .
 GLCN(103)         .                .                .                .
 GLCN(111)         .                .                .                .
 GLCN(119)         .                .         ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX
 GLCN(127)   ±.XXXXXXXXE±XX   ±.XXXXXXXXE±XX
 GLCN(131)    aaaa   aaaa   aaaa   aaaa   aaaa
 GLCN(151)      .      .      .      .      .
 GLCN(141)      .      .      .      .      .
 GLCN(151)      .      .      .      .      .
 GLCN(161)      .      .      .      .      .
 GLCN(171)    aaaa   aaaa   aaaa   aaaa   aaaa
     5    10   15   20   25   30   35   40   45   50   55   60   65   70   75
```

TABLE 4-III.- Continued

(b) Display parameter definition table for the global constants array GLOCON

PROCESSOR MDTP

GLOBAL CONSTANTS ARRAY GLOCON	
Display parameter label	Parameter definition
GLCN	Global constants array, master data base element; see JSC IN 78-FM-60, volume I, table 7.2-III for definition of array locations.

TABLE 4-III.- Continued

(c) Session constants array SESCON

PROCESSOR MDTP

```
SESSION CONSTANTS ARRAY SESCON
SESCN(  1)   ±XXXXX    ±XXXXX    ±XXXXX    ±XXXXX    ±XXXXX    ±XXXXX
SESCN(  7)   ±.XXXXXXXXE±XX      ±.XXXXXXXXE±XX      ±.XXXXXXXXE±XX
SESCN( 13)    aaaa      aaaa      aaaa      aaaa      aaaa      aaaa      aaaa
SESCN( 27)   ±.XXXXXXXXEXXX      ±.XXXXXXXXE±XX      ±.XXXXXXXXE±XX
SESCN( 33)         .                   .                   .
SESCN( 39)         .                   .                   .
SESCN( 45)         .                   .                   .
SESCN( 51)         .                   .                   .
SESCN( 57)         .                   .                   .
SESCN( 63)         .                   .                   .
SESCN( 69)         .                   .                   .
SESCN( 75)         .                 ±.XXXXXXXXE±XX      ±.XXXXXXXXE±XX
SESCN( 81)   ±.XXXXXXXXE±XX      ±.XXXXXXXXXXXXD±XX
```

TABLE 4-III.- Concluded

(d) Display parameter definition table for the session constants array SESCON

PROCESSOR MDTP

SESSION CONSTANTS ARRRAY SESCON	
Display parameter label	Parameter definition
SESCN	Session constants array; see JSC IN 78-FM-60, volume I, table 7.2-II for definition of array locations.

TABLE 4-IV.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR MDTP

Processor name	Processor abstract prompt (maximum 256 characters)
MDTP	The purpose of the MDTP processor is to provide a means of obtaining a formatted listing of the contents of the global constants array (GLOCON) and/or the session constants array (SESCON).
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
GLOCON	Name of the global constants array to be displayed; normally defaulted to !!GLCN
SESCON	Name of the session constants array to be displayed; normally defaulted to !SESCN
PRINT	Position independent Hollerith print flag. Two names may be entered; the options are "GLOCON", and "SESCON".

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

MISSION PLAN TABLE PROCESSOR (MPTP)

1.0 PURPOSE

The mission plan table processor displays maneuvers in a mission plan table (MPT) that were calculated by various FDS maneuver planning processors and stored in a position/velocity phase table.

2.0 FUNCTIONAL DESCRIPTION

The MPTP has the capability to generate a maneuver data display (the MPT) that describes a series of maneuvers that were originally computed by other FDS processors. The display includes: (1) the name of the phase table used for the display, (2) the base date, (3) the Greenwich mean time (GMT) of lift-off, (4) the ground elapsed time (GET) at burn initiation, (5) the delta-time (delta-T) since the previous MPT maneuver, (6) the delta-velocity (delta-V) of the maneuver, (7) the delta-V remaining after the maneuver, (8) the height of apogee, (9) the height of perigee, and (10) a code containing the vehicle ID, engine ID, and maneuver ID.

Provided a position/velocity phase table, the initial propellent loads, and ISP's, the MPTP calculates the ground elapsed time, delta-V remaining, height of apogee, and height of perigee.

3.0 ASSUMPTIONS AND LIMITATIONS

a. MPTP will display a maximum of 10 maneuvers

b. MPTP will display maneuvers only up to the first 'DD' vector it encounters in the phase table

c. MPTP will display maneuvers for only one vehicle

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the MPTP interface table parameters is provided in table 4-I.

GLOCON is a set of constants, universal to all processors, that is maintained in the master data base. MPTP accesses this array for the constants it needs. The default values are stored in !!GLCN.

SESCON is a set of session related constants generated by the user upon execution of the system utility processor BASTM. The default values are stored in !SESCN.

LIST is a user-defined variable that identifies the logical unit of the output device to which all printed output is written. The default value of zero is a flag that sets the user's terminal as the output device.

PVTAB is a phase table containing a set of up to 20 user-specified position/velocity vectors that are used in generating the MPT. PVTAB may be subscripted to indicate the starting vector.

NUMANV is a user-defined variable that specifies the number of maneuvers that are to be displayed in the MPT.

OMSP is a user-defined variable specifying the initial OMS propellant load. The default value is zero.

RCSP is a user-defined variable specifying the initial RCS propellant load. The default value is zero.

SUSP is a user-defined variable specifying the initial SUS propellant load. The default value is zero.

OMSISP is a user-defined variable specifying the specific impulse for OMS in seconds.

RCSISP is a user-defined variable specifying the specific impulse for RCS in seconds.

SUSISP is a user-defined variable specifying the specific impulse for SUS in seconds.

VEHID is a two-character user-defined maneuver vehicle identification code that becomes part of the maneuver code display parameter.

b. Interface table data array definitions - The definition of the input data arrays appearing in the MPTP interface table is provided in table 4-II.

c. Interface table data file definitions - None.

d. Processor solicited (prompted) inputs - None.

e. Processor displays and display parameter definition table - The MPTP generates a mission plan table display. The format of this display is shown in table 4-III and an example of the MPTP display is provided in table 4-IV. A definition of the display variables is provided in table 4-V. All display parameters are in user-selected external units except: GMTLO, GETBI, DELTAT, which are in hours, minutes, and seconds, and DATE, which is in month, day, and year. Code is built in the following order: 2-CHAR vehicle ID, 1-CHAR engine ID, and a 4-CHAR maneuver ID. Vehicle ID may be any 2-CHAR. Engine ID codes are as follows: S-solid rocket motors, M-main engine, O-orbital maneuvering system, R-reaction control system-AFT and FWD. Maneuver ID codes are as follows: Zero - no maneuver codes; all other maneuver codes are currently undefined and will be displayed as blanks.

f. Processor message table - In general, MPTP does not generate any messages to the user during execution. However, in the cases of certain types of user input errors, an error message may be displayed to the user. The messages the user may expect to see in those cases are shown in table 4-VI.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VII.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR MPTP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
LIST	AWA	Intg	I	1	1	0	Logical unit of output device onto which mission plan table is written; default value of zero is a flag that sets the user's terminal as the output unit
PVTAB	AWA	Real	I	1200	30		Phase table containing up to 20 position/velocity vectors; subscripts may be used to indicate starting vector
NUMANV	AWA	Intg	I	1	1		Number of maneuvers to be displayed in the MPT. Maximum value is 10
OMSP	AWA	Real	I	2	1	0.	Initial OMS propellant load
RCSP	AWA	Real	I	2	1	0.	Initial RCS propellant load
SUSP	AWA	Real	I	2	1	0.	Initial SUS propellant load
OMSISP	AWA	Real	I	2	1		Specific impulse for OMS, seconds
RCSISP	AWA	Real	I	2	1		Specific impulse for RCS, seconds

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR MPTP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
SUSISP	AWA	Real	I	2	1		Specific impulse for SUS, seconds
VEHID	AWA	2CH	I	1	1		Vehicle identification code

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR MPTP

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array master data base element; see table 7.2-III in JSC IN 78-FM-60, volume I for definition of contents.
SESCON	(1) . . . (90)	!SESCN	Input/output session constants array; see table 7.2-II(a) in JSC IN 78-FM-60, volume I for definition of contents.
PVTAB	(1,1) . . . (30,20)		Position/velocity phase table; see figure 7.2-13 in JSC IN 78-FM-60, volume I for definition of contents. PVTAB holds a maximum of 20 position/velocity state and propagation data vectors.

TABLE 4-III.- PROCESSOR DISPLAY TABLE

PROCESSOR <u>MPTP</u>

```
                                  MISSION  PLAN  TABLE
  DATE   αααα XX, XXXX                                                    GMTLO   XXX  XX  XX.
                                  PHASE  TABLE  αααααα
      GETBI          DELTAT          DELTAV        DYREM              HA               HP          CODE
±XXX  XX  XX.X              ±XXXXX.X   ±XXXXX.X   ±XXXXXXXXX.XX  ±XXXXXXXXXX.X     αααααα
             .±XXX  XX  XX.X
```

TABLE 4-IV.- MPTP DISPLAY EXAMPLE

MISSION PLAN TABLE

DATE OCT 31,1979 GMTLO 16 30 .0

PHASE TABLE CASE1

GETBI	DELTAT	DELTAV	DVREM	HA	HP	CODE
0 8 22.0		100.0	-99.9	108.28	40.32	OMOZERO
	0 27 37.6					
0 35 59.6		443.5	-543.2	108.94	33.05	OMOZERO
	0 34 .4					
1 10 .0		423.6	-966.5	105.40	40.00	OMOZERO
	0 47 2.7					
1 57 2.7		159.0	-1125.4	104.94	98.18	OMOZERO
	0 25 5.5					
2 22 8.2		509.8	-1634.8	399.96	99.89	OMOZERO

TABLE 4-V.- DISPLAY PARAMETER DEFINITION TABLE

PROCESSOR MPTP

MISSION PLAN TABLE

Display parameter label	Parameter definition
DATE	Base date
GMTLO	GMT of lift-off
PHASE TABLE	Name of phase table used in generating the MPT
GETBI	GET at burn initiation; hours, minutes, seconds
DELTAT	The delta-T since the previous MPT maneuver; hours, minutes, seconds.
DELTAV	The delta-V of the maneuver, in user defined external units
DVREM	The delta-V remaining for the engine identified in code, after the maneuver. DVREM is displayed in user defined external units
HA	Predicted height of apogee after the maneuver in user defined external units
HP	Predicted height of perigee after the maneuver in user defined external units
CODE	Code containing: 2-CHAR maneuver vehicle ID., 1-CHAR engine ID., 4-CHAR maneuver ID.

