NASA CR-132691

MCR-75-143
NAS1-13611

Volume III

Final Report

April 1975

PROGRAMMER'S MANUAL

PROGRAM TO OPTIMIZE SIMULATED TRAJECTORIES (POST)

(NASA-CR-132691) PROGRAM TO OPTIMIZE SIMULATED TRAJECTORIES (POST). VOLUME 3: PROGRAMMER'S MANUAL Final Report (Martin Marietta Corp.) 242 p CSCL 22A

N75-32148

Unclas
G3/13 40816

Prepared by

G. L. Brauer, D. E. Cornick, A. R. Habeger,
F. M. Petersen, and R. Stevenson

Approved

D E Cornick

D. E. Cornick
Program Manager

MARTIN MARIETTA CORPORATION
P.O. Box 179
Denver, Colorado 80201

FOREWORD

This final report describing the formulation of the Program to Optimize Simulated Trajectories (POST) is provided in accordance with Part 3.0 of NASA Contract NAS1-13611. The report is presented in three volumes as follows:

Volume I - POST - Formulation Manual;

Volume II - POST - Utilization Manual;

Volume III - POST - Programmer's Manual.

This work was conducted under the direction of Mr. Joseph Rehder of the Space Systems Division, National Aeronautics and Space Administration, Langley Research Center.

CONTENTS

FINAL REPORT
PROGRAM TO OPTIMIZE SIMULATED TRAJECTORIES (POST)

VOLUME III - PROGRAMERS MANUAL

By G. L. Brauer, D. E. Cornick, A. R. Habeger,
F. M. Petersen, and R. Stevenson
Martin Marietta Corporation

SUMMARY

This report documents the program structure and logic, subroutine descriptions, and other pertinent programing information.

POST, a generalized point mass, discrete parameter targeting and optimization program, provides the capability to target and optimize point mass trajectories for a powered or unpowered vehicle operating near a rotating oblate planet. POST has been used successfully to solve a wide variety of atmospheric flight mechanics and orbital transfer problems. The generality of the program is evidenced by its N-phase simulation capability, which features generalized planet and vehicle models. This flexible simulation capability is augmented by an efficient discrete parameter optimization capability that includes equality and inequality constraints.

POST was originally written in Fortran IV for the CDC 6000 series computers. However, it is also operational on the IBM 370 and UNIVAC 1108 computers.

Other volumes in the final report are:

Volume I - Formulation Manual - Documents the equations and numerical techniques used in POST.

Volume II - Utilization Manual - Documents information pertinent to users of the program. It describes the input required and output available for each of the trajectory and targeting/optimization options.

I. INTRODUCTION

The program was written according to guidelines designed to provide complete generality wherever possible without sacrificing computational speed or computer storage. The guidelines adhered to are:

1) Computer core size of approximately 104 000 octal;

2) Fortran IV programming language;

3) Minimum program execution language;

4) Modular program construction;

5) Generalize routines to allow simulation of various types of vehicles;

6) Generality of input, output, targeting, and stopping variables;

7) Compatibility of operation on both 6500 and 6600 CDC computers using either MACE or SCOPE operating systems.

Information pertinent to the programmer is presented in the following sections of this report. Included are descriptions of the program logic, flow diagrams, Fortran symbols, and subroutine listings.

II. PROGRAM STRUCTURE AND LOGIC

POST is coded exclusively in FORTRAN IV. Overlays are used to minimize computer core requirements. The program requires approximately 104 000 octal cells of computer storage (see table II-1). More than 104 000 cells may be required to load the program, depending on the operating system used.

Executive programs are used throughout. These control the program flow by calling subroutines containing the actual mathematical formulations. This procedure allows the program to be modified quickly and easily.

TABLE II-1. - SUMMARY OF POST OCTAL CORE REQUIREMENTS

Overlay	Basic program	Scope 3.4.1 system	Subtotal (absolute)	Blank common	Operating total (absolute)
(0,0)	23 141	3 375	26 536	3 720	32 456
(1,0)	31 766	2 076	62 623		66 543
(2,0)	24 777	1 006	54 544		60 464
(2,1)	5 103	0	61 647		65 567
(2,2)	15 205	74	72 046		75 766
(2,3)	21 147	405	77 674		103 614
(2,5)	6 103	0	62 651		66 576
(2,6)	1 676	0	56 442		62 362
Total required			77 674		103 614

Overall Program Logic

POST is structured in three overlay levels, as shown in figure II-1. The first overlay (0,0) is the master executive overlay, which controls the overall program. This overlay controls the read-in of input data and determines which trajectory computations are to be performed.

Overlay (0,0) first calls overlay (1,0), which reads the namelist input data from cards and stores the processed data on disc for later use.

Overlay (2,0) is called by (0,0) after (1,0) has completed the input processing tasks. The first decision in overlay (2,0) concerns the type of simulation; i.e., single trajectory or search/optimization mode. If a single trajectory is to be run, the program calls overlays (2,1), (2,2), and (2,3) sequentially, then returns to the master overlay (0,0). If the search/optimization mode is to be used, the program control is turned over to subroutine MINMYS, which calls overlays (2,1), (2,2), (2,3), (2,5), and (2,6) as required to perform the search/optimization function. When convergence has been achieved or the maximum number of iterations has been exceeded, control reverts back to the master overlay (0,0) for the next problem.

An outline of the approximate calling sequence for each routine is presented in the following section of this report. This outline shows which subroutines are called by a given routine, thereby allowing the detailed logic flow to be followed easily. The overall program logic described by the overlays is as follows:

1) Overlay (2,1) reads the previously processed input data from tape, locates the data for the current phase (event), and initializes the program values based on this input;

2) Overlay (2,2) initializes the equations of motion for the current phase;

3) Overlay (2,3) integrates the equations of motion from time t_i to a specified stopping condition for the current phase;

4) Overlay (2,5) calculates the control corrections based on the search/optimization algorithm being used, limits the control parameters that violate the control parameter constraints, and tests for convergence;

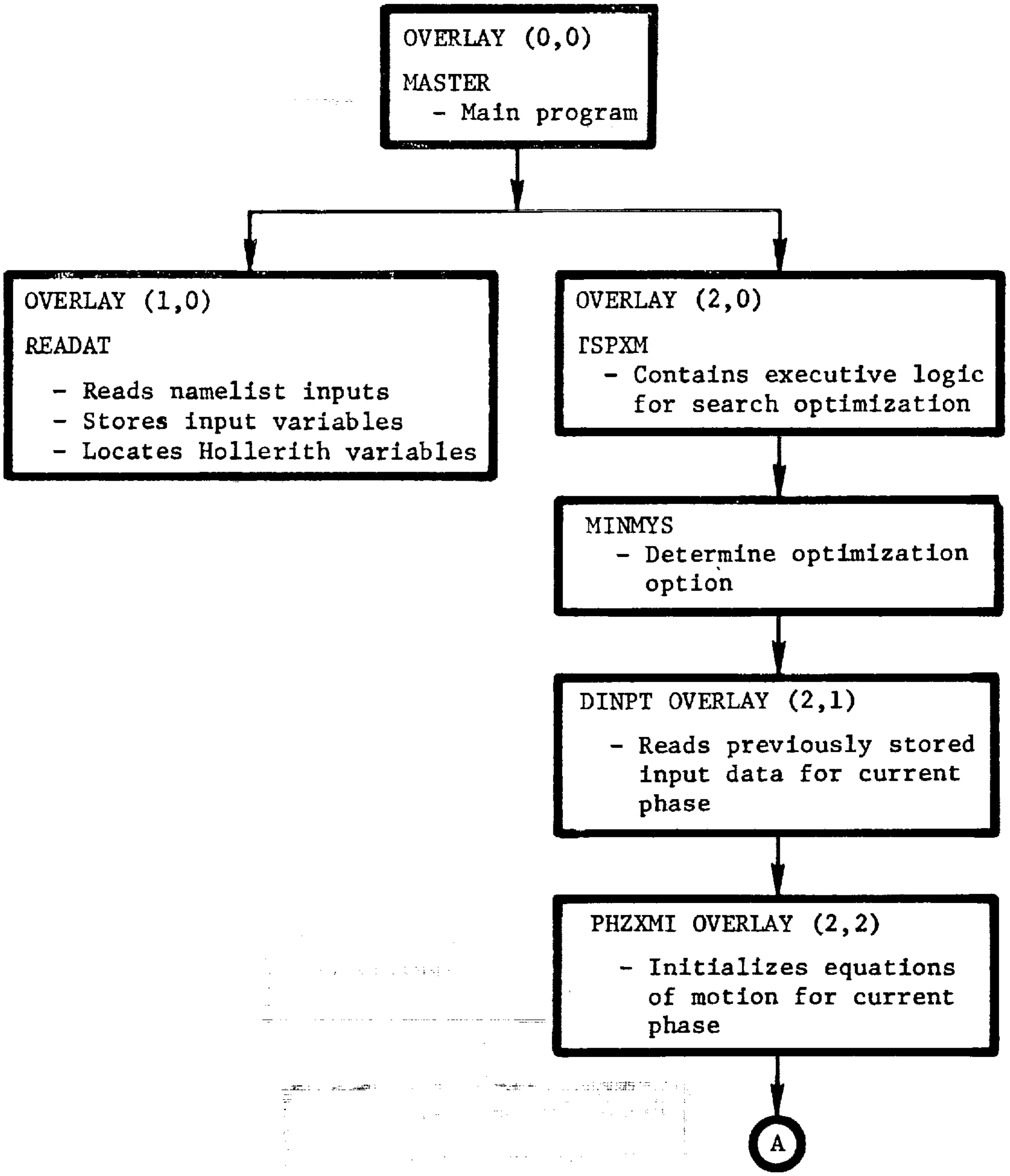

OVERLAY (0,0)
MASTER
- Main program
OVERLAY (1,0)
READAT
- Reads namelist inputs
- Stores input variables
- Locates Hollerith variables
OVERLAY (2,0)
ΓSPXM
- Contains executive logic for search optimization
MINMYS
- Determine optimization option
DINPT OVERLAY (2,1)
- Reads previously stored input data for current phase
PHZXMI OVERLAY (2,2)
- Initializes equations of motion for current phase
A

Figure II-1. - Program Macrologic

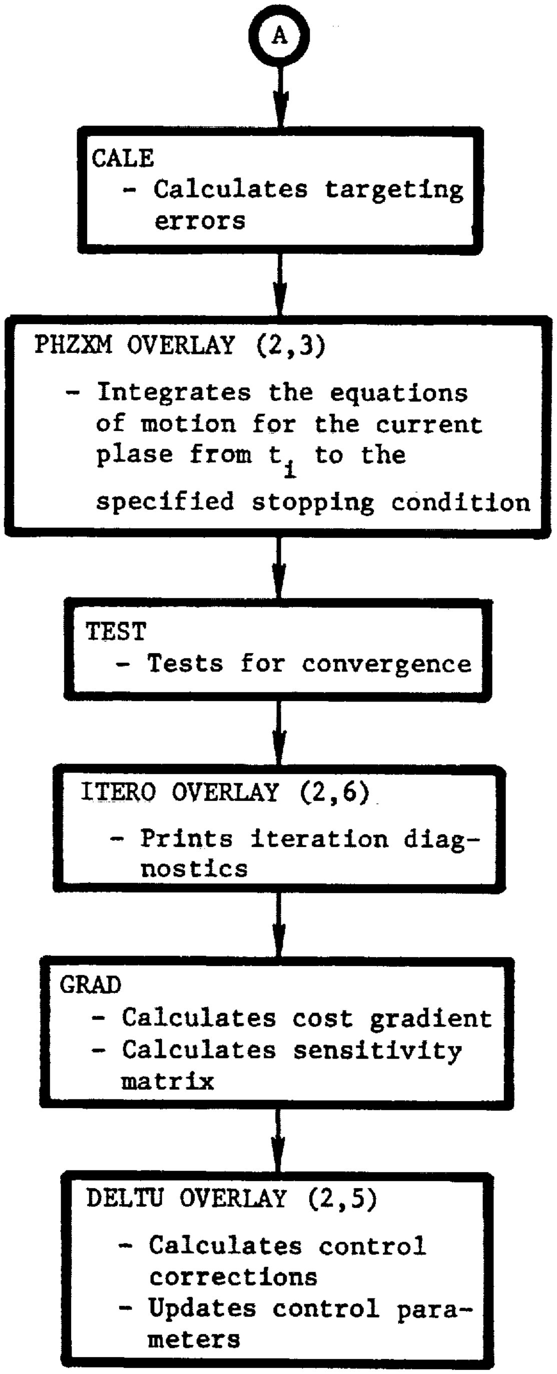

A
CALE
- Calculates targeting errors
PHZXM OVERLAY (2,3)
- Integrates the equations of motion for the current plase from t_i to the specified stopping condition
TEST
- Tests for convergence
ITERO OVERLAY (2,6)
- Prints iteration diagnostics
GRAD
- Calculates cost gradient
- Calculates sensitivity matrix
DELTU OVERLAY (2,5)
- Calculates control corrections
- Updates control parameters

Figure II-1.- Concluded

5) Overlay (2,6) prints out an iteration summary at the end of each iteration. It also performs any other information output tasks required by search/optimization algorithm, such as printing trial step summaries.