TABLE 4-VI.- PROCESSOR MESSAGE TABLE

PROCESSOR MPTP

MSG no.	Message ID block	Message text block and explanation
1	*MPTP*	INVALID INPUT VALUE FOR NUMANV = XXXX. Meaning: The user has assigned an incorrect value to NUMANV. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use Interface Table Editor to enter a value between 1 and 10; then resume execution.
2	*MPTP*	NO 'DD' VECTOR IN PHASE TABLE. Meaning: A search of the phase table from the user specified beginning vector has turned up no data documentation vector. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: User the Interface Table Editor to enter a valid phase table name in response to PVTAB.
3	*MPTP*	NUMANV .GT. NUMBER OF AVAILABLE MANEUVERS. Meaning: User requested more maneuvers displayed in the mission plan table than were provided in the phase table. Severity: This is a warning message only and is displayed following execution of MPTP. Action required by user: None
4	*MPTP*	INCOMPATIBLE UNITS (SESCON/PVTAB). REFER TO PTP PROCESSOR. Meaning: Units specified in the 'DD' vector of PVTAB do not correspond to the SESCON units. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Refer to the phase table print processor to display phase table information.
5	*MPTP*	PVTAB VECTOR NOT TEG/CARTESIAN. Meaning: Incorrect reference axis, element set, and/or element type. Severity: Processor will terminate and control will be returned to the FDS Executive Action required by user: Transform vector to TEG/Cartesian and resume execution.

TABLE 4-VI.- Concluded

PROCESSOR MPTP

MSG no.	Message ID block	Message text block and explanation
6	*MPTP*	INCORRECT TYPE. PHASE TABLE αααααα MUST BE TYPED REAL. Meaning: The phase table is not typed Real in the AWA. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to enter the name of a correctly typed phase table.
7	*MPTP*	SIZE OF αααααα NOT AN INTEGER MULTIPLE OF 60. Meaning: Phase table is not the right size to be made up of 60 word (30 Real word) position/velocity vectors. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use Interface Table Editor to enter the name of a valid phase table.

TABLE 4-VII.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR MPTP

Processor name	Processor abstract prompt (maximum 256 characters)
MPTP	The MPTP generates a mission plan table (MPT) display containing maneuvers calculated by various FDS maneuver planning processors and stored in a position/velocity phase table.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
GLOCON	Global constants array; master data base element normally defaulted to !!GLCN.
SESCON	Session constants array; normally defaulted to !SESCN.
LIST	Variable that identifies the logical unit of the output device onto which the mission plan table is written. The default value of zero is a flag that sets the user's terminal as the output device.
PVTAB	Phase table containing up to 20 position/velocity vectors that are used in generating the MPT. The phase table may be subscripted to indicate a starting vector.
NUMANV	Variable indicating the number of maneuvers to be displayed in the MPT.
OMSP	Variable specifying the initial OMS propellant load. Default value is zero.
RCSP	Variable specifying the initial RCS propellant load. Default value is zero.
SUSP	Variable specifying the initial SUS propellant load. Default value is zero.
OMSISP	Variable specifying the specific impulse for OMS in seconds.
RCSISP	Variable specifying the specific impulse for RCS in seconds.

TABLE 4-VII.- Concluded

PROCESSOR MPTP

Processor name	Processor abstract prompt (maximum 256 characters)

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
SUSISP	Variable specifying the specific impulse for SUS in seconds.
VEHID	Two-character vehicle identification code.

5.0 PROCESSOR ROUTINES

5.1 ROUTINE NAME - MAIN PROGRAM MPTP

5.1.1 Purpose

The main routine of the Mission Plan Table Processor (MPTP) controls all input, variable initialization error testing and error message display, flag setting, and calculations for the generation of the MPT display.

5.1.2 Functional Description

The main routine inputs data from an interface table and will calculate the ground elapsed time, delta velocity, delta time between maneuvers, delta velocity remaining, HA and HP, and will construct a code that identifies the vehicle, engine, and maneuver. These data are provided to a subroutine that generates the MPT display.

5.1.3 Assumptions and Limitations

a. MPTP will obtain and display information on a maximum of 10 maneuvers.

b. MPTP will obtain and display information only on the manuevers preceding the first 'DD' vector it encounters in the phase table.

c. MPTP will display only one vehicle identification code.

5.1.4 Method

a. Calculate the delta velocity remaining after a maneuver

Math symbol	Internal symbol	Definition
W_I	WHTI	Total vehicle weight after the maneuver
W_F	WHTF	W_I-fuel remaining for the engine used in the maneuver
ISP_{ID}	ISP ENGNID	Specific impulse for engine identified by ENGNID
ΔV_{REM}	DVREM	Delta velocity remaining
G_{SL}	GSL	Acceleration of gravity at sea level

$$\Delta V_{REM} = G_{SL} * ISP_{ID} * LOG\ (W_I/W_F)$$

5.1.5 Routine Input/Output Variables

A description of all input/output variables is given in table 5.1-I. This routine also generates the error messages identified under the processor messages topic.

5.1.6 Functional Logic Flow

A functional logic flow for MPTP is presented in figure 5.1-1 and a detailed flow is given in figure 5.1-2.

5.1.7 Diagnostics and Debug

None.

5.1.8 Special Comments

This routine utilizes the following subroutines, which are documented under the indicated section of the FDS-1 system design document.

GMTIM - Volume III
RARP - Volume III
SSVUC - Volume VII
XPATR - Volume VII
XPGET - Volume VII
XPGTI - Volume VII
XRMOV - Volume VII
XPXIT - Volume VII

5.1.9 References

None.

TABLE 5.1-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine MPTP

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
GLOCON		Real	I		IT	GLOCON	Global constants array
SESCON		Intg	I		IT	SESCON	Session constants array
PVTAB		Real	I		IT	PVTAB	Position/velocity phase table
OMSP		Real	I		IT	OMSP	Initial OMS propellant load
RCSP		Real	I		IT	RCSP	Initial RCS propellant load
SUSP		Real	I		IT	SUSP	Initial SUS propellant load
LIST		Intg	I		IT	LIST	Logical unit of output device
NUMANV		Intg	I		IT	NUMANV	Number of maneuvers to be displayed
OMSISP	I_{SP}	Real	I	sec	IT	OMSISP	Specific impulse for OMS
RCSISP	I_{SP}	Real	I	sec	IT	RCSISP	Specific impulse for RCS
SUSISP	I_{SP}	Real	I	sec	IT	SUSISP	Specific impulse for SUS
VEHID		2CH	I		IT	VEHID	Vehicle identification code

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.1-I.- Concluded

Routine MPTP

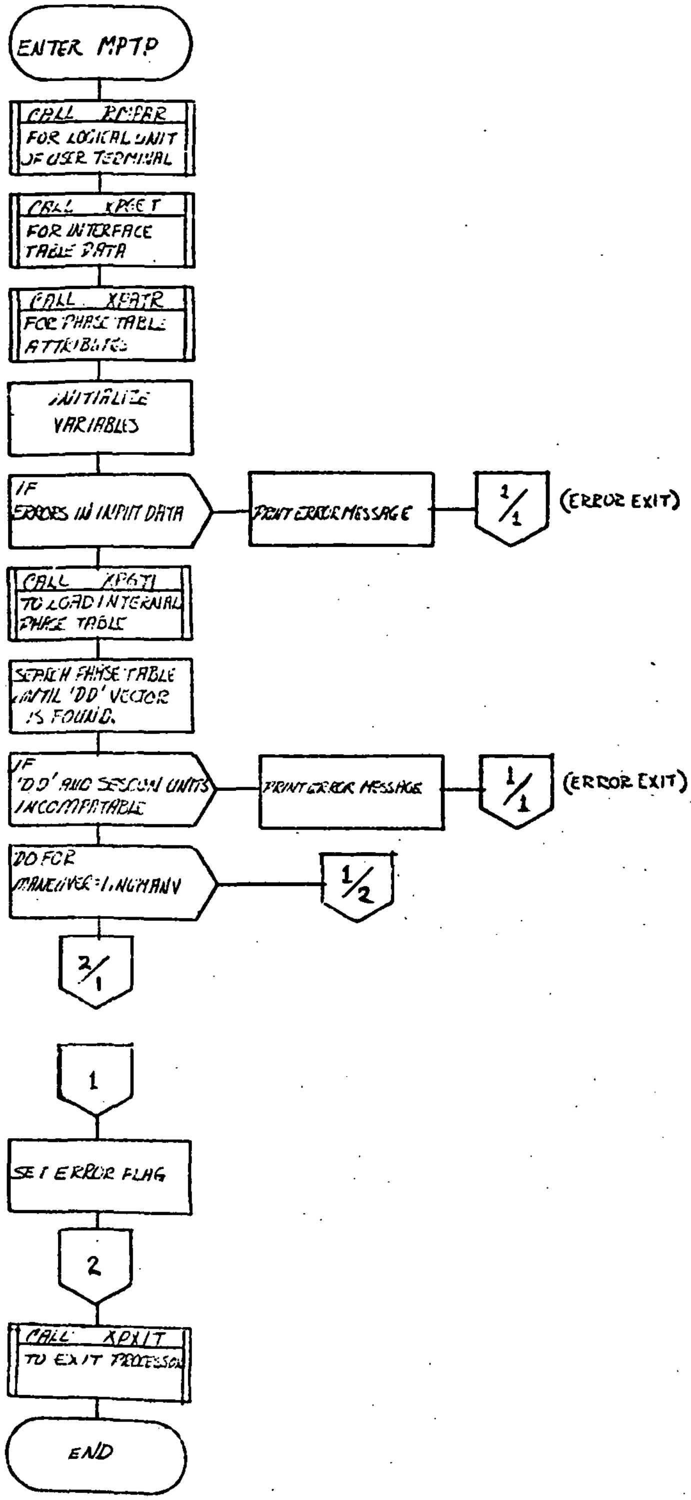
Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
IPRM(1)		Intg	I		SP	1ST PARAMETER	Logical unit of user terminal
IPRM(1)		Intg	O		SP	1ST PARAMETER	Error return flag

NOTES:

TYPE
Free Dubl 18CH Mix
Intg 2CH 36CH Char
Real 6CH 72CH Bin

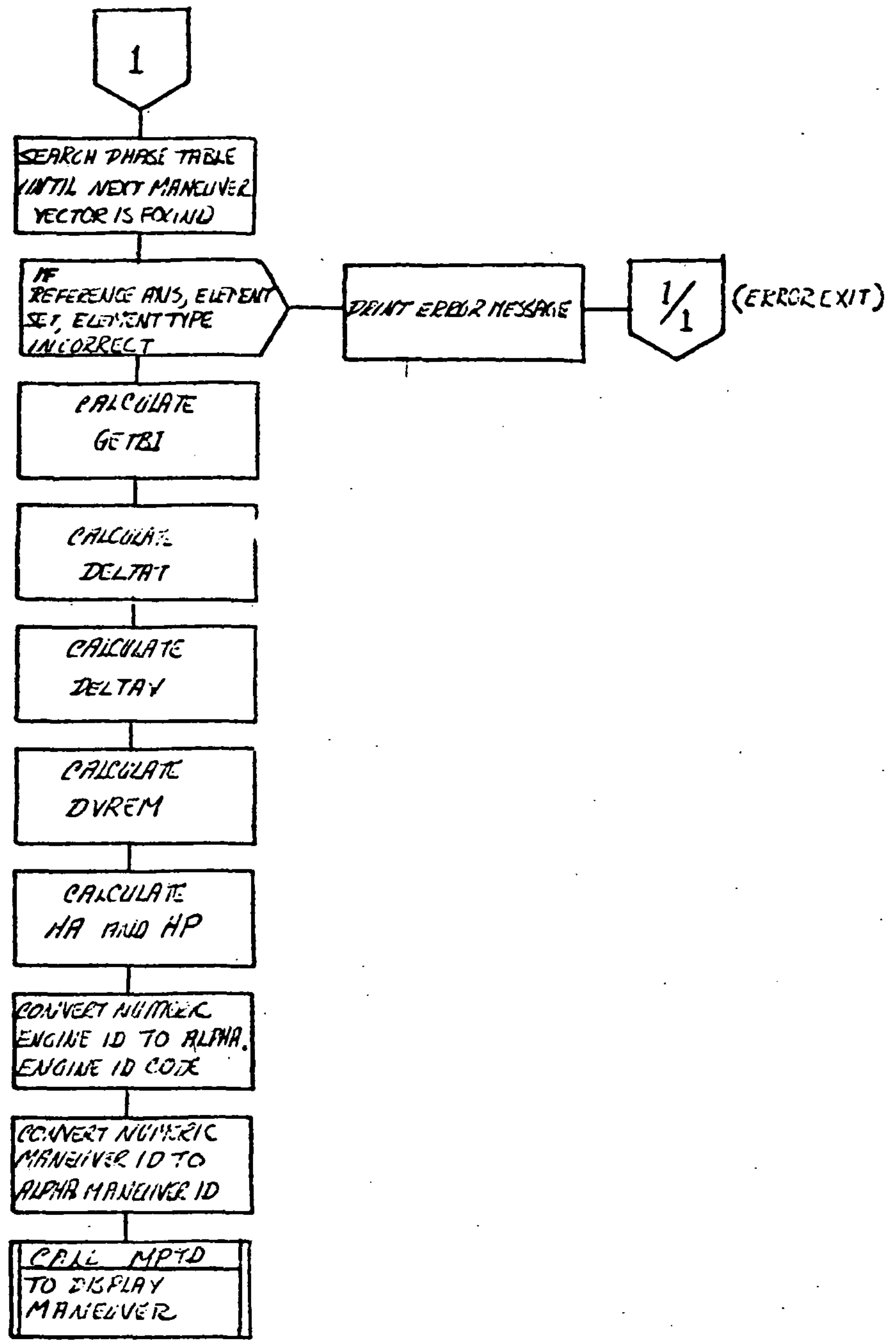
USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



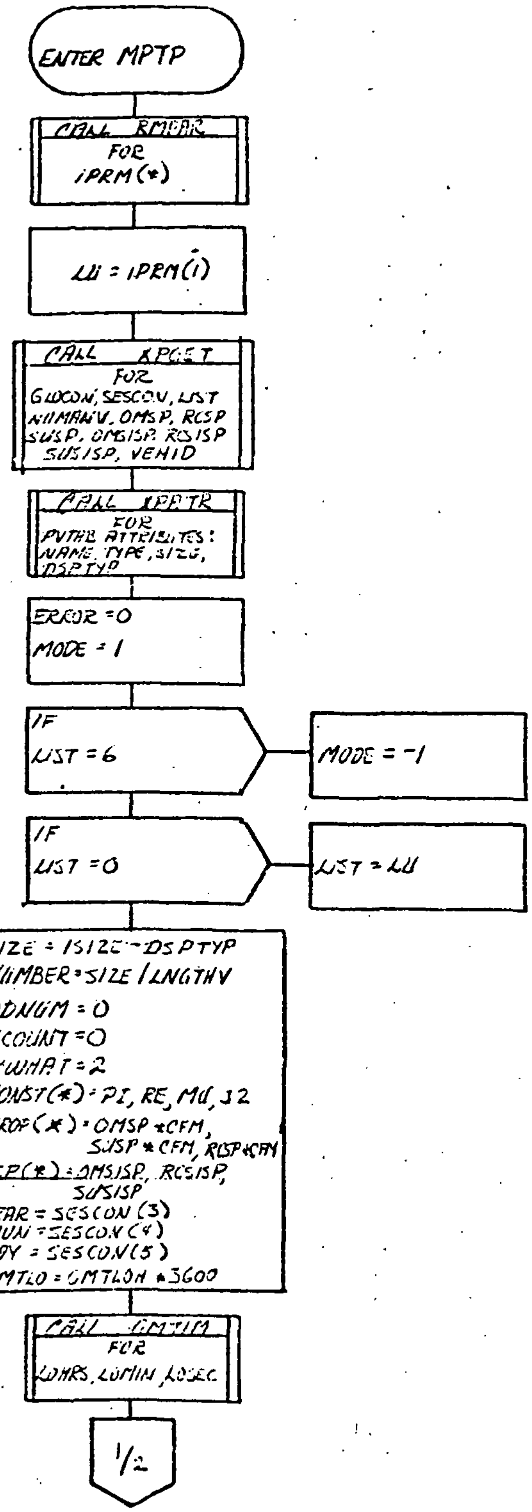
Page 1 of 2

Figure 5.1-1.- MPTP functional logic flow.



Page 2 of 2