The program dictionary (subroutine DICT) performs a one-to-one mapping of variables in common and the Hollerith names by which the user can select the variables for a variety of uses, including output, stopping conditions, control variables, and targeting variables.

All variables in the dictionary are located in common with respect to two labeled commons, IV and END. The first of these commons defines the starting reference; the last defines the ending reference. These commons must bracket all commons required by the dictionary.

POST uses a generalized table storing and look-up procedure whereby the size of tables is limited only by the total data storage allocation of 1500 cells. Each table has its own multiplier. This is accomplished by dimensioning the table by (2). The first location contains the address of the table and the second location contains the table function multiplier. The generalized table lookup (GENTAB) is set up to handle all allowable types of tables, namely, constant-value, monovariant, bivariant, and trivariant.

Outline of Program Logic

This outline shows the calling sequence for a single iteration for the trajectory and optimization logic. Certain routines are called only if certain options are requested. These routines are presented in their approximate calling sequence. The outline allows the user to follow the program flow either forward or backward from a given routine to aid in understanding the logic flow. Each subroutine that is called by another routine is listed immediately below and to the right of the calling routine.

```
1.1 MASTER(OVERLAY(POST00,1,0)),BLKDAT
    2.1 READAT(OVERLAY(POST10,1,0)),DICT
        3.1 RSEARCH
            4.1 NMLTER
        3.2 RGENDAT
            1.1 NMLTER
        3.3 RTBLMLT
            4.1 NMLTER
        3.4 RTAB
            4.1 NMLTER
        3.5 INPUTX
    2.2 TSPXM(OVERLAY(POST20,2,0)),DATA
        3.1 TRAJX
            4.1 CLSPFL
        3.2 MINMYS
            4.1 NOMINL
                5.1 SETIC
                5.2 TRAJ
                    6.1 DINPT(OVERLAY(POST21,2,1)
                        7.1 SETESN
                    6.2 SAVIC
                    6.3 SETIV
                    6.4 PHZXMI(OVERLAY(POST22,2,2)
                        7.1 CYCXMI
                            8.1 DYNXMI
                                9.1 DYNXA (OR DYNXB)
                                    10.1 DERVI
                                         11.1 MOTIAL
                                              12.1 ORBTR
                                              12.2 WINDS
                                              12.3 GUIDI
                                                   13.1 QUAT1
                                                   13.2 QUAT2
                                              12.4 GUID1
                                                   13.1 OLGM
                                                   13.2 OLGOM
                                              12.5 IBMTRX
                                              12.6 WGTINI
                                              12.7 INTGRL
                                              12.8 ATMOS
                                    10.2 DYSI1
                                         11.1 DLOOK
                            8.2 AUXFMI
                                9.1 EPHEM
                                    10.1 FORMN
                                    10.2 SUN
                            8.3 TGOEMI
                            8.4 INFXMI
                                9.1 PAGER
                                9.2 PRNTIC
```

```
6.5 CALE
6.6 PHZXM(OVERLAY(POST23,2,3)
    7.1 TGOEM
    7.2 CLGM
        8.1 GSENSR
        8.2 GNAV
        8.3 GGUID
        8.4 GCNTRL
    7.3 CYCXM, OR CYCYM
    7.4 DYNXM
        8.1 DYNS1, DYNS2, OR DYNS3
            9.1 RUK, SVDQ, OR TWOBDY
            9.2 DERIV
                10.1 GSA
                     11.1 UPNOMS
                     11.2 MOTION
                     11.3 AUXFM
                     11.4 CALSPE
                     11.5 CALES
                     11.6 GRADS
                10.2 MOTION, OR MOTENC
                     11.1 ATMOS
                          12.1 ATMOS1, OR
                               ATMOS2, OR
                               ATMOS3
                     11.2 WINDS
                     11.3 GUID1
                          12.1 OLGM
                          12.2 OLGOM
                     11.4 IBMTRX
                     11.5 AERO
                     11.6 PROP
                          12.1 TRIM
                               13.1 XITER
                     11.7 TMOTM
                          12.1 GRAV
                     11.8 AEROHI
                     11.9 GUID2
                          12.1 OLGM
    7.5 AUXFM
        8.1 GAMLAM
        8.2 DGAMLAM
        8.3 BACKOI
        8.4 BACKOR
        8.5 XRNGE1
        8.6 XRNGE2
        8.7 DPRNG
        8.8 CONIC
        8.9 MONITR
        8.10 HSWGT
        8.11 TRACKER
        8.12 ANMPT
        8.13 CALSPEC
```

```
            7.6 INFXM
                8.1 CONVO
                8.2 PAGER
                8.3 PBLOCK
            7.7 CLSPFL
    5.3 TEST
    5.4 ITERO(OVERLAY(POST26,2,6)
    5.5 SETIC
    5.6 TRAJ
4.2 GRAD
    5.1 SETIC
    5.2 TRAJ
    5.3 PAD
4.3 DELTU(OVERLAY,POST25,2,5)
    5.1 WUCAL
    5.2 GMAC
    5.3 SDM, OR
    5.3 CGM, OR
    5.3 DGMP2, OR
        6.1 DGM
    5.3 PGM
        6.1 UPDATS
        6.2 DROP
            7.1 REVISE
                8.1 UPDATS
            7.2 COMBIN
            7.3 UPDATS
        6.3 DGM
    5.4 UNITDU
    5.5 GABDD
4.4 TRYIT1
    5.1 GENMIN
        6.1 FGAMA
            7.1 SETIC
            7.2 TRAJ
            7.3 ITERO(OVERLAY,POST26,2,6)
        6.2 TPOSM
        6.3 THPOSM
        6.4 BUCKET
        6.5 THPM
        6.6 FCPMIN
    5.2 DELTU(OVERLAY,POST25,2,5)
    5.3 UPNOM
    5.4 DELTU(OVERLAY,POST25,2,5)
4.5 TRYIT2
    5.1 FGAMA
    5.2 GENMIN
4.6 UPNOM

(RETURN TO 3.2 AND REPEAT UNTIL CONVERGED)
```

Tape or File Designations

The program uses several Tape (File) designations internally to perform the simulation tasks. These files are normally stored on discs, but tapes can be used by assigning them the proper file designations. The file designations are as follows:

Tape (or File)	Definition
1	Contains the general data and table multipliers for the problem
2	Contains the initial conditions for each event that has a control parameter
3 and 4	Store input data for multiple runs
5 (INPUT)	Stores input data
6 (OUTPUT)	Stores output data
8 (PROFIL)	Contains the simulation profile

Common Designations

POST uses several labeled commons to provide communication between subroutines. In addition, a blank common is used to act as a data buffer for the table input data and the event criteria. The blank common could be labeled, if desired, without adversely affecting the operation of the program.

The labeled commons are briefly described below in alphabetical order. The variables are listed in the following section alphabetically to provide an easy cross-reference.

AUXVC: Common AUXVC contains the variables that are computed as auxiliaries at the end of each integration step.

CYCVC: Common CYCVC contains variables and flags used to perform cycling functions during forward integration.

DPGVC: Common DPGVC contains the variables and flags associated with the guidance (steering) options.

DYNVC: Common DYNVC contains variables and flags required to perform dynamics functions during the forward integration. Primarily this includes time references and discontinuity flags.

DYTEM: Common DYTEM contains variables and storage used by the integration algorithms to integrate the equations of motion forward. No variable in this common may be input or output.

END: Common END is used to define the end of the dictionary. Any variable defined in a common after common END cannot be input, output, or used as a search parameter. This common contains only one variable, namely, END.

GENIC: Common GENIC contains variables of a general nature that are required in overlay (0,0).

GUIDIC: Common GUIDIC contains the input variables for the generalized guidance routines.

GUIDVC: Common GUIDVC contains the computed variables for the generalized guidance routines.

HØLINC: Common HØLINC contains all of the Hollerith input variables.

INFVC: Common INFVC contains variables and flags that may be used in the information output routines at any phase.

IV: Common IV is used to define a reference to the dictionary region. All variables that are to be input, output, or used as search parameters must be defined in a common between common IV and common END. IV contains the size of this region. This common contains one variable, namely, IV(2).

KRØIC: Common KRØIC contains input variables for the variable step/order predictor-corrector integrator.

KRØVC: Common KRØVC contains variables and flags calculated in the variable step/order predictor-corrector integrator.

LØCAL: Common LØCAL contains parameters used in computing the equations of motion and the auxiliary equations that are not required to be input or output. If a common variable is to be added and it is not needed as an output or an input, it should be added to this common.

MNMMLT: Common MNMMLT contains a list of mnemonic multipliers associated with the aerodynamic tables. The first cell contains the value 1.0. The remaining cells contain the address of any input variable within the dictionary.

MØTBL: Common MØTBL defines all tables to be interpolated by the general table lookup routine GENTAB. Each table requires two consecutive storage locations. The first is the table address and the second is the value of the table multiplier. Whenever a table is added to this common, subroutines DICT and DATA must be modified accordingly.

MØTIC: Common MØTIC contains all parameters that are required as inputs to the equations of motion. Input parameters do not have to be defined in this common; however, when such a parameter is defined in a lower common (e.g., MØTVC), the program must search for the dictionary and, hence, runs longer.

MØTVC: Common MØTVC contains all variables used in the equations of motion. These are generally not input or constant parameters. They are available for output, table arguments, or search parameters through the dictionary.

MULTRC: Common MULTRC contains the variables associated with the multiple-run capability.

ØVRLY25: Common ØVRLY25 contains the variables required by overlay (2,5), which contains the direction-of-search logic.

PHZVC: Common PHZVC contains flags and constants required to perform the phasing fun-tions.

REDAT: Common REDAT is defined in overlay (1,0) by READAT and contains variables and storage data required to build the general and table data buffers.

SEARC: Common SEARC is defined in BLKDAT and, in general, contains all parameters required by the iteration algorithms. Variables in common SEARC can be input only once per run through namelist SEARCH. They cannot be changed through input at a phase. Since SEARC is defined in overlay (0,0), it is available to every routine in the program.

SERVC: Common SERVC is a service common available to all routines in the program. This common contains 50 cells of temporary storage, 5 commonly used index parameters, and a list of the most frequently used fixed- and floating-point constants. This common should be used whenever possible in order to conserve storage.

SPECAL: Common SPECAL contains the variables associated with the special calculation routine CALSPEC.

TARGVC: Common TARGVC contains parameters calculated for the target vehicle.

TGOVC: Common TGOVC contains variables and flags required to perform the time-to-go functions.

TRACKC: Common TRACKC contains the variables associated with the tracking station option.

Program Additions

The guidance, navigation, and flight control routines will generally be coded by the user. The coding of these routines may require the user to make minor program additions. The most frequently requested types of program additions are: addition of new general variables, addition of generally new tables, and addition of new integrals. Instructions for making these additions are presented in this section. Other types of additions will generally require in-depth knowledge of the program code, and a programmer familiar with the program should be consulted.

Addition of new general variables.- General variables are any variable that are computed in the simulation portion of the program and are to be input, output, used as table arguments, search parameters, integrals, or derivatives. The program Executive processing algorithm expects to find all general variables defined in a labeled common which is loaded between the labeled common /IV/ and the labeled common /END/. The labeled commons /IV/ and /END/ are defined in subroutine DICT for overlay (1,0) and in subroutine DATA for overlay (2.0). The labeled commons defined between /IV/ and /END/ must be in the same order and must be the same size in subroutine DICT [overlay (1,0)] and in subroutine DATA [overlay (2,0)]. That is, a one-to-one mapping of parameters in subroutine DICT to subroutine DATA relative to labeled common /IV/ must be maintained; /END/-/IV/ in overlay (1,0) must equal /END/-/IV/ in overlay (2,0). This is absolutely required for proper program operation.

The following steps should be followed to add a general variable:

1) Add the new variable(s) to an appropriate labeled common. For example, if the new variable is an auxiliary parameter, it should be added to labeled common /AUXVC/. New variables should be added on to the end of an existing common. Only the length of the common should change, NOT the structure.

   If the user does not want to add to an existing common, a new labeled common may be defined, and the new variables included in it. However, this is not generally necessary.

2) The labeled common to which the new variables have been added, or the new labeled common, is replaced or added into subroutine DICT in overlay (1,0). If a new labeled common is being added it must be placed after common /IV/ and before common /END/; but not between common /MOTBL/ and common /MOTEND/. The locations from common /MOTBL/ through common /MOTEND/ are reserved for tables.

3) For every new variable added, the Hollerith name by which it is to be known must be set into its location for use during input processing. This is done by a DATA statement in subroutine DICT. For example, if a new variable called AROANG is added, then the data statement DATA AROANG/6HAROANG/ must be added in the subroutine DICT.

4) If the new variable is going to be on input quantity it must be added to the NAMELIST/GENDAT/. The input NAMELIST/GENDAT/ is defined in subroutine RGENDA in overlay (1,0). Include the new or updated labeled common and add the new input variable to the namelist.

5) Subroutine DATA, in overlay (2,0), establishes the initial or nominal values of the variables to be used in the simulation. Every new variable must have a nominal value set, even if it is zero. Add the new common, or update existing common with new variables, in the subroutine DATA.

6) Add data statement in subroutine DATA to set nominal value of new variable.

7) Add or change common for new variable in routines where it is to be used. Add necessary coding to perform computations involving new variable.

Adding new tables.- The program has a generalized table accessing feature that allows new tables to be added without adding dimensional arrays, hard-wired table arguments, table types, table dimensions, etc. The program input processor packs all tables input by the user, into an array in blank common. At execution time, the table interpolation routine, GENTAB, is directed to a particular table in the blank common array by a pointer, which is set at data initialization at the beginning of each phase. Thus, each table has a pointer associated with it. Each table also has a multiplier associated with it, by which the table is scaled during execution. To add a new table the user need only add the table pointer and the table multiplier. Because the pointer and multiplier can change from phase to phase, they are included in the general data area of program. That is, they are defined between labeled common /IV/ and labeled /END/ as they are specified in Subroutine DICT and DATA for overlays (1,0) and (2,0), respectively. The table input processor expects to find all table pointers and multipliers together, and in pairs. The pairs must be defined between labeled common /MOTBL/ and labeled common /MOTEND/, as declared in Subroutine DICT and Subroutine DATA. A new table pointer and multiplier should be added to labeled common /MOTBL/. A new labeled common could be declared between common /MOTBL/ and /MOTEND/, but this is generally not necessary.

To add a new table the following procedure is to be followed:

1) Add two locations to labeled common /MOTBL/ for the table pointer and multiplier.

2) Replace labeled common /MOTBL/ in Subroutine DICT in overlay (1,0).

3) Add data statement in subroutine DICT to set table name in pointer, and Hollerith name of multiplier into table multiplier. For example, if a new table called EMFT is to be added, then EMFT(2) is added to common /MOTBL/. In Subroutine DICT the data statement DATA EMFT/4HEMFT, 6HEMFTM/ is added. This sets the table name, EMFT, and the table multiplier EMFTM for the input processor.

4) Table multipliers are input through namelist /TBLMLT/. Thus the table multiplier must be added to namelist /TBLMLT/ in subroutine RTBLML in overlay (1,0). Replace the labeled common /MOTBL/. Include equivalence statement to equate table multiplier with desired input name. Add input name to namelist /TBLMLT/. For example, EQUIVALENCE (EMFT(2), EMFTM); add EMFTM to namelist /TBLMLT/.

5) Replace labeled common /MOTBL/ in subroutine DATA in overlay (2,0).

6) Add data statement in subroutine DATA to set table pointer to zero and table multiplier to desired nominal value. Generally the table multiplier will be set to 1.0. For example, DATA EMFT/0 , 1.0 /.

7) To reference the new table add or replace labeled common /MOTBL/ in routine where table look-up is to be performed. To perform the interpolation, the interpolation routine GENTAB is called with the table pointer as an argument. For example:

   ```
   VOLT = GENTAB (EMFT)
   ```

   If the table is not input, GENTAB will return as zero.

<u>Adding new integrals</u>.- Any general variable computed in the simulation model can be integrated provided it satisfies the necessary conditions of differentiable and continuity as required by the integration algorithms. The variables must be computed in the inner loop of the simulation, and be defined as a general variable in Subroutine DICT and Subroutine DATA. The program determines which variables are to be integrated during any pha

from an integration list, which is defined in BLKDAT. The integration list contains three entries for each integral. These entries are the integral name, the derivative name, and a flag to indicate whether this integral is to be integrated or not. During phase initialization this flag can be set to turn the integration on or off. The integration list is defined in labeled common /DYNIL/ in BLKDAT. The first location in labeled common /DYNIL/ contains the total size of the list including itself. Thus, to add an integral the common /DYNIL/ must be increased by three, and the contents of DYNIL(1) increased by three.

To add a new integral the following procedure should be followed:

1) In BLKDAT, overlay (0,0), increase the dimension of DYNIL in labeled common /DYNIL/ by three for each new integral.

2) In the associated data statement increase the number prestored into DYNIL(1) by three for each integral added.

3) Add the DATA statment to set the Hollerith name of the integral, of the derivities and a nominal value of 0 or 1, depending when the integral is to be nominally off or on, into three new locations defined in DYNIL.

   For example, to add DYNPI as the integral of DYNP, the following DATA statement should appear.

   ```
   DATA DYNIL/M, 6HTIME, 6HDTIME , 1
               ,       .
               ,       .
               ,       .
                 6HDYNPI, 6HDYNP , 0  /
   ```

4) Add integral and derivative, if required, to simulation as described previously under addition of general variables.

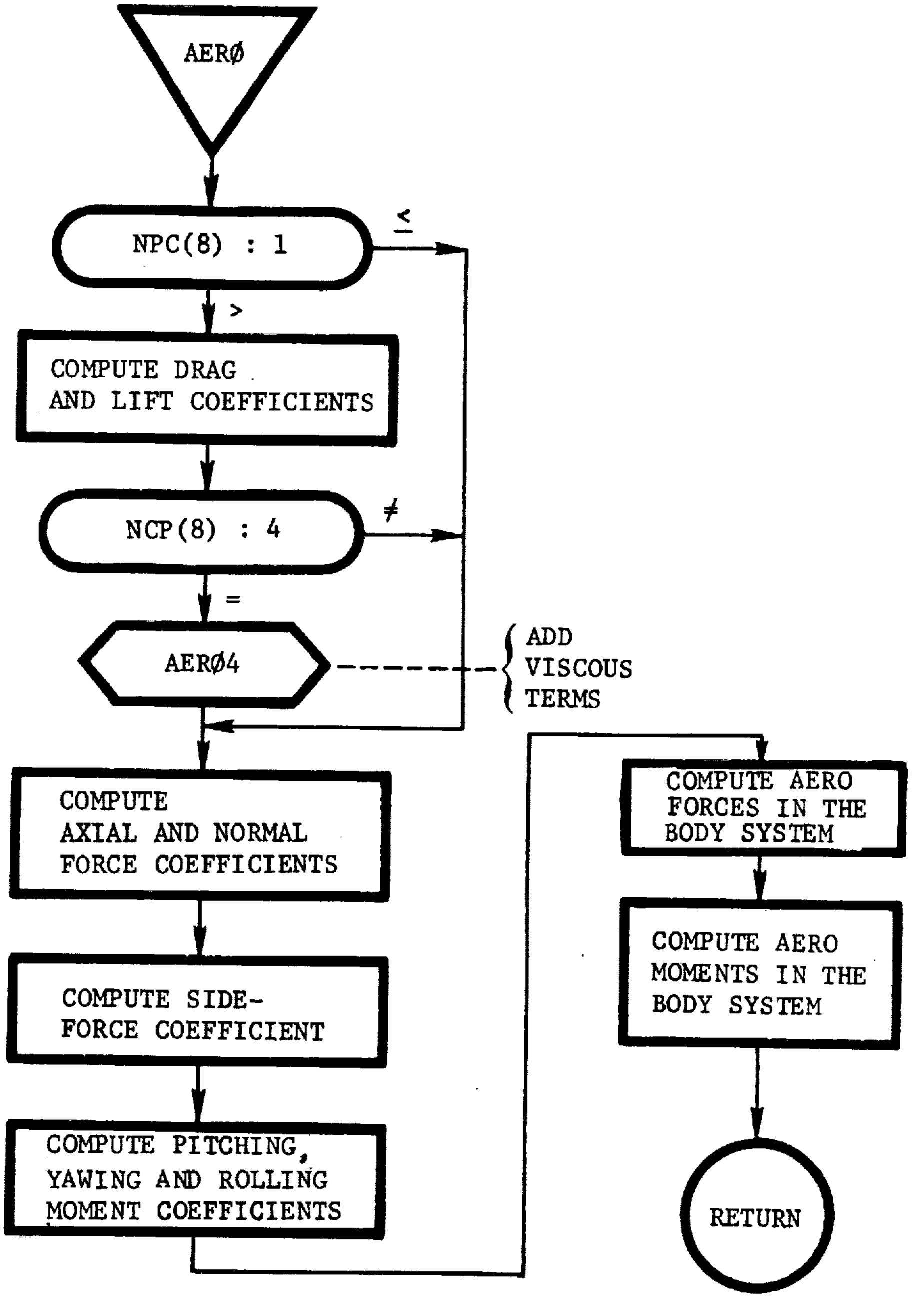
5) If the user desires to turn the integral on or off as a function of input, or model selected, then, the associated flag must be set in the integration list in Subroutine MOTIAL (overlay 2,2). The utility routine INTGRL can be used. The first argument is the position of the integral in the list, the second is the number of integrals to be set, and the third the flag zero or one.