Figure 5.1-1.- Concluded.



Page 1 of 5

Figure 5.1-2.- MPTP detailed logic flow.

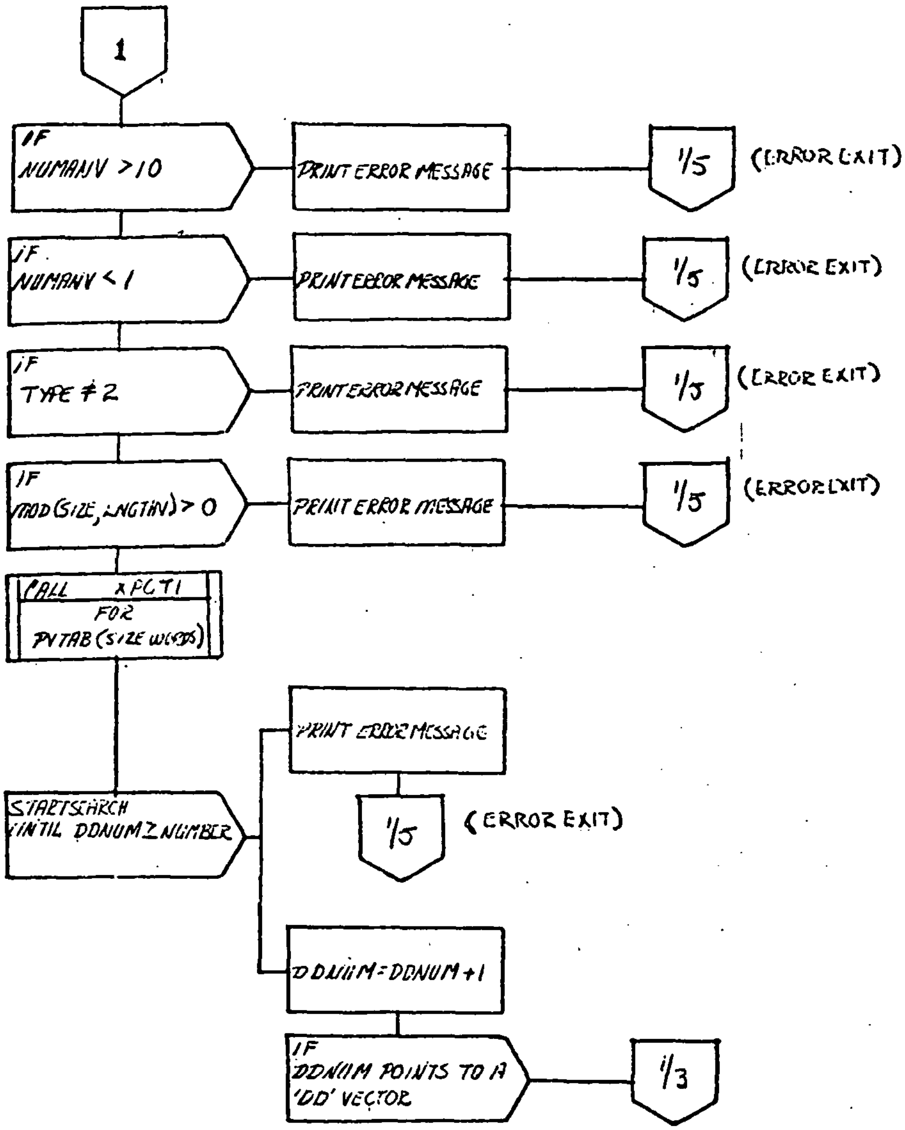
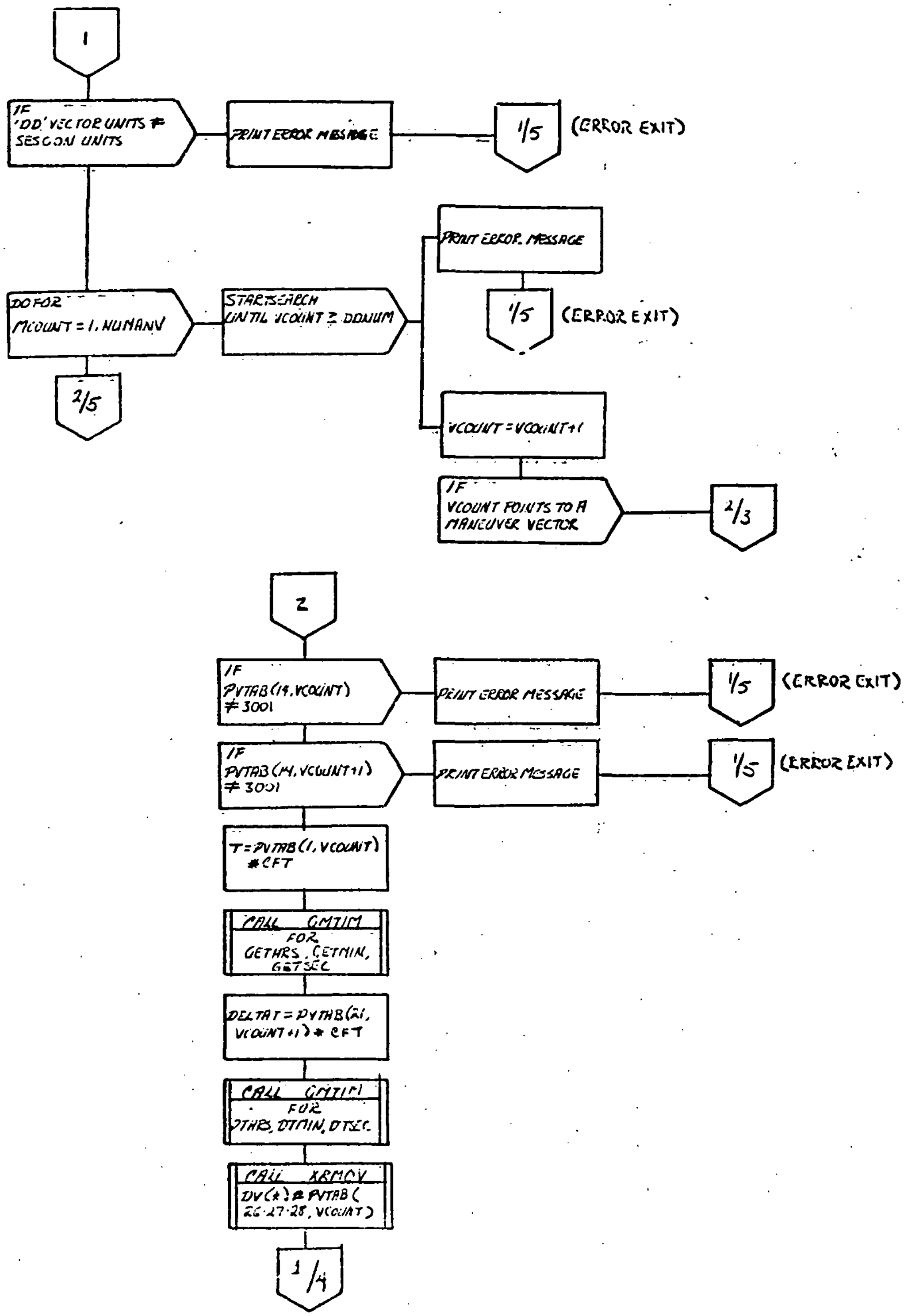
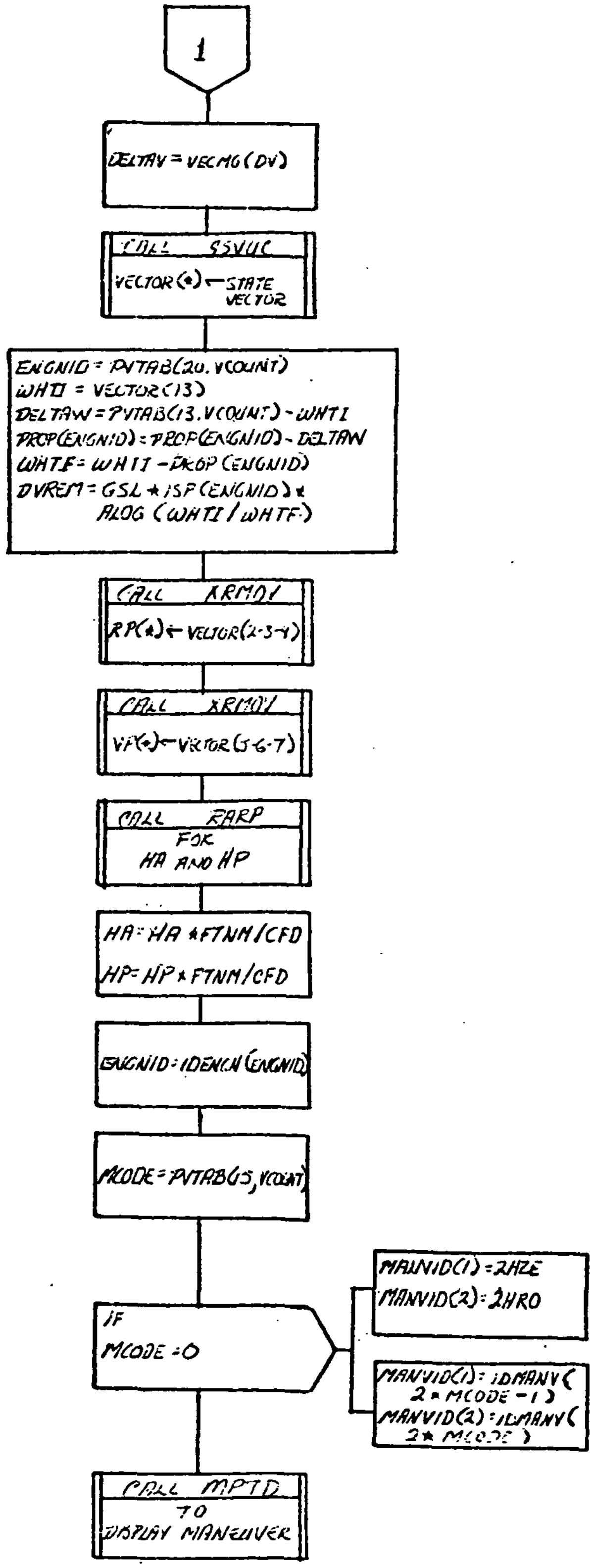


Figure 5.1-2.- Continued.



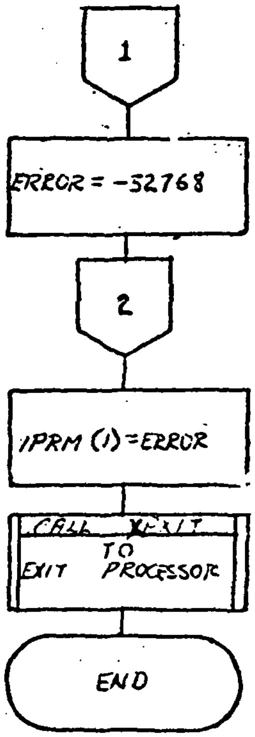
Page 3 of 5

Figure 5.1-2.- Continued.



Page 4 of 5

Figure 5.1-2.- Continued.



Page 5 of 5

Figure 5.1-2.- Concluded.

5.2 ROUTINE NAME - SUBROUTINE MPTD

5.2.1 Purpose

The Mission Plan Table Display (MPTD) routine displays information received from MPTP in a mission plan table.

5.2.2 Functional Description

The routine generates the display described under the processor display topic.

5.2.3 Assumptions and Limitations

None.

5.2.4 Method

The routine generates the display header and one maneuver on the first call, and one maneuver entry with each succeeding call. Thus, the complete display requires multiple calls to MPTD when more than one maneuver is to be displayed.

5.2.5 Routine Input/Output Variables

The MPTD input/output variables are presented in table 5.2-I.

5.2.6 Functional Logic Flow

No functional flow is provided. A detailed flow is provided in figure 5.2-1.

5.2.7 Diagnostics and Debug

None.

5.2.8 Special Comments

None.

5.2.9 References

None.

TABLE 5.2-I.- ROUTINE INPUT/OUTPUT VARIABLES

Routine MPTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
DAY		Intg	I/O		C		Days component of base date
DELTAV		Real	I/O		C		Delta-velocity magnitude
DTHRS		Intg	I/O		C		Hours component of delta time (DELTAT) between maneuvers
DTMIN		Intg	I/O		C		Minutes component of DELTAT
DTSEC		Real	I/O		C		Seconds component of DELTAT
DVREM		Real	I/O		C		Delta-V remaining
ENGNID		Intg	I/O		C		Engine identification code
GETHRS		Intg	I/O		C		Hours component of ground elapsed time of maneuver ignition
GETMIN		Intg	I/O		C		Minutes component of ground elapsed time of maneuver ignition
GETSEC		Real	I/O		C		Seconds component of ground elapsed time of maneuver ignition

NOTES:

TYPE: Free, Intg, Real, Dubl, 2CH, 6CH, 18CH, 36CH, 72CH, Mix, Char, Bin

USE:
I = Input
O = Output
I/O = Input/Output

SOURCE:
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory

TABLE 5.2-I.- Continued

Routine MPTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
HA		Real	I/O		C,T		Predicted height of apogee
HP		Real	I/O		C,T		Predicted height of perigee
LIST		Intg	I		C		Logical unit of output device
LOHRS		Intg	I/O		C,T		Hours component of lift-off time
LOMIN		Intg	I/O		C,T		Minutes component of lift-off time
LOSEC		Real	I/O		C,T		Seconds component of lift-off time
MANVID		Intg	I/O		C,T		Maneuver identification code
MCOUNT		Intg	I		C		Maneuver count number
MODC		Intg	I		C		Print mode
MON		Intg	I		C		Base date month number
NAME		Intg	I/O		C,T		Name of phase table
NUMANV		Intg	I		C		Number of maneuvers to be displayed
NOTES:		TYPE Free Dubl 18CH Mix Intg 2CH 36CH Char Real 6CH 72CH Bin				USE I = Input O = Output I/O = Input/Output	SOURCE IT = Interface Table T = Terminal A = Calling Argument C = Common F = Disk File SAM = System Available Memory

TABLE 5.2-I.- Concluded

Routine MPTD

Code symbol	Math symbol	Type	Use	Units	Source	External label	Definition
VEHID		Intg	I/O		C,T		Vehicle identification code
YEAR		Intg	I/O		C,T		Base date year
MONTH		Intg	O		T		Base date month

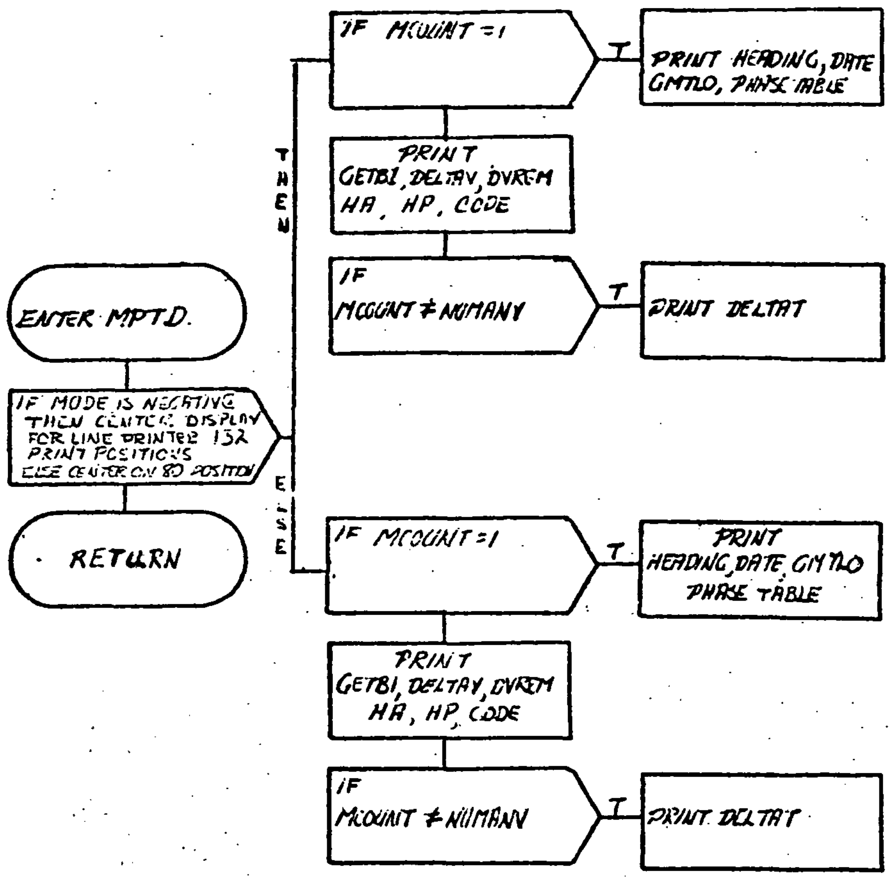
NOTES:

TYPE

Free	Dubl	18CH	Mix
Intg	2CH	36CH	Char
Real	6CH	72CH	Bin

USE
I = Input
O = Output
I/O = Input/Output

SOURCE
IT = Interface Table
T = Terminal
A = Calling Argument
C = Common
F = Disk File
SAM = System Available Memory



Page 1 of 1

Figure 5.2-1.- MPTD detailed logic flow.