III. SUBROUTINE DESCRIPTIONS, FLOW CHARTS, AND SELECTED LISTINGS

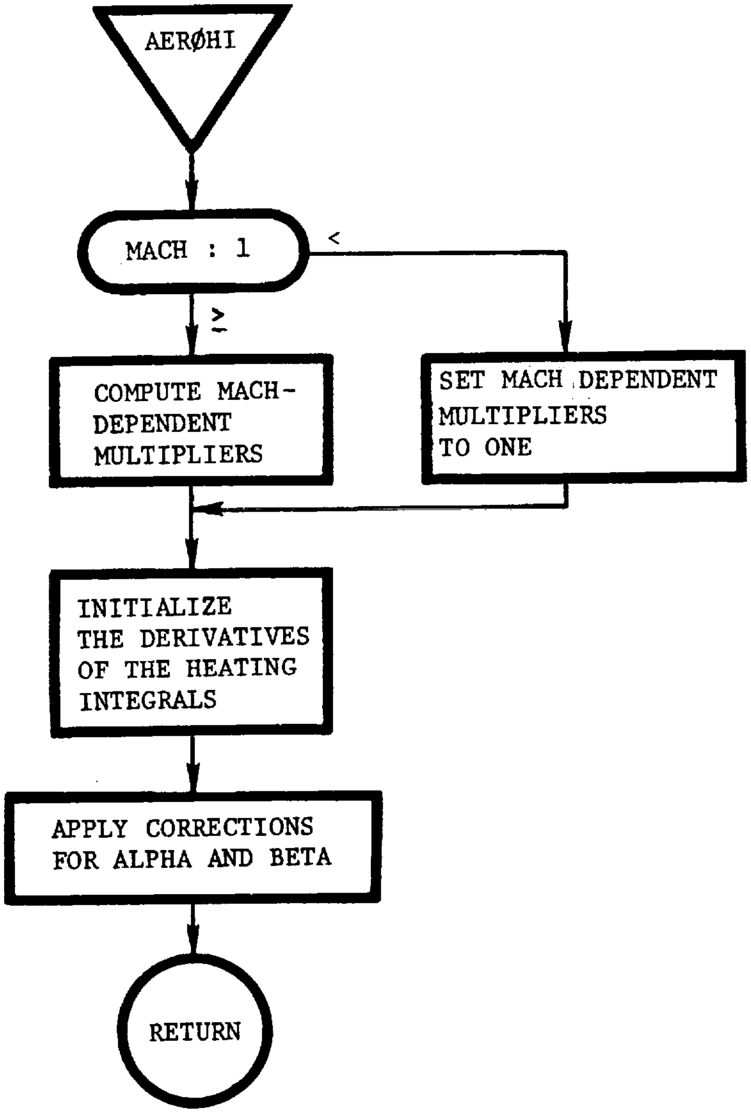
This section describes the subroutines used in the program. Flow charts and/or listings are also presented in order to show the detailed operation of the subroutines.

Note that the routines are presented alphabetically, rather than in the order shown in the previous outline of program logic. The outline enables the user to follow the program logic flow from one subroutine to another with a minimum of searching to find the next routine, but this alphabetical listing makes it easier to find subroutines at random.

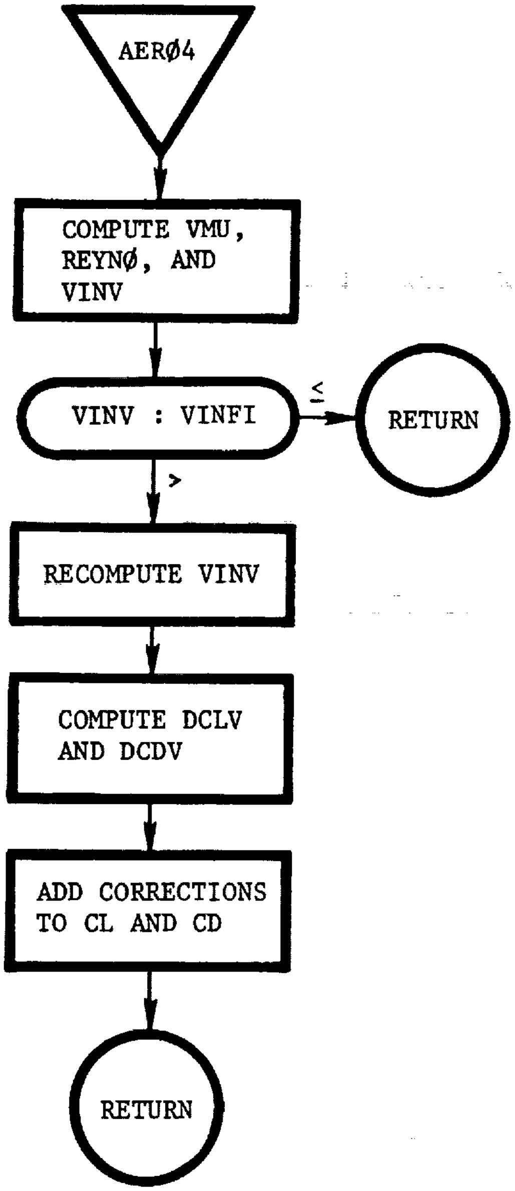
AERØ: This routine calculates the aerodynamic forces and moments in the body coordinate system.


AERØ
NPC(8) : 1
≤
>
COMPUTE DRAG AND LIFT COEFFICIENTS
NCP(8) : 4
≠
=
AERØ4
ADD VISCOUS TERMS
COMPUTE AXIAL AND NORMAL FORCE COEFFICIENTS
COMPUTE SIDE-FORCE COEFFICIENT
COMPUTE PITCHING, YAWING AND ROLLING MOMENT COEFFICIENTS
COMPUTE AERO FORCES IN THE BODY SYSTEM
COMPUTE AERO MOMENTS IN THE BODY SYSTEM
RETURN

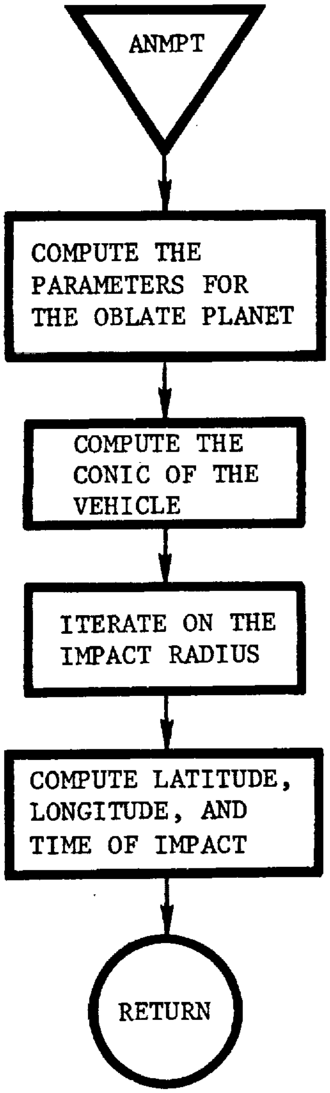
AERØHI: This routine calculates aeroheating indicators that are functions of angle of attack, sideslip, and Mach number.


AERØHI
MACH : 1
<
≥
COMPUTE MACH-DEPENDENT MULTIPLIERS
SET MACH DEPENDENT MULTIPLIERS TO ONE
INITIALIZE THE DERIVATIVES OF THE HEATING INTEGRALS
APPLY CORRECTIONS FOR ALPHA AND BETA
RETURN

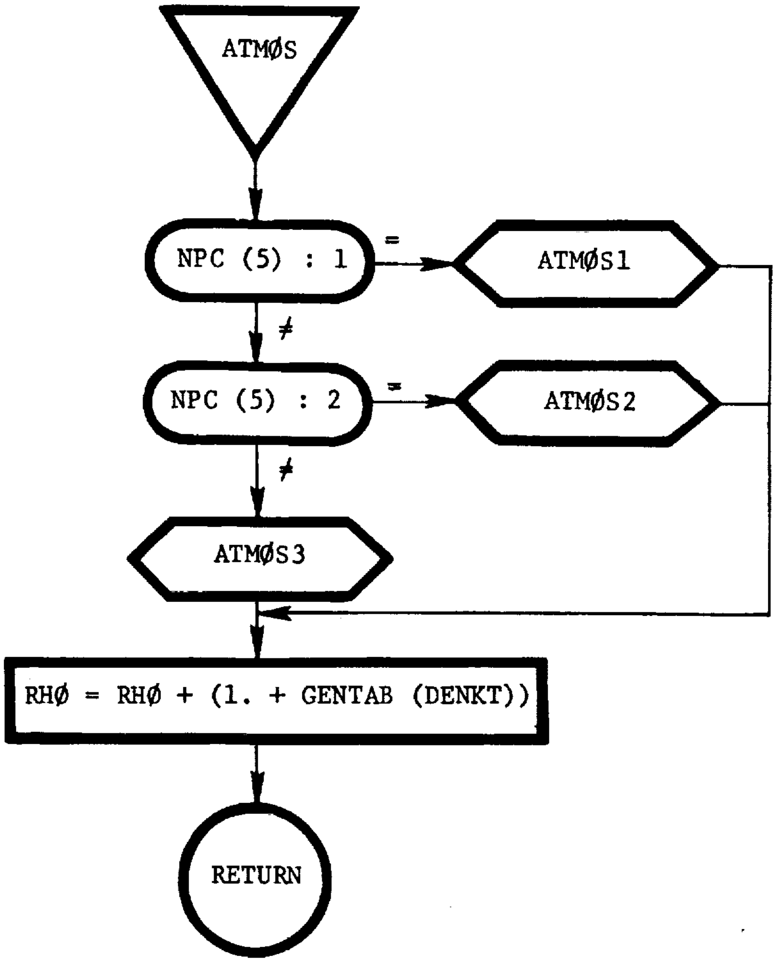
AERØ4: This routine calculates the corrections to the lift and drag coefficients (DCLV and DCDV) to be applied to CL and CD to account for viscous interaction effects.



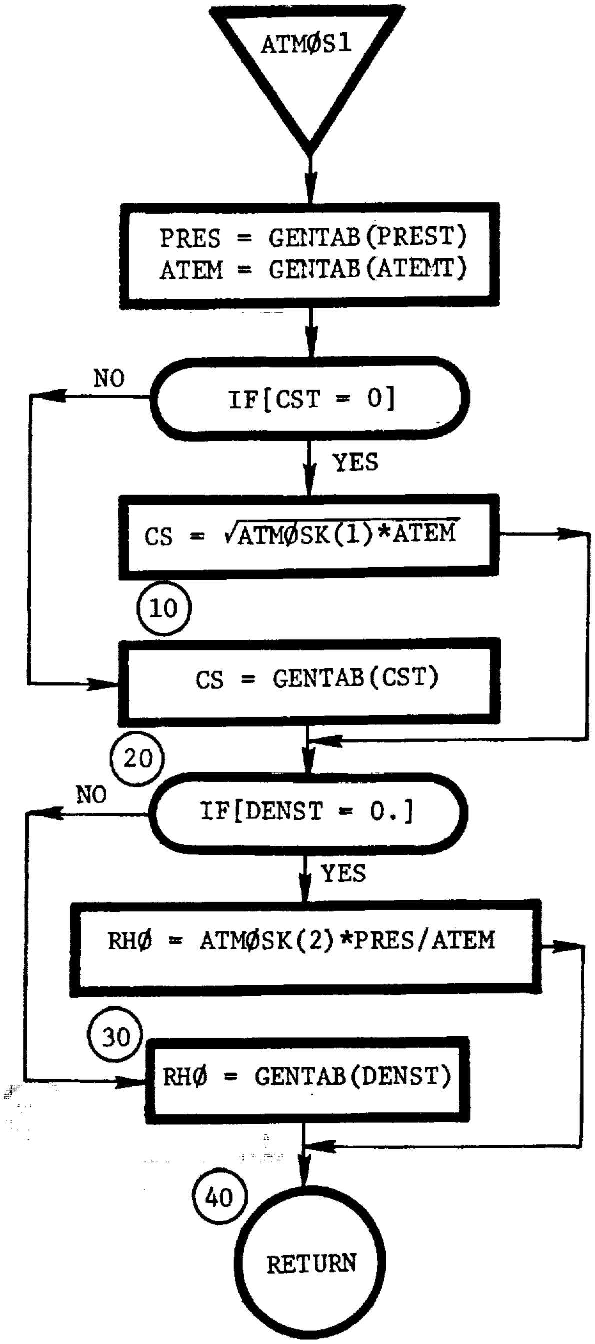
ANMPT: This routine calculates the vacuum impact point of the vehicle and the oblate planet at the specified altitude (ALTIP).


ANMPT
COMPUTE THE PARAMETERS FOR THE OBLATE PLANET
COMPUTE THE CONIC OF THE VEHICLE
ITERATE ON THE IMPACT RADIUS
COMPUTE LATITUDE, LONGITUDE, AND TIME OF IMPACT
RETURN

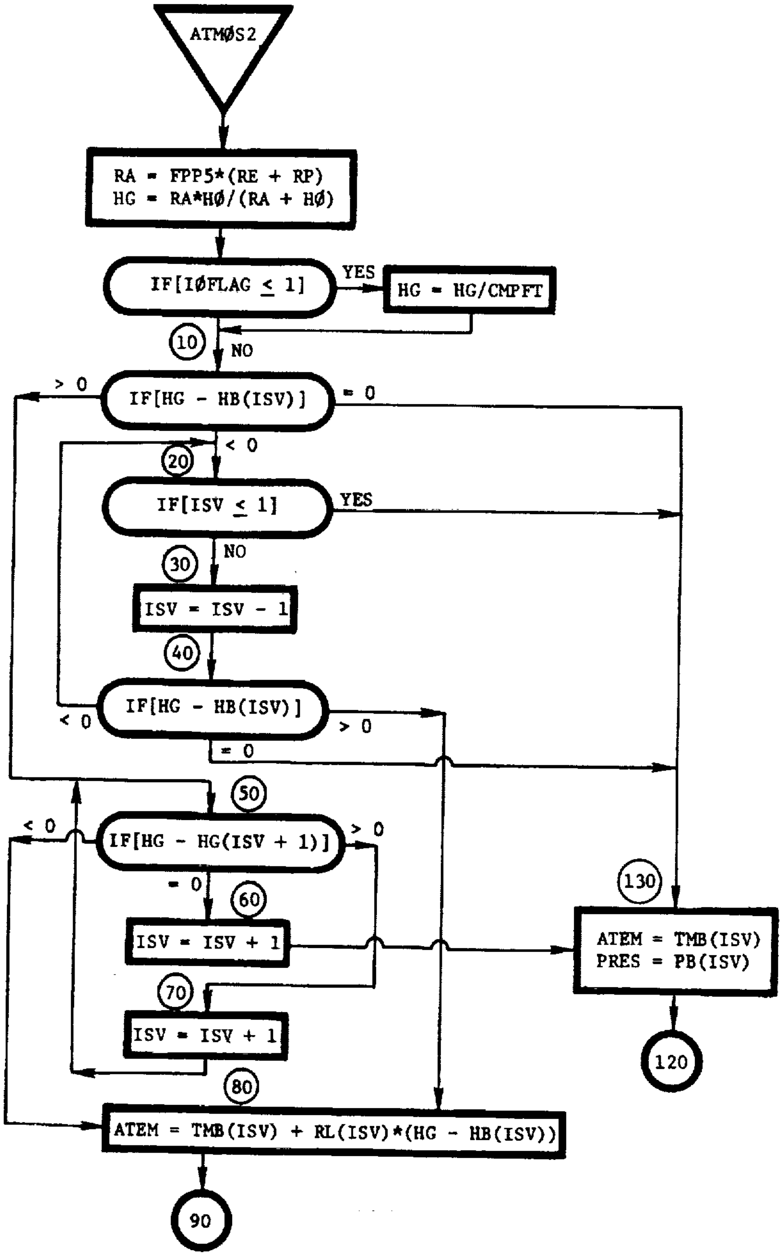
ATMØS: This routine determines which atmosphere model is to be used.

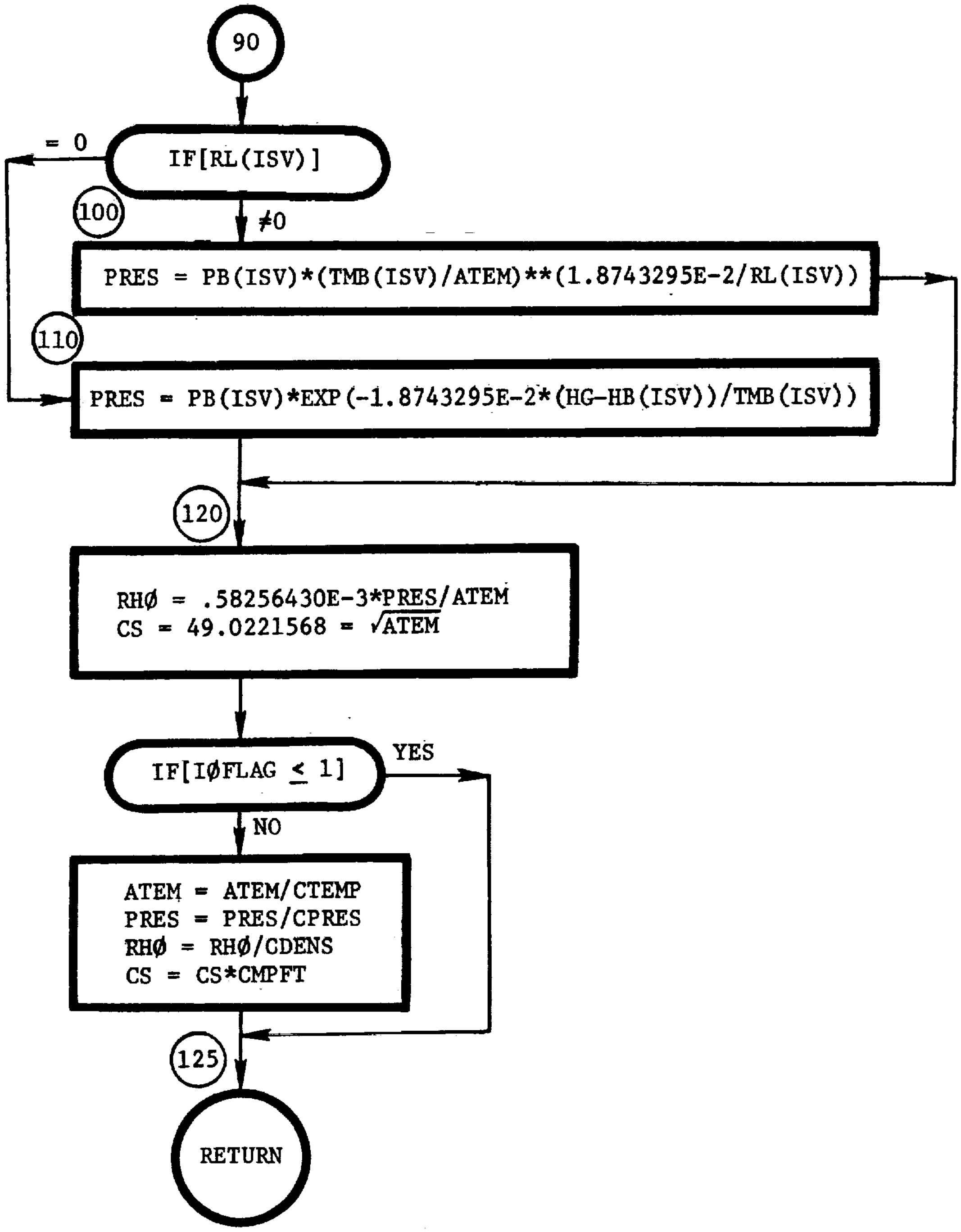


ATMØS1: This routine computes the atmospheric parameters using generalized table lookups.

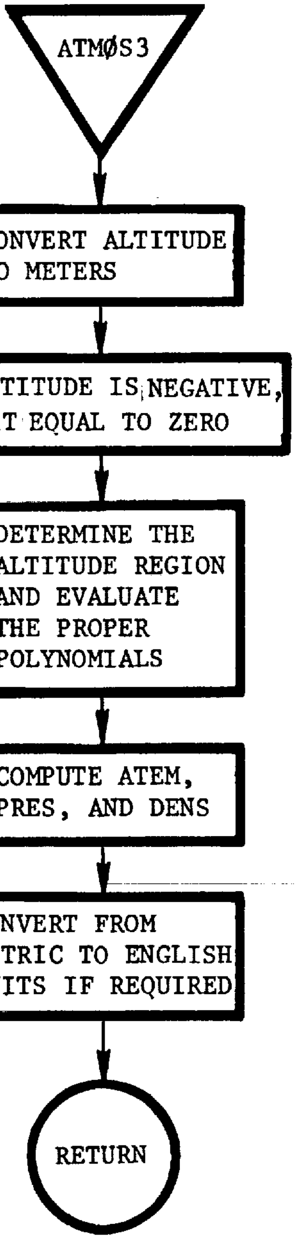


ATMØS2: This routine computes the atmospheric parameters based on the 1962 U.S. Standard atmosphere model as a function of geopotential altitude.