NODE DEFINER PROCESSOR (NODE)

TO BE SUPPLIED

ORBITAL MANEUVER PROCESSOR (OMP)

1.0 PURPOSE

The Orbital Maneuver Processor (OMP) provides the capability to compute targeting conditions and auxiliary planning data for a sequence of orbital maneuvers which will result in a rendezvous between an active chasing vehicle and another orbiting target vehicle or payload. The processor accepts as input a sequence of standard maneuver types and associated targeting parameters and constraints. The processor then computes and outputs resulting maneuver and trajectory data for a complete rendezvous sequence or part of a rendezvous sequence as specified by the input.

2.0 FUNCTIONAL DESCRIPTION

OMP has the capability to compute the following standard rendezvous maneuver types:

a. NPC - Plane change maneuver executed at an intersection point of the target and primary vehicles' orbital planes and targeted to make the primary vehicle coplanar with the target vehicle.

b. NS - Common node shift maneuver. A maneuver targeted to place the common node approximately 90° from the maneuver point.

c. NC - Horizontally executed maneuver targeted to obtain desired phasing conditions at a subsequent maneuver point.

d. NH - Horizontally executed maneuver targeted to obtain desired differential altitude between target orbit and primary orbit at a subsequent maneuver point.

e. NCC - A Lambert targeted maneuver used to correct the rendezvous trajectory to achieve desired phasing, desired differential altitude, and the creation of a common node at a subsequent NSR maneuver.

f. NSR - A maneuver targeted to put the primary vehicle in an orbit coelliptic to the orbit of the target vehicle. The coelliptic condition is defined as a condition wherein there are coincident lines of apsides and equal differential altitudes at apogee and perigee. This maneuver also nulls the out-of-plane relative velocity automatically when it is immediately preceded by an NCC maneuver.

g. TPI - Terminal-phase initiation. A Lambert targeted maneuver that puts the primary vehicle on an intercept trajectory with the target vehicle.

h. TPM - Terminal-phase midcourse maneuver. A Lambert targeted maneuver that computes post-TPI corrections to the intercept trajectory.

i. TPF - Terminal-phase finalization; the second maneuver in the Lambert two-impulse TPI/TPF solution to the intercept problem. This maneuver is targeted but never executed. During the actual mission, terminal braking is accomplished by the execution of a set of several manually controlled braking maneuvers.

j. SOI - Stable orbit initiation. A Lambert targeted maneuver that is targeted to cause the chaser vehicle to intersect the orbit of the target vehicle at a given downrange relative position from the target vehicle.

k. SOM - Stable orbit midcourse. A Lambert targeted maneuver that makes corrections to the post-SOI trajectory.

l. SOR - Stable orbit rendezvous. This maneuver is targeted to put the chaser vehicle into the same orbit as that of the target vehicle. The maneuver occurs on arrival at the orbit intersection point targeted to by the SOI maneuver.

OMP also has the capability to compute the following standard orbit shaping maneuvers:

m. EXDV LV - External $\overline{\Delta V}$ local vertical. This maneuver is specified by input as a ΔV vector in "local vertical local horizontal" (LVLH) coordinates.

n. EXDV LOS - External $\overline{\Delta V}$ line of sight. This maneuver is specified by input as a ΔV vector in "line-of-sight" coordinates. Where X is along the line of sight, Y is perpendicular to line of sight and positive is the direction of negative angular momentum, Z completes the right handed system.

o. CIRC - Orbital circularization maneuver.

p. HA - Height adjust maneuver. A horizontally executed maneuver targeted to achieve a desired orbital altitude 180° from the maneuver point.

q. HASH - A maneuver targeted to effect a desired altitude 180° from the maneuver point and simultaneously move the apsidal line to the maneuver point.

r. PC - Plane change maneuver. Not constrained to be relative to a target vehicles' plane.

s. NOSH - Node shift maneuver. Moves orbit line of equatorial nodes through specified angle. Maneuver may be constrained to preserve orbital inclination while shifting the node.

t. APSH - Apsis shift maneuver. Moves line of apses through a specified angle. Maneuver is constrained to preserve the premaneuver values of apogee and perigee radii.

OMP has the capability to null the relative out-of-plane velocity. This option is available for each maneuver to be targeted except for maneuvers (m), (n), (r), and (s) listed in this section.

OMP has the capability to achieve desired relative geometric conditions at each maneuver point. The relative (chaser-to-target) conditions desired at each maneuver point may be chosen from the following options:

a. No relative geometric constraints.

b. Any one or two of the following:

   (1) Differential altitude desired at the maneuver point.

   (2) Phase angle desired at the maneuver point.

   (3) Elevation angle desired at the maneuver point.

c. Specification of one of the following special constraints on NCC:

   (1) Phase angle at NCC is to be determined such that the NCC maneuver has a specified radial component.

   (2) Phase angle at NCC is to be determined such that the NCC maneuver has a specified maneuver pitch angle in the LVLH coordinate system.

   (3) Phase angle at NCC is to be determined such that the subsequent NSR maneuver has a specified maneuver pitch angle in the LVLH coordinate system.

   (4) Phase angle at NCC is to be determined such that the total delta-V for the NCC/NSR combination will be as close as possible to a specified value. This option can be used to minimize the total delta-V required for these two maneuvers.

d. Flag indicating whether the coelliptic maneuver NSR is to occur at the point of minimum relative radial rate. This option automatically causes an iteration on the time of the first maneuver in the sequence with the intent to end up placing NSR at the optimum position to minimize the radial component of the coelliptic maneuver.

e. Curvilinear downrange distance at SOR. Specified only for the SOR maneuver point.

OMP has the capability of achieving other special targeting requirements. They are listed below for the indicated orbit shaping maneuver.

a. Delta-V vector for the external ΔV maneuvers EXDV_LV and EXDV_LOS.

b. Orbital altitude to be achieved by the height adjust maneuvers HA and HASH.

c. Angular value of node shift to be achieved by the equatorial node shift maneuver NOSH.

d. Flag indicating whether the node shift maneuver NOSH is constrained to maintain the orbital inclination or is free to change it.

e. Wedge angle specifying the plane change maneuver PC.

f. Angular value of apsis shift to be achieved by the apsis shift maneuver APSH.

OMP has a variety of maneuver time/location options for when and/or where each of the standard maneuvers listed are to occur. Also, some of the options require the specification or computation of a threshold time. These options are:

Maneuver Time/Location Options

a. Specified time.

b. Delta-T from last maneuver.

c. Central travel angle from last maneuver.

d. M orbital periods (including fractional portion) from last maneuver.

e. Nth apsis crossing from last maneuver.

f. Nth apsis crossing after the threshold time.

g. Nth apogee arrival after the threshold time.

h. Nth perigee arrival after the threshold time.

i. First arrival at a specified orbital altitude (spherical Earth) after the threshold.

j. First arrival at a specified orbital radius after the threshold time.

k. First arrival at a specified Earth longitude after the threshold time.

l. First arrival at a specified Earth geocentric latitude after the threshold time.

m. First arrival at a specified angle from perigee after the threshold time.

n. First arrival at a point in the orbit where, if a node shift was performed, it would not change orbital inclination after the threshold time.

o. First arrival at a specified angle from apogee after the threshold time.

p. First arrival at a specified argument of latitude after the threshold time.

q. Delta-T from first orbital sunrise or sunset after the threshold time.

r. First arrival at a specified elevation angle to the other vehicle after the threshold time.

s. First arrival at the common node of two orbits after the threshold time.

Maneuver Time Threshold Options - The following is a list of the options available for specification of threshold time for any of the above maneuver time options requiring a threshold time:

a. Threshold time is specified directly by input.

b. Threshold time is specified to be an input delta-T from the last maneuver.

c. Threshold time is specified by an input central travel angle from the last maneuver.