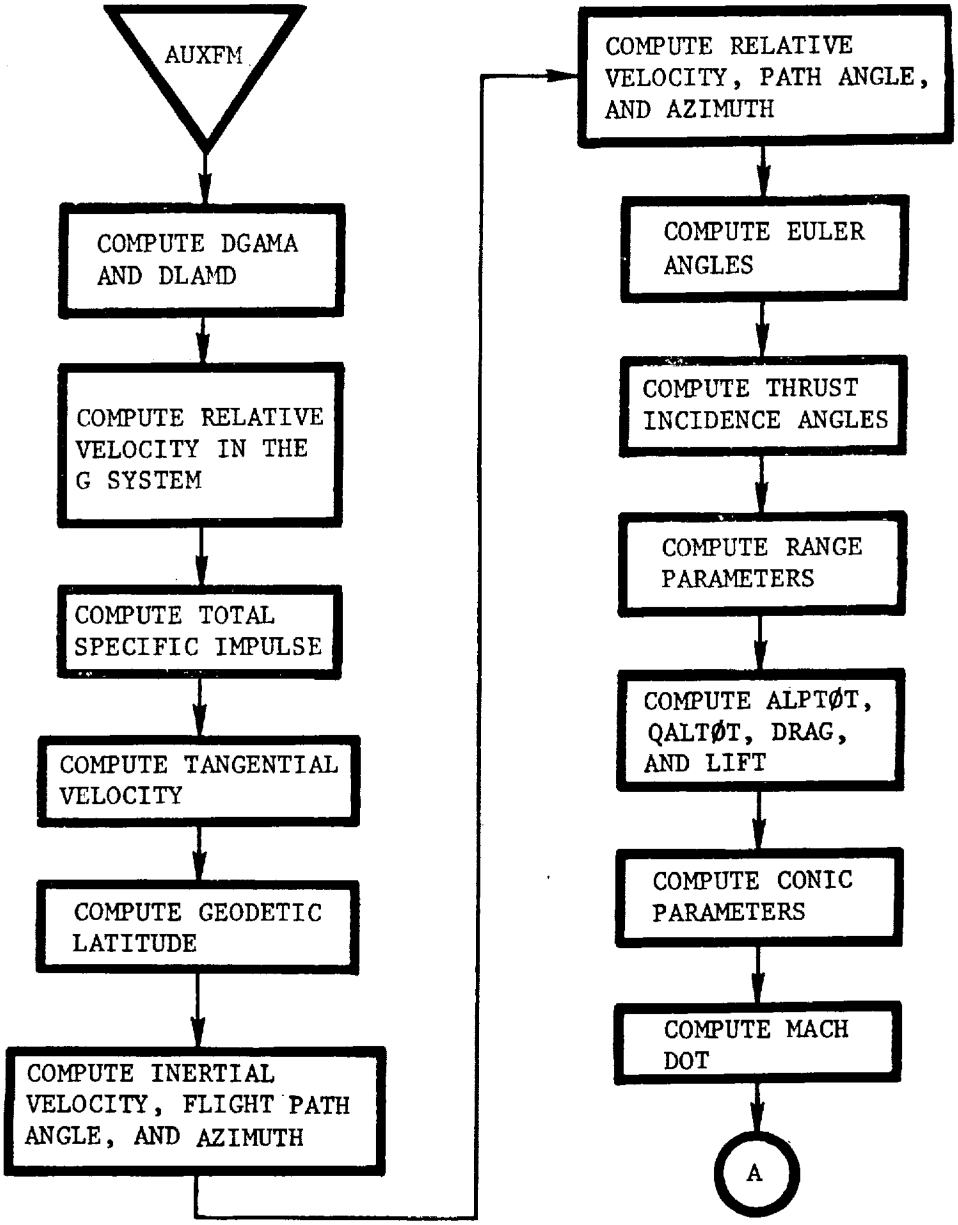


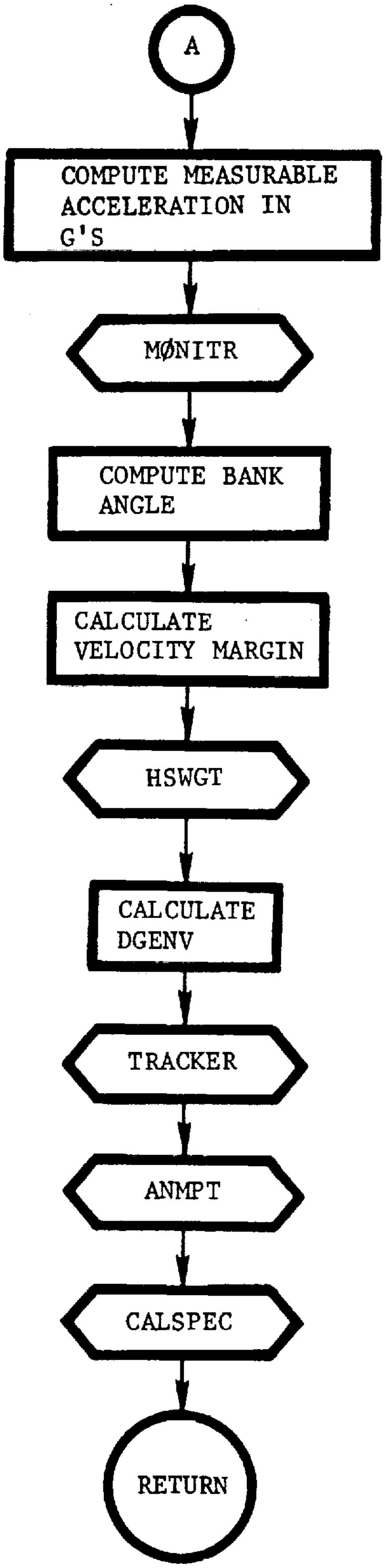

90
IF[RL(ISV)]
= 0
100
≠0
PRES = PB(ISV)*(TMB(ISV)/ATEM)**(1.8743295E-2/RL(ISV))
110
PRES = PB(ISV)*EXP(-1.8743295E-2*(HG-HB(ISV))/TMB(ISV))
120
RHØ = .58256430E-3*PRES/ATEM
CS = 49.0221568 = √ATEM
IF[IØFLAG ≤ 1]
YES
NO
ATEM = ATEM/CTEMP
PRES = PRES/CPRES
RHØ = RHØ/CDENS
CS = CS*CMPFT
125
RETURN

ATMØS3: This routine computes the atmospheric parameters based on the 1963 Patrick AFB atmosphere model.

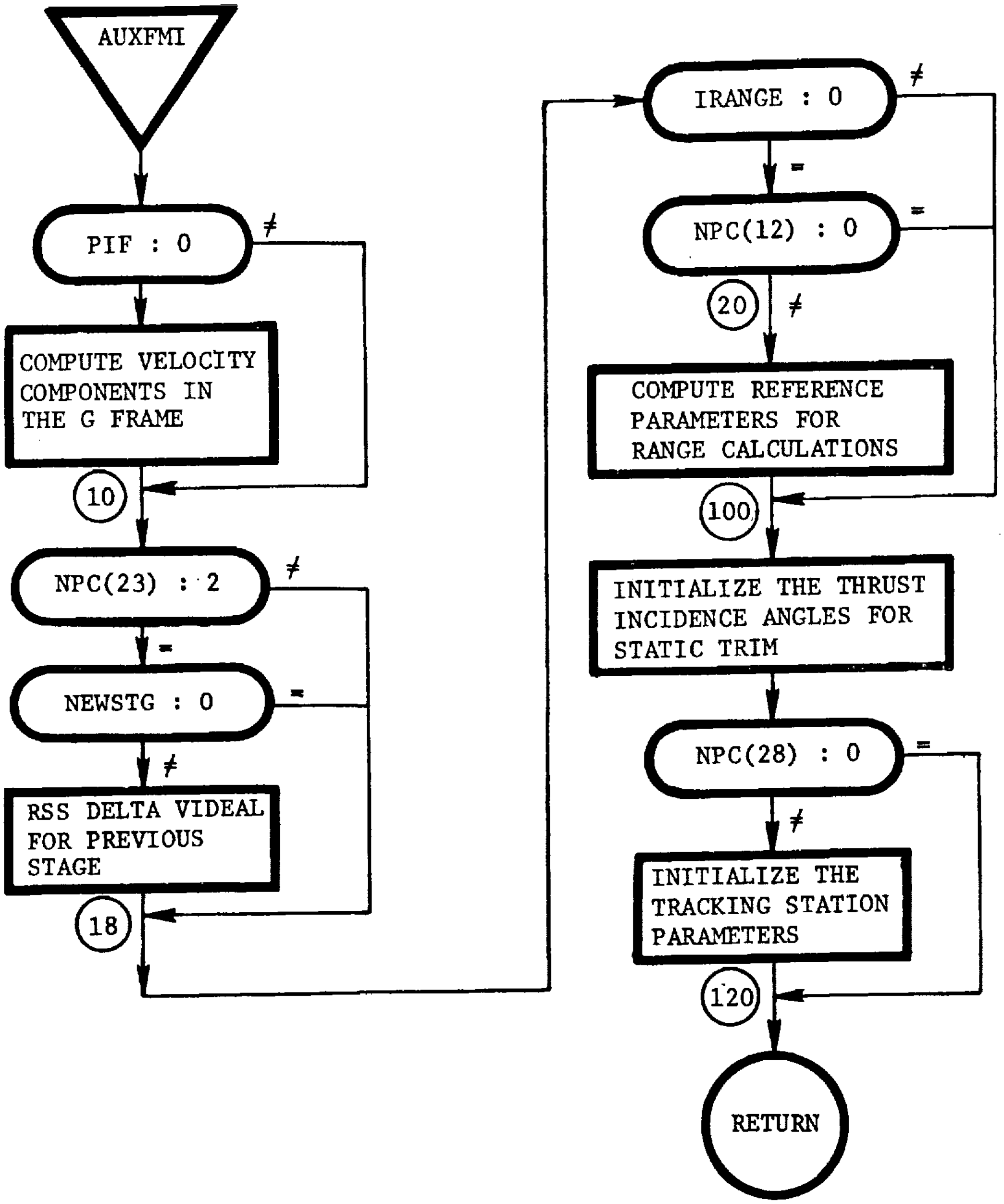

ATMØS3
CONVERT ALTITUDE TO METERS
IF ALTITUDE IS NEGATIVE, SET IT EQUAL TO ZERO
DETERMINE THE ALTITUDE REGION AND EVALUATE THE PROPER POLYNOMIALS
COMPUTE ATEM, PRES, AND DENS
CONVERT FROM METRIC TO ENGLISH UNITS IF REQUIRED
RETURN

AUXFM: This routine calculates the auxiliary variables at the end of each integration step.

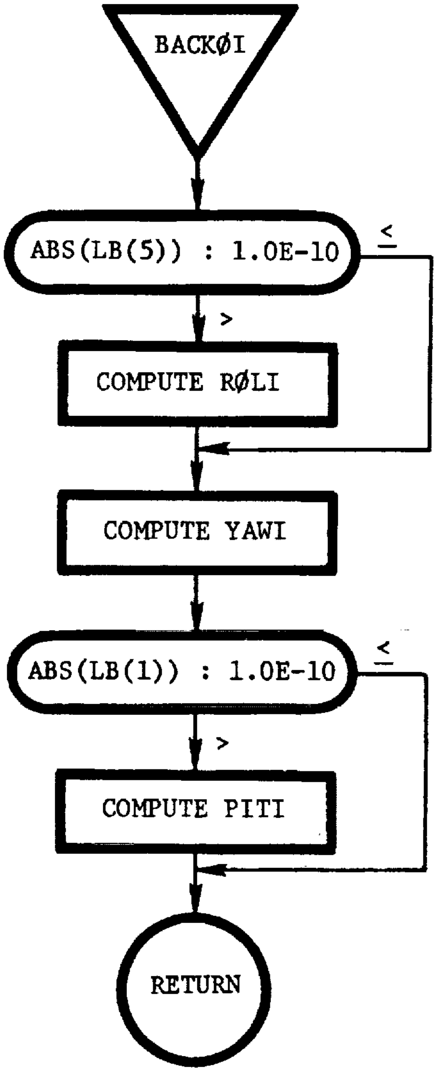

AUXFM
COMPUTE DGAMA AND DLAMD
COMPUTE RELATIVE VELOCITY IN THE G SYSTEM
COMPUTE TOTAL SPECIFIC IMPULSE
COMPUTE TANGENTIAL VELOCITY
COMPUTE GEODETIC LATITUDE
COMPUTE INERTIAL VELOCITY, FLIGHT PATH ANGLE, AND AZIMUTH
COMPUTE RELATIVE VELOCITY, PATH ANGLE, AND AZIMUTH
COMPUTE EULER ANGLES
COMPUTE THRUST INCIDENCE ANGLES
COMPUTE RANGE PARAMETERS
COMPUTE ALPTØT, QALTØT, DRAG, AND LIFT
COMPUTE CONIC PARAMETERS
COMPUTE MACH DOT
A


A
COMPUTE MEASURABLE ACCELERATION IN G'S
MØNITR
COMPUTE BANK ANGLE
CALCULATE VELOCITY MARGIN
HSWGT
CALCULATE DGENV
TRACKER
ANMPT
CALSPEC
RETURN

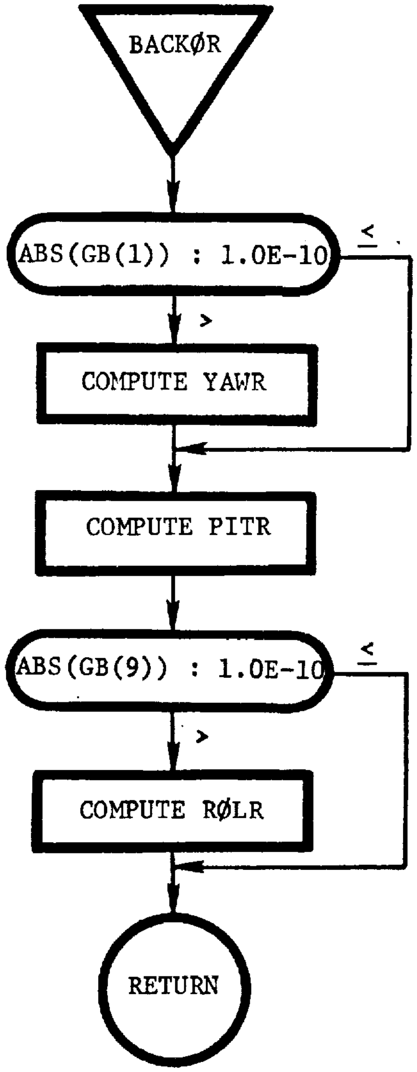
AUXFMI: This routine initializes the auxiliary calculations.


AUXFMI
PIF : 0
≠
COMPUTE VELOCITY COMPONENTS IN THE G FRAME
10
NPC(23) : 2
≠
=
NEWSTG : 0
=
≠
RSS DELTA VIDEAL FOR PREVIOUS STAGE
18
IRANGE : 0
≠
=
NPC(12) : 0
=
20
≠
COMPUTE REFERENCE PARAMETERS FOR RANGE CALCULATIONS
100
INITIALIZE THE THRUST INCIDENCE ANGLES FOR STATIC TRIM
NPC(28) : 0
=
≠
INITIALIZE THE TRACKING STATION PARAMETERS
120
RETURN

BACKØI: This routine computes the inertial Euler angles, given the LB matrix.



BACKØR: This routine computes the relative Euler angles, given the GB matrix.



BLKDAT: This routine is not called explicity but performs its function of presetting all internal program values at load time. Certain of these values can be overridden later by input if desired.

```
*DECK,BLKDAT
      BLOCK DATA
C*** BLKDAT
C              SET INITIAL PROGRAM VALUES
C*** THE DATA STATEMENTS IN THIS ROUTINE ARE STANDARDIZED
C
C
C     STANDARD FORTRAN DECLARATIONS
C              COMMON
C              DIMENSION
C              EQUIVALENCE
C              TYPE (CAN BE ANYWHERE ABOVE)
C              DATA (CAN BE ANYWHERE)
C              NAMELIST
C
C                           - W A R N I N G -
C               *** COMMON MULTRC MUST BE FIRST COMMON IN BLKDAT ***
C
*CALL MULTRC
C
*CALL SERVC
*CALL GENIC
*CALL SEARC
      COMMON/INFIV/ INFIV(1)
      COMMON/INFIC/ PF(396),HEADER(10)
      COMMON/INFND/ INFND
      COMMON /DYNIL/ DYNIL(184)
      COMMON/INPVC/ INPCF
*IF -IDENT,SMALLC,1
      COMMON /GENRL/ IGEN(2500)
*IF -IDENT,LARC,1
      COMMON IBKT(2500)
*IF IDENT,LARC,1
      COMMON IBKT(1500)
C
C
      DIMENSION DYNIL1(52)
      DIMENSION DYNIL2(48)
      DIMENSION DYNIL3(48)
      DIMENSION DYNIL4(18)
      DIMENSION DYNIL5(36)
      DIMENSION SEARC1(782)
      DIMENSION SEARC2(478)
      DIMENSION SEARC3(53)
      DIMENSION SEARC4(53)
      DIMENSION SEARC5(35)
      DIMENSION SEARC6(172)
      DIMENSION SEARC7(31)
      DIMENSION SEARC8(83)
C
      EQUIVALENCE (DYNIL1(1),DYNIL(  1))
      EQUIVALENCE (DYNIL2(1),DYNIL( 53))
      EQUIVALENCE (DYNIL3(1),DYNIL(101))
      EQUIVALENCE (DYNIL4(1),DYNIL(149))
      EQUIVALENCE (DYNIL5(1),DYNIL(167))
      EQUIVALENCE (SEARC1(1),DFDC(1)  )
      EQUIVALENCE (SEARC2(1),GP(1)    )
      EQUIVALENCE (SEARC3(1),ITC(1)   )
      EQUIVALENCE (SEARC4(1),NAC      )
      EQUIVALENCE (SEARC5(1),PG1(1)   )
      EQUIVALENCE (SEARC6(1),P2NOM    )
      EQUIVALENCE (SEARC7(1),YES(1)   )
      EQUIVALENCE (SEARC8(1),TIMIN    )
C
C     COMMON MULTRC
C
      DATA IN    /3    /
      DATA IO    /3    /
      DATA MULTRF/1    /
      DATA ICASE /0    /
      DATA IGSIZ /0      /
C
C     COMMON SERVC
C
      DATA TEMP  /50*0        /
      DATA STEMP /25*0.0      /
      DATA IR1   /0           /
      DATA I     /0           /
      DATA J     /0           /
      DATA K     /0           /
      DATA L     /0           /
      DATA M     /0           /
      DATA NULL  /1HU         /
      DATA N00   /0           /
      DATA N01   /1           /
      DATA N02   /2           /
      DATA N03   /3           /
      DATA N04   /4           /
      DATA N05   /5           /
      DATA N06   /6           /
      DATA N07   /7           /
      DATA N08   /8           /
      DATA N09   /9           /
      DATA N10   /10          /
      DATA N11   /11          /
      DATA N12   /12          /
      DATA N13   /13          /
      DATA N14   /14          /
      DATA N15   /15          /
      DATA FP00  /0.0         /
      DATA FPP5  / .5         /
      DATA FP1   /1.0         /
      DATA FP2   /2.0         /
      DATA FP3   /3.0         /
      DATA FP4   /4.0          /
      DATA FP5   /5.0          /
      DATA FP6   /6.0          /
      DATA FP7   /7.0          /
      DATA FP8   /8.0          /
      DATA FP9   /9.0          /
      DATA FP10  /10.0         /
      DATA FP11  /11.0         /
      DATA FP12  /12.0         /
      DATA FP13  /13.0         /
      DATA FP14  /14.0         /
      DATA FP15  /15.0         /
      DATA FP60  /60.0         /
      DATA FP90  /90.0         /
      DATA FP180 /180.0        /
      DATA FP270 /270.0        /
      DATA FP360 /360.0        /
      DATA PI02  /1.5707963267948965/
      DATA PI    /3.141592653589793/
      DATA RPD   /0.01745329251994329/
      DATA DPR   /57.29577951308232/
      DATA TWOPI /6.283185307179586/
      DATA FTPNM  /6076.1155     /
      DATA CMPFT  /.3048         /
      DATA IOFLAG /0/
C*** CFORCE = NEWTONS PER POUND
      DATA CFORCE /4.4482216152605/
C***  CPRES = LB/FT**2 PER NEWTONS/METERS**2
      DATA CPRES /.0208854347/
C***  CTEMP = DEGREES F PER DEGREES K
      DATA CTEMP  /1.8           /
C*** CDENS = SLUGS/FT**3 PER KILOGRAM/METER**3
      DATA CDENS /.00194031965 /
C***  CHEAT= JOULES PER BTU
      DATA CHEAT /1054.350264488888 /
C***  CMASS = KILOGRAMS PER SLUG
      DATA CMASS /14.5939029 /
      DATA CVDIST/6076.1155      /
      DATA IDENT /1.0   ,0.0   ,0.0   ,
     1            0.0   ,1.0   ,0.0   ,
     2            0.0   ,0.0   ,1.0   /
      DATA IVSZ  /0            /
      DATA XINF  /1.0E+11        /
C
      DATA INFIV /0            /
      DATA INPCF /0             /
C***
C
                            DATA PE
     1/0,6HTIME  ,0,6HTIMES ,0,6HTDURP ,0,6HDENS  ,0,6HPRES  ,0,6HATEM
     2,0,6HALTITU,0,6HGCRAD ,0,6HGDLAT ,0,6HGCLAT ,0,6HLONG  ,0,6HLONGI
     3,0,6HVELI  ,0,6HGAMMAI,0,6HAZVELI,0,6HXI    ,0,6HVXI   ,0,6HAXI
     4,0,6HVELR   ,0,6HGAMMAR,0,6HAZVELR,0,6HYI     ,0,6HVYI    ,0,6HAYI
     5,0,6HVELA   ,0,6HGAMMAA,0,6HAZVELA,0,6HZI     ,0,6HVZI    ,0,6HAZI
     6,0,6HGAMAD  ,0,6HAZVAD  ,0,6HDWNRNG,0,6HCRRNG ,0,6HDPRNG1,0,6HDPRNG2
     7,0,6HTHRUST,0,6HWEIGHT,0,6HWDOT   ,0,6HWEICON,0,6HWPROP  ,0,6HASMG
     8,0,6HETA    ,0,6HETAL   ,0,6HIPNULL,0,6HIYNULL,0,6HINCPCH,0,6HINCYAW
     9,0,6HFTXB   ,0,6HFAXB   ,0,6HAXB    ,0,6HALPHA ,0,6HALPDOT,0,6HALPTOT
     0,0,6HFTYB   ,0,6HFAYB   ,0,6HAYB    ,0,6HBETA  ,0,6HBETDOT,0,6HQALPHA
     A,0,6HFTZB   ,0,6HFAZB   ,0,6HAZB    ,0,6HBNKANG,0,6HBNKDOT,0,6HQALTOT
     B,0,6HCA     ,0,6HCD     ,0,6HDRAG   ,0,6HROLI  ,0,6HYAWR   ,0,6HROLBD
     C,0,6HCN     ,0,6HCL     ,0,6HLIFT   ,0,6HYAWI  ,0,6HPITR   ,0,6HPITBD
     D,0,6HCY     ,0,6HHEATRT,0,6HTLHEAT,0,6HPITI   ,0,6HROLR   ,0,6HYAWBD
     E,0,6HDYNP   ,0,6HMACH   ,0,6HREYNO ,0,6HASXI  ,0,6HASYI   ,0,6HASZI
     F,216*0
     G/
C
      DATA HEADER /10*1H /
C
C
C
C ..... INTEGRATION FLAG  0- INTEGRAL NOT ON
C                         1- INTEGRAL ON      FIRST ORDER
C                         2- INTEGRAL ON      SECOND ORDER
C          BOTH FIRST AND SECOND DERIVATIVES MUST BE INCLUDED IN LIST
C          FOR SECOND ORDER EQUATIONS.  SECOND ORDER EQUATIONS MUST
C          BE FLAGED WITH -2-  FOR KROGH INTEGRATOR
C
      DATA DYNIL1
     0 / 202
     1 ,6HTIME  ,6HDTIME ,1
     2 ,6HXI    ,6HVXI   ,1
     3 ,6HYI    ,6HVYI   ,1
     4 ,6HZI    ,6HVZI   ,1
     5 ,6HVXI   ,6HAXI   ,2
     6 ,6HVYI   ,6HAYI   ,2
     7 ,6HVZI   ,6HAZI   ,2
     8 ,6HMASS  ,6HDMASS ,1
     9 ,6HE0    ,6HDE0   ,0
     0 ,6HE1    ,6HDE1   ,0
     A ,6HE2    ,6HDE2   ,0
     B ,6HE3    ,6HDE3   ,0
     C ,6HFVAL1 ,6HDFVAL1,0
     D ,6HFVAL2 ,6HDFVAL2,0
     E ,6HFVAL3 ,6HDFVAL3,0
     F ,6HTLHEAT,6HHEATRT,1
     G ,6HHTRTI1,6HHTRT1 ,1
     H /
      DATA DYNIL2
     1/6HDLR    ,6HDLRD  ,0
     2,6HTVLR   ,6HTVLRD ,0
     3,6HATL    ,6HATLD  ,0
     4,6HGLR    ,6HGLRD  ,0
     5,6HVIDEAL,6HVIDLD  ,0
```