d. Threshold time is specified by an input number of orbital periods from the last maneuver.

e. Threshold time is specified as arrival at the Nth apsidal crossing after the last maneuver.

OMP has the capability of accepting a first guess for the magnitude of the maneuver delta-V for each input. It is not used for all maneuver types. It is specifically needed to start the phasing (NC) iteration to either increase or decrease the catch-up rate of the active vehicle.

OMP has the capability of using either a conic or analytic perturbations and atmospheric drag effects when the propagation/prediction of state vectors is required.

The Orbital Maneuver Processor is capable of solving a sequence of very specialized linked two-point boundary value problems. That is, it computes the maneuver parameters for a sequence of standardized maneuver types such that several intermediate targeting conditions are satisfied, while at the same time satisfying the desired final rendezvous end condition. The maneuver targeting algorithms of this processor are characterized as involving the following types of computations: Orbital state vector propagation is performed with either two-body theory or analytic gravity perturbation theory with atmospheric drag; maneuvers are modeled impulsively. Boundary value problems are solved numerically with Newton-Raphson iteration. Several standard maneuver types are targeted with a conic Lambert rendezvous targeting algorithm using a universal variable formulation. If precision propagation is selected by the user, the Lambert targeted maneuvers are computed using the Lambert off-set targeting technique.

The general processing flow of this processor cycles through the input list of standard maneuver types to be targeted and performs the following:

a. Uses a state propagator to compute the time and vehicle states at the specified maneuver point.

b. Computes the delta-V vector ($\overline{\Delta V}$) required to satisfy the targeting specifications on the maneuver. The maneuver ΔV computation algorithm depends upon the type of maneuver being computed. Some maneuvers require processing to establish internal targeting conditions. Some maneuvers have explicit analytic solutions while others require numerical iteration.

c. The maneuver $\overline{\Delta V}$ is impulsively applied to the chaser's initial velocity vector at the maneuver point.

This process is sequentially carried out for all maneuvers in the input sequence. The controlling logic of the processor will loop back to an earlier point in the sequence if necessary to iterate on a maneuver that has not converged on its subsequent target. For example, the following is a brief description of the standard Shuttle "double coelliptic" rendezvous sequence and the processing required to compute the maneuver parameters for the sequence. The sequence generally will consist of the seven maneuvers described as follows; however, if the target orbit is highly elliptical, a circularization maneuver will be performed at the first apogee and then the standard sequence will begin along the line of apsides of the target orbit. The maneuvers of the double coelliptic sequence are all standard maneuver types selected from the standard maneuver list and are listed in chronological order:

a. NC - Phasing maneuver; controls desired phase angle at NCC.

b. NH - Height maneuver, executed some specified multiple of half revolutions after NC, controls desired differential altitude at NCC.

c. NSR_1 - First coelliptic maneuver, executed one-half revolution after NH.

d. NCC - Corrective combination maneuver, executed from one-half to one-and-a-half revolutions after NSR_1 (dependent on lighting conditions).

e. NSR_2 - Second coelliptic maneuver, executed within the next revolution following NCC, its time chosen by premission analysis to optimize the effectiveness of NCC.

f. TPI - Terminal phase initiation, executed within the revolution following NSR_2.

g. TPF - Terminal phase finalization, velocity match at intercept, actually executed as a series of braking maneuvers.

The sequence was developed to provide a standard, general-purpose, rendezvous profile that is capable of controlling both lighting and relative range from the half revolution preceding NCC through to TPF. This sequence allows the mission planners to fix, for all practical purposes, the relative geometries, the lighting conditions, the maneuver spacing, and the maneuver characteristics. This situation is extremely desirable because not only is a great degree of standardization introduced into planning and training but, more importantly, the standard NCC through TPF relative geometries may be set up so that highly critical optical tracking can occur within the half revolution prior to NCC - that is, relative range and lighting constraints on navigation can be handled in a standard fashion. The sequence can be used to satisfy those constraints independent of lift-off time (i.e., insertion phase angle) and target altitude. The Orbital Maneuver Processor can accomplish the above maneuver sequence definition along with its appropriate maneuver time specifications and maneuver constraint options; when such a sequence is specified the processor carries out the following processing:

a. Initializes iteration parameters such as initial estimates of delta-V required for NC and NH maneuvers.

b. Propagates both input states to the time of the NC maneuver and implusively applies the current estimate of the required NC ΔV.

c. Propagates both states to the time of the NH maneuver and impulsively applies the current estimate of the required NH ΔV.

d. Propagates both states to the time of the NSR maneuver and computes and impulsively applies the coelliptic ΔV required.

e. Propagates both states to the time of the NCC maneuver. If the phase angle obtained is not equal to the desired phase angle (within tolerance) the processor computes a new estimate of the required NC ΔV and cycles back to step b. If the differential altitude obtained is not equal to the desired differential altitude (within tolerance) the processor computes a new estimate of the required NH ΔV and cycles back to step c. If both are within tolerance, processing moves on to step f.

f. The desired chaser NSR_2 position vector is then computed in order to establish a target vector for NCC maneuver targeting. The process of generating that target vector involves first computing the desired chaser TPI state vector and propagating that state vector backward in time to the NSR_2 time. The resulting NSR_2 state vector becomes the target for the NCC maneuver. The processing required to generate the desired NSR_2 position vector is as follows: propagate the target vehicle's state to the time of TPI, off-set from that state by the desired differential altitude and phase angle to establish the desired position vector, compute the coelliptic velocity vector desired on arrival at TPI, and then propagate this position and velocity vector back to the NSR_2 time.

g. Computes the NCC $\overline{\Delta V}$ required to arrive at the desired NSR_2 time with the NSR_2 position vector. Applies the ΔV impulsively at NCC time.

h. Propagates both states to the time of the NSR_2 maneuver and computes and impulsively applies the coelliptic ΔV required.

i. Propagates both states to the time of the TPI maneuver and computes and impulsively applies the ΔV required at TPI to intercept the target vehicle at the desired TPF time.

j. Propagates both states to the time of TPF and computes the TPF velocity match ΔV.

The above processing description is typical of this processor; however, many different maneuver sequences may be targeted other than the standard Shuttle double coelliptic sequence.

3.0 ASSUMPTIONS AND LIMITATIONS

The assumptions and/or limitations of OMP are:

a. All maneuver times will be equal to or greater than the latest state vector time, either chaser or target.

b. The number of maneuvers in a maneuver sequence is 9 or less.

c. All maneuvers are impulsive maneuvers.

d. The vehicle that performs the initial maneuver of a maneuver sequence will perform all the maneuvers of that sequence. The maneuver sequences are NC, NH, NSR; NCC, NSR; TPI, TPM, TPF; and SOI, SOM, SOR.

4.0 PROCESSOR INPUT/OUTPUT

a. Processor interface table - The definition of the OMP interface table parameters is provided in table 4-I. GLOCON is a set of constants universal to all the processors that will be maintained in the master data base. OMP will access this array for the constants it requires. The default values are stored in !!GLCN.

OMP will access the SESCON array to obtain the session related constants generated by the user upon execution of the system utility processor BASTM. The default values are stored in !SESCN and are standard for all processors.

Specific constants and tolerances required by OMP will be maintained through default values for the parameter RMPCON. These parameters are used mainly for iteration tolerances and internal print analysis.

CHASER and TARGET are the input vehicle state vectors in the standard format which includes position and velocity vectors, times, vehicle weights, drag quantities (C_D and Area) and the predictor or propagator to be used. These states are expressed in the standard TEG reference axis and Cartesian element set.

SVPROP selects the propagation/prediction models used to update a state vector from one point to another. The options are conic or two-body motion, Analytic Ephemeris Generator (AEG) with the gravitational effects added and atmospheric drag, and an AEG with a simplified gravity model.

The following interface table parameter names are used to specify the desired maneuver sequence:

MANJ where J = 1,9	Maneuver and vehicle identification code to describe the maneuver option and the vehicle that will perform the maneuver (chaser or target)
TMOPTJ where J = 1,9	Maneuver time/threshold time ID code or location ID code and value
CONVLJ where J = 1,9	The ID code and value of the maneuver constraints to be satisfied either by this or at this maneuver
DELTVJ where J = 1,9	The first guess of the maneuver delta velocity magnitude
ENGINJ where J = 1,9	The engine and thrust profile to be used in the computation of the maneuver burn times and in the vehicle weight loss due to the maneuver

The capability also exists to input five different thrust profiles with the parameter for (TFILEJ, J = 1,5) where each profile consists of a constant thrust and specific impulse.

The following interface table names are the output of OMP:

(a) APVTAB - The Active or Chaser vehicle state vector phase table. The initial input state vector and the preburn and postburn vectors will be output to the table in the TEG coordinate system and the Cartesian element set.

(b) PPVTAB - The passive or target vehicle state vector phase table.

The two phase tables are identical in size and contain the same kind of information except the vehicle not performing the maneuver would have its maneuver point state vector stored in both the preburn and postburn locations in the table.

The units associated with the parameters in the interface tables are in the user selected external units.

b. Interface table data array definitions - The definition of the input/output data arrays appearing in the OMP interface table is provided in table 4-II.

c. Interface table data file definitions - None.

d. Processor solicited (prompted) inputs - None.

e. Processor displays and display parameter definition table - One display is generated by OMP that varies in size due to the number of maneuvers in a maneuver sequence. The format of this display is shown in table 4-III and a definition of the display variables is provided in table 4-IV.

All the display parameters are in the user selected external units with the following exceptions. The maneuver times will be in days, hours, minutes, and seconds. The maneuver burn time will be in seconds. All angles will be in degrees. The apogee and perigee altitudes will be either NM or KM, based on the user's choice being either English or metric units.

f. Processor message table - When OMP converts the user input maneuver sequence definition to internal flags required to solve the requested problem, testing is performed to ensure that there is compatibility between the maneuver types and maneuver constraints. If compatibility does not exist, a message will be printed, no maneuver targeting will occur, and an error exit is taken to return to the FDS Executive.

   The messages generated by OMP are listed in table 4-V.

g. Interface table extended prompts - The processor extended prompts for each interface table parameter keyword are provided in table 4-VI.

TABLE 4-I.- PROCESSOR INTERFACE TABLE

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
GLOCON	AWA	Free	I	180	180	!!GLCN	Global constants array master data base element
SESCON	AWA	Free	I	90	90	!SESCN	Session constants array
SCOEF	AWA	Real	I	44	22	!SCOEF	Solar coefficients array
SMCN	AWA	Real	I	80	40	!!SMCN	Sun and Moon constants array
RMPCON	AWA	Free	I	100	50		OMP constants/tolerance ARRAY
CHASER	AWA	Real	I	30	15		Chaser state vector
TARGET	AWA	Real	I	30	15		Target state vector
SVPROP	AWA	Symb	I	5	1		Propagation option to be used for targeting = 'CONIC'; two body = 'AEGS'; small analytic ephemeris generator = 'AEGL'; large analytic ephemeris generator
TFILE1	AWA	Real	I	4	2,1		Thrust profile number 1, containing the thrust and specific impulse (ISP) in seconds

NOTES	CLASS	TYPE		USE
	AWA	Free 2CH	72CH	I = Input
	Disk	Intg 6CH	Mix	O = Output
		Real 18CH	Symb	I/O = Input/Output
		Dubl 36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TFILE2	AWA	Real	I	4	2,1		Thrust profile number 2
TFILE3	AWA	Real	I	4	2,1		Thrust profile number 3
TFILE4	AWA	Real	I	4	2,1		Thrust profile number 4
TFILE5	AWA	Real	I	4	2,1		Thrust profile number 5
MAN1	AWA	Symb	I	10	1		Maneuver and vehicle ID codes in the form of 'A,B' where B is vehicle ID: CHASER or TARGET and A is the maneuver ID from the following list: NC - Phasing maneuver NH - Height maneuver NCC - Corrective combination maneuver NSR - Coelliptic maneuver TPI - Initial terminal phase maneuver TPM - Terminal phase midcourse maneuver TPF - Final terminal phase maneuver

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
MAN1							NS - Common node shift maneuver NPC - Common node plane change maneuver SOI - Initial stable orbit maneuver SOM - Stable orbit midcourse maneuver SOR - Final stable orbit maneuver EXDV - Input external delta-V maneuver in LVLH system CIRC - Circularization maneuver LSDV - Input external delta-V maneuver in line-of-sight system PC - Single vehicle plane change maneuver HA - Height maneuver to an altitude 180° from maneuver point HASH - Height maneuver to an altitude 180° from maneuver point plus shift apsides to maneuver point NOSH - Node shift maneuver APSH - Apsides shift maneuver END - End of maneuver sequence if the sequence has less than nine maneuvers

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TMOPT1	AWA	Symb	I	50	1		Maneuver time/threshold time ID code and value plus maneuver location ID code and value. The form of the input is 'A = X' or 'A = X, B = X' X is either a real or integer value or value list A is one of the following for maneuver/threshold ID: T - Input time DT - Delta-T from last maneuver M - Number of orbital periods (including fractional portion) from last maneuver WT - Central travel angle from last maneuver N - Number of apsides crossing from last maneuver

NOTES	CLASS	TYPE			USE
	AWA Disk	Free Intg Real Dubl	2CH 6CH 18CH 36CH	72CH Mix Symb	I = Input O = Output I/O = Input/Output

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TMOPT1							B is one of the following for maneuver location ID: EL - Desired elevation angle N - Maneuver as at the Nth apsis crossing ALT - Maneuver is at an altitude LAT - Maneuver is at a latitude LON - Maneuver is at a longitude NA - Maneuver is at Nth apogee crossing NP - Maneuver is at Nth perigee crossing CN - Maneuver is at a common node R - Maneuver is at a radius P - Maneuver is at an angle from perigee OPT - Maneuver is at the optimum node shift point A - Maneuver is at an angle from apogee U - Maneuver is at an argument of latitude

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
TMOPT1							NITI - Maneuver is near the entrance to darkness NITM - Maneuver is near the midpoint of darkness (midnight) NITO - Maneuver is near the exit of darkness LITI - Maneuver is near the entrance to daylight LITM - Maneuver is near the midpoint of daylight (noon) LITO - Maneuver is near the exit of daylight
CONVL1	AWA	Symb	I	60			The ID code and value of the maneuver constraint(s) to be satisfied either by or at this maneuver is of the following forms 'A=X, A=X, A=X, A=X' or 'A=X, A=X, A=X' or 'A=X, A=X' or 'A=X'

NOTES	CLASS	TYPE			USE
		Free	2CH	72CH	I = Input
	AWA	Intg	6CH	Mix	O = Output
	Disk	Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
CONVL1							X is either a real or integer value or value list. A is one of the following: DH - Desired delta altitude at maneuver point PHA - Desired phase angle at maneuver point EL - Desired elevation angle at maneuver point DR - Desired trailing displacement for a stable orbit HD - Altitude desired 180° from maneuver point DVLV - Input LVLH delta-V vector DVLS - Input line-of-sight delta-V vector DNOD - Desired node shift by this maneuver DPC - Desired plane shift by this maneuver RDOT - Iterate to achieve a minimum or desired radial velocity at either NCC or NSR maneuver points

NOTES	CLASS	TYPE			USE
	AWA Disk	Free Intg Real Dubl	2CH 6CH 18CH 36CH	72CH Mix Symb	I = Input O = Output I/O = Input/Output

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
CONVL1							DVCR - Iterate to achieve a minimum or desired total delta-V magnitude for the NCC/NSR combination CPIT - Iterate to achieve a desired maneuver pitch angle of the NCC maneuver SPIT - Iterate to achieve a desired maneuver pitch angle of the NSR maneuver in the NCC/NSR combination
DELTV1	AWA	Real	I	2	1		First guess of the delta-V magnitude of either or both NC or NH maneuvers
ENGIN1	AWA	Symb	I	10			Engine and thrust profile to be used for this maneuver of the form 'A, X' A is one of the following: OMS - OMS system RCS A - RCS-aft RCS F - RCS-fwd

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Continued

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
ENGIN1							X is an integer value. The value of X selects one of the user-defined thrust profiles in the parameters TFILE1, TFILE2, TFILE3, TFILE4, and TFILE5. Example: = 1; selects TFILE1 = 2; selects TFILE2 etc.
MAN2 . . . MAN9	AWA . . . AWA	Symb . . . Symb	I . . . I	10 . . . 10	1 . . . 1		Same as MAN1 except for maneuvers 2 through 9
TMOPT2 . . . TOMPT9	AWA . . . AWA	Symb . . . Symb	I . . . I	50 . . . 50			Same as TOMPT1 except for maneuvers 2 through 9
CONVL2 . . . CONVL9	AWA . . . AWA	Symb . . . Symb	I . . . I	60 . . . 60			Same as CONVL1 except for maneuvers 2 through 9

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-I.- Concluded

PROCESSOR OMP

Parameter keyword name	Class	Type	Use	Size	Array dimension (I,J)	Values stored in default interface table	Definition
DELTV2	AWA	Real	I	2	1		Same as DELTV1 except for maneuvers 2 through 9
.	.	.	.	.	.		
.	.	.	.	.	.		
.	.	.	.	.	.		
DELTV9	AWA	Real	I	2	1		
ENGIN2	AWA	Symb	I	10	1		Same as ENGIN1 except for maneuvers 2 through 9
.	.	.	.	.	.		
.	.	.	.	.	.		
.	.	.	.	.	.		
ENGIN9	AWA	Symb	I	10	1		
APUTAB	AWA	Real	O	1140	30,19		Active or chaser vehicle phase table
PPUTAB	AWA	Real	O	1140	30,19		Passive or target vehicle phase table

NOTES	CLASS	TYPE			USE
	AWA	Free	2CH	72CH	I = Input
	Disk	Intg	6CH	Mix	O = Output
		Real	18CH	Symb	I/O = Input/Output
		Dubl	36CH		

TABLE 4-II.- INTERFACE TABLE DATA ARRAY DEFINITIONS

PROCESSOR OMP

Array name	Index location	Default value	Definition
GLOCON	(1) . . . (180)	!!GLCN	Global constants array, master data base elements; see table 7.2-III of JSC IN 78-FM-60, vol. I for definition of contents
SESCON	(1) . . . (90)	!SESCN	Session constants array; see table 7.2-II(a) of JSC IN 78-FM-60, vol. I for definition of contents
RMPCON	(1)	0	Output unit on which to write all print. The default value is a flag to use the user terminal; a positive value identifies the output unit
	(2)	0	Internal/debug print flag option
			= 0; no internal print
			= 1; internal print
	(3)	0	Output unit for internal print; the default value is a flag to use the users terminal
	(4)	20	Maximum iteration limit
	(5)	2.0E-04	Angular tolerance, radians
	(7)	250.0	Distance or radius tolerance, feet
	(9)	6.0E-04	Eccentricity tolerance
	(11)	2.0	First guess change in value of independent variable; used in subroutine ITERV
	(13)	50.0	Tolerance used when elevation angle and differential altitude is not compatible
	(15)	0.1	Sine of the angle used to determine if the Lambert problem is near a 180-degree transfer
	(17)	1000.0	Maximum allowable delta-V
	(19)	.025	Percentage of maximum delta-V to be used as a first guess at the phasing maneuver solution

TABLE 4-II.- Concluded

PROCESSOR OMP

Array name	Index location	Default value	Definition
	(21)	900.0	Maximum step in elevation angle search option
SCOEF	(1) . . . (90)	!SCOEF	Solar coefficients constants array; see table 7.2-II(b) of JSC IN 78-FM-60, vol. I for definitions of contents
SMCN	(1) . . . (90)	!!SMCN	Sun and Moon constants array; see table 7.2-IV of JSC IN 78-FM-60, vol. I definitions of contents
APVTAB			Chaser vehicle output phase table
	(1,1)		State vector at the initial time
	(1,2)		State vector at 1st maneuver - preburn; impulsive
	(1,3)		State vector at 1st maneuver - postburn; impulsive
	(1,4)		State vector at 2nd maneuver - preburn
	(1,5)		State vector at 2nd maneuver - postburn
	. . .		. . .
	(1,19)		State vector at 9th maneuver - postburn
PPVTAB			Target vehicle output phase table
	(1,1)		State vector at the initial time
	(1,2)		State vector at 1st maneuver - preburn
	(1,3)		State vector at 1st maneuver - postburn
	. . .		. . .
	(1,19)		State vector at 9th maneuver - postburn
			The preburn and postburn vectors for the vehicle not performing the maneuver will be identical

TABLE 4-III.- PROCESSOR DISPLAY TABLE

PROCESSOR OMP

```
                            RENDEZVOUS  TARGETING  MANEUVER  TABLE
        BASE  DATE  XXXX  XX  XX                           GMTLO  XXX  XX.XX  XX.XXX

  MAN             GMTI          DVLV-X      VG-X      HA        DH        YP          DV
  VEH             GETI          DVLV-Y      VG-Y      HP        PH        YPDOT     PITCH
T  PROF         GMTTIG          DVLV-Z      VG-Z                DT        TBURN       YAW
                 TIME
X  XXXX  XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX
         XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX
      X  XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX             ±XXX.XX   ±XXX.XX   ±XXX.XX
              XX  XX  XX.X     UNTIL
X  XXXX  XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX
         XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX   ±XXX.XX
      X  XXX  XX  XX  XX.X    ±XXX.XX    ±XXX.XX             ±XXX.XX   ±XXX.XX   ±XXX.XX
              XX  XX  XX.X     UNTIL
```

TABLE 4-IV.- DISPLAY PARAMETER DEFINITION TABLE

PROCESSOR OMP

RENDEZVOUS TARGETING MANEUVER TABLE

Display parameter label	Parameter definition
BASE DATE	Base date in years, months, days
GMTLO	GMT of lift-off in days, hours, minutes, and seconds from the base date
MAN	Maneuver number in the total sequence and the maneuver ID code
GMTI	GMT of the impulsive maneuver in days, hours, minutes, and seconds from the base date
DVLV-X	X-component of the LVLH delta-V vector
VG-X	X-component of the inertial delta-V vector
HA	Post maneuver apogee altitude in NM or KM
DH	Altitude difference between target and chaser at the maneuver point
YP	Out-of-plane distance at the maneuver point
DV	Total delta-V magnitude of the maneuver
VEH	Vehicle performing maneuver either Chaser or Target
GETI	GET of the impulsive maneuver in days, hours, minutes, and seconds from the lift-off time (GMTLO)
DVLV-Y	Y-component of the LVLH delta-V vector
VG-Y	Y-component of the inertial delta-V vector
HP	Post maneuver perigee altitude in NM or KM
PH	Phase angle between vehicles at the maneuver point in degrees
YPDOT	Out-of-plane velocity at the maneuver point
PITCH	LVLH maneuver pitch angle of the maneuver in degrees
T PROF	Thrust profile used for the maneuver
GMTTIG	GMT of engine ignition, using constant thrust and ISP of the thrust profile (T PROF) in days, hours, minutes, and seconds from the base date
DVLV-Z	Z-component of the LVLH delta-V vector
VG-Z	Z-component of the inertial delta-V vector
DT	Time between current maneuver and the next maneuver in the sequence, seconds
TBURN	The burn time for the maneuver, seconds
YAW	LVLH maneuver YAW angle of the maneuver
TIME	Time to the next lighting condition, hours, minutes, and seconds
UNTIL	Introduces the lighting condition
	The next lighting condition either LIGHT or DARK

TABLE 4-V.- PROCESSOR MESSAGE TABLE

PROCESSOR OMP

MSG no.	Message ID block	Message text block and explanation
1	*OMP*GMLOD*	XXXX IS INVALID MANEUVER TIME ID. Meaning: The user specified an incorrect maneuver time ID code. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to correct the oversight, then restart the execution.
2	*OMP*GMLOD*	NC MANEUVER HAS NO PHA OR EL CONSTRAINT SPECIFIED. Meaning: User did not supply either the phase angle or elevation angle for targeting the phasing maneuver. Severity: The processor will terminate and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to correct the oversight, then restart the execution.
3	*OMP*GMLOD*	NH MANEUVER HAS NO DH CONSTRAINT SPECIFIED. Meaning: User did not specify a differential altitude for targeting the height maneuver. Severity: Processor will terminate and control will be returned to the FDS Executive. Action required by user: Use the Interface Table Editor to correct the oversight and restart the execution.
4	*OMP*GMLOD*	NCC MANEUVER HAS NO DH CONSTRAINT SPECIFIED. ZERO ASSUMED. Meaning: User did not specify a differential altitude for the corrective combination Lambert maneuver to an offset. Processor will use a zero delta altitude. Severity: Warning only. The processor will continue. Action required by user: The user should examine the solution or subsequent error messages to determine if the desired solution was calculated.
5	*OMP*GMLOD*	NCC MANEUVER HAS NEITHER PHA OR EL CONSTRAINTS SPECIFIED. Meaning: User did not specify the phase offset for the corrective combination (Lambert) maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and then restart the execution.

TABLE 4-V.- Continued

PROCESSOR OMP

MSG no.	Message ID block	Message text block and explanation
6	*OMP*GMLOD*	TPF WAS NOT SPECIFIED AFTER THE TPI OR TPM MANEUVERS. Meaning: The final terminal phase maneuver was not specified, even though a TPI or TPM maneuver was. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
7	*OMP*GMLOD*	SOR WAS NOT SPECIFIED AFTER SOI OR SOM MANEUVERS. Meaning: The final stable orbit maneuver (SOR) was not specified, even though a TPI or TPM maneuver was. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
8	*OMP*GMLOD*	SOR MANEUVER HAS NO DR CONSTRAINT SPECIFIED. Meaning: The trailing displacement (DR) was not specified for the final stable orbit maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
9	*OMP*GMLOD*	HA OR HASH MANEUVER HAS NO HD CONSTRAINT SPECIFIED. Meaning: Desired altitude 180° from maneuver point was not specified for the HA or HASH maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
10	*OMP*GMLOD*	EXDV MANEUVER HAS NO DVLV CONSTRAINT SPECIFIED. Meaning: The LVLH delta-V vector was not specified for the external delta-V (EXDV) maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.

TABLE 4-V.- Concluded

PROCESSOR OMP

MSG no.	Message ID block	Message text block and explanation
11	*OMP*GMLOD*	LSDV MANEUVER HAS NO DVLS CONSTRAINT SPECIFIED. Meaning: The line-of-sight delta-V vector (DVLS) was not specified for the line-of-sight external delta-V maneuver (LSDV). Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor to correct the oversight and restart the execution.
12	*OMP*GMLOD*	NOSH MANEUVER HAS NO DNOD CONSTRAINT SPECIFIED. Meaning: The desired node shift (DNOD) was not specified for the node shift maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
13	*OMP*GMLOD*	PC MANEUVER HAS NO DPC CONSTRAINT SPECIFIED. Meaning: No wedge angle change (DPC) was specified for the single vehicle plane change (PC) maneuver. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.
14	*OMP*GMLOD*	IS AN INVALID MANEUVER ID. Meaning: The user specified an invalid maneuver ID code. Severity: Processor will terminate and control will return to the FDS Executive. Action required by user: Use the Interface Table Editor and correct the oversight and restart the execution.

TABLE 4-VI.- INTERFACE TABLE EXTENDED PROMPTS

PROCESSOR OMP

Processor name	Processor abstract prompt (maximum 256 characters)
OMP	The Orbital Maneuver Processor computes rendezvous and orbit shaping maneuvers that can result in the rendezvous of two vehicles.

Parameter keyword name	Parameter definition prompt (maximum 256 characters)
GLOCON	Global constants array; master data base element normally defaulted to !!GLCN
SESCON	Session constants array; defaults to !SESCN
SCOEF	Solar coefficients array; defaults to !SCOEF
SMCN	Sun and Moon constants array; defaults to !!SMCN
RMPCON	OMP constants and tolerances
CHASER	Chaser or active vehicle input state vector
TARGET	Passive or target vehicle input state vector
SVPROP	Propagation option selection code; either "CONIC", "AEG-S", or "AEG-L"
MAN 1 . . . MAN 9	Maneuver and vehicle ID of the form 'A,B' where B is CHASER or TARGET and A is NC, NH, NCC, NSR, TPI, TPM, TPF, NS, NPC, SOI, SOM, SOR, EXDV, LSDV, CIRC, PC, HA, HASH, NOSH, APSH, or END.

TABLE 4-VI.- Continued

PROCESSOR OMP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
TMOPT 1 . . . TMOPT 9	Maneuver time (or threshold time ID code and value(s)) and maneuver location and value of the form 'A=X' or 'A=X, B=X' where X is the value, A is T, DT, M, WT or N; B is EL, N, ALT, LAT, LON, NA, NP, CN, R, P, OPT, A, U, NITI, NITM, NITO, LITI, LITM, LITO.
CONVL 1 . . . CONVL 9	Constraint ID and value, of the form 'A=X, B=X, C=X, D=X' where X is the value and A, B, C, and D are DH, PHA, EL, DR, HD, DVLV, DVLS, DNOD, DPC, RDOT, DVCR, CPIT, SPIT.
DELTV 1 . . . DELTV 9	First guess of the NC or NH delta velocity magnitude
ENGIN 1 . . . ENGIN 9	Engine and thrust profile to be used for the maneuver of the form 'A,X' where A is OMS, RCSA, or RCSF, and X is the thrust profile. Refers to TFILEJ where J=1,5
TFILE 1	Thrust profile number 1 includes thrust and ISP in seconds
TFILE 2	Thrust profile number 2 includes thrust and ISP in seconds

TABLE 4-VI.- Concluded

PROCESSOR OMP

Processor name	Processor abstract prompt (maximum 256 characters)
Parameter keyword name	**Parameter definition prompt (maximum 256 characters)**
TFILE 3	Thrust profile number 3 includes thrust and ISP in seconds
TFILE 4	Thrust profile number 4 includes thrust and ISP in seconds
TFILE 5	Thrust profile number 5 includes thrust and ISP in seconds
APVTAB	Active or chaser output state vector phase table. States at the initial time and before and after each maneuver.
PPVTAB	Passive or target output state vector phase table. States at the initial time and before and after each maneuver.

5.0 PROCESSOR ROUTINES

The only available routine documentation is contained on the comment cards in the software listing.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1979—671-099/ 23 78

DISTRIBUTION FOR JSC IN 77-FM-18, REVISION 1, VOLUME III

JM6/Technical Library (2)
JM61/Center Data Management (3)
FD7/J. Fisher
FM/Chief
FM/E. Davis
FM2/Chief
    Section Heads (4)
    D. Alexander
FM4/Chief
    C. Graves
FM6/Chief
    A. Nolting
    D. Braley
    R. Davis (2)
    E. Fridge
    J. Martin
    G. Martinez
    R. Merriam
    W. Pruett
    W. Reini
    R. Reynolds
    R. Rogan
    G. Roush
    G. Weisskopf
FM8/Chief
FM14/Report Control Files (25)
    S. Cole
    B. Woodland
FM17/Chief
CSC/M30/O. Dial
    R. Herder
IBM/MC56/R. Turner (2)
MDTSCO/B. Brown (10)

Deletions or additions to this distribution must be coordinated through Gloria Martinez/FM6; 483-4491.