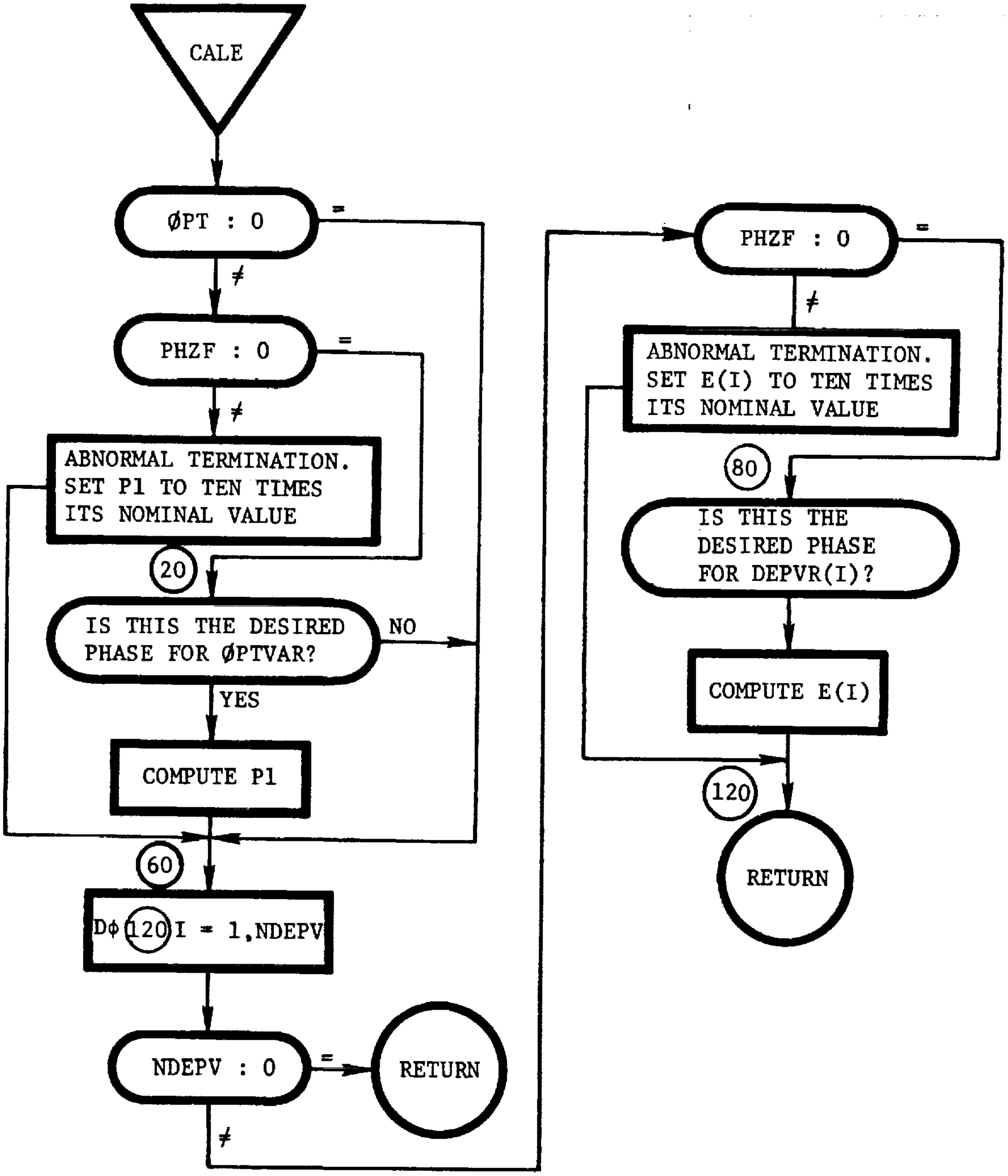
```
6,6HPWPROP,6HPWDOT ,0
7,6HAHI    ,6HAHID  ,0
8,6HHTBT   ,6HHTBTD ,0
9,6HHTTP   ,6HHTTPD ,0
0,6HHTRT   ,6HHTRTD ,0
A,6HHTLF   ,6HHTLFD ,0
B,6HHTURB  ,6HHTURBD,0
C,6HTIMRF1,6HDTIMR1,0
D,6HTIMRF2,6HDTIMR2,0
E,6HTIMRF3,6HDTIMR3,0
F,6HTIMRF4,6HDTIMR4,0
G/
 DATA DYNIL3
1 /6HDX      ,6HDVX    ,0
2 ,6HDY      ,6HDVY    ,0
3 ,6HDZ      ,6HDVZ    ,0
4 ,6HDVX     ,6HDAX    ,0
5 ,6HDVY     ,6HDAY    ,0
6 ,6HDVZ     ,6HDAZ    ,0
7,6HGINT1  ,6HGDERV1,0
8,6HGINT2  ,6HGDERV2,0
9,6HGINT3  ,6HGDERV3,0
0,6HGINT4  ,6HGDERV4,0
A,6HGINT5  ,6HGDERV5,0
B,6HGINT6  ,6HGDERV6,0
C,6HGINT7  ,6HGDERV7,0
D,6HGINT8  ,6HGDERV8,0
E,6HGINT9  ,6HGDERV9,0
F,6HGINT10,6HGDERV0,0
G /
 DATA DYNIL4
1 /6HDIAMP1,6HDIARP1,0
2 ,6HDIAMP2,6HDIARP2,0
3 ,6HDIAMP3,6HDIARP3,0
4 ,6HDLI    ,6HDLID   ,0
5 ,6HTVLI   ,6HTVLID  ,0
6 ,6HGLI    ,6HGLID   ,0
I /
 DATA DYNIL5
1 /6HDXI    ,6HDVXI   ,0
2 ,6HDYI    ,6HDVYI   ,0
3 ,6HDZI    ,6HDVZI   ,0
4 ,6HDVXI   ,6HDAXI   ,0
5 ,6HDVYI   ,6HDAYI   ,0
6 ,6HDVZI   ,6HDAZI   ,0
7 ,6HDXPT   ,6HDVXPT  ,0
8 ,6HDYPT   ,6HDVYPT  ,0
9 ,6HDZPT   ,6HDVZPT  ,0
0 ,6HDVXPT  ,6HDAXPT  ,0
A ,6HDVYPT  ,6HDAYPT  ,0
B ,6HDVZPT  ,6HDAZPT  ,0
C /
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C     COMMON SEARC
C
      DATA ACOB  /625*0              /
      DATA CONEPS/     89.9   ,
     1                 5*.1   /
      DATA CONSEX/  .000001
     1           ,     .001   /
      DATA CTHA  /       .5   /
      DATA CTHAT /0                /
      DATA DEPPH / 25*900.0   /
      DATA DEPTL /   25*1.0   /
      DATA DEPVAL/   25*0.0   /
      DATA DEPVR /   25*0.0   /
      DATA SEARC1/782*0          /
      DATA FITEPR/  .000001
     1           ,     .001   /
      DATA GAMAS /0                /
      DATA GAMASS/0                /
      DATA GAMAX /10/
      DATA SEARC2/478*0          /
      DATA ICGM  /63               /
      DATA IDAV  /0                /
      DATA IDEB  /        0   /
      DATA IDEPVR/     25*0   /
      DATA IFDEG /     25*0   /
      DATA IHADIT/0                /
      DATA IMAX  /0                /
      DATA IMIN  /0                /
      DATA INDPH /     25*0   /
      DATA INDVR /     25*0   /
      DATA INTRPL/0                /
      DATA INTRY1/0                /
      DATA IOPT  /0                /
      DATA IPRO  /        0   /
      DATA ISTART/63               /
      DATA SEARC3/53*0        /
      DATA LIMIT /5000B    /
      DATA MAXITR/       10   /
      DATA MODEW /        1   /
      DATA SEARC4/53*0           /
      DATA NDEPV /        0   /
      DATA NEQC  /0                /
      DATA NETF  /0                /
      DATA NFLAG /0                /
      DATA NINDV /        0   /
      DATA NOMF  /0                /
      DATA NSTEP /0                /
      DATA NTC   /0                /
      DATA OLDG2 /0                /
      DATA OLDP1 /0                /
      DATA OLDP2 /0                /
      DATA OLDU  /0            /
      DATA OPT   /        0   /
      DATA OPTPH /900.        /
      DATA OPTVAR/        0   /
      DATA PCTCC /       .3   /
      DATA PCTOLD/       .3   /
      DATA PERT  /25*1.E-4      /
      DATA PGEPS /     1.0   /
      DATA SEARC5/35*0          /
      DATA P2MIN /      1.0   /
      DATA SEARC6/172*0         /
      DATA SRCHM /        0   /
      DATA STEP  /6*0           /
      DATA STMINP/       .1
     1           ,       .1   /
      DATA STPMAX/  1.0E+10   /
      DATA TGRAD /0             /
      DATA TTRIAL/0             /
      DATA U      /  25*0.0   /
      DATA UMAG  /0             /
      DATA WCON   /   100.0   /
      DATA WOPT   /     1.0   /
      DATA WU     /  25*1.0   /
      DATA SEARC7/31*0          /
      DATA ISFLG /63            /
      DATA IWTFLG/0             /
      DATA NPAD   /10,5,10 /
      DATA SEARC8 /83*0/
C
C     COMMON GENIC
C
      DATA IPRT  /63            /
      DATA LISTIN/2             /
C
      END
```

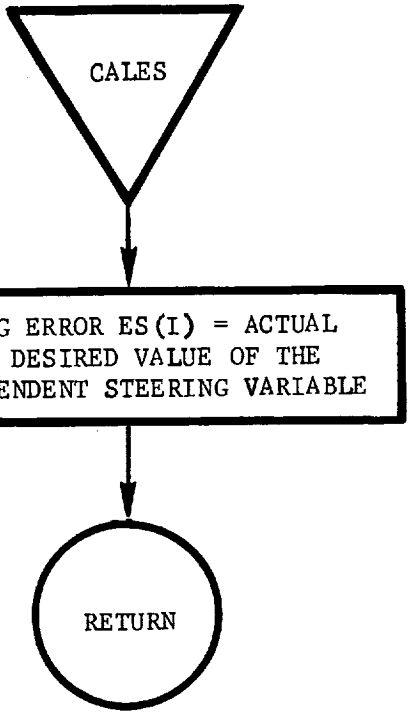
ORIGINAL PAGE IS
OF POOR QUALITY

CALE: This routine calculates the performance index (P1) and the error in the target variables (E(I)).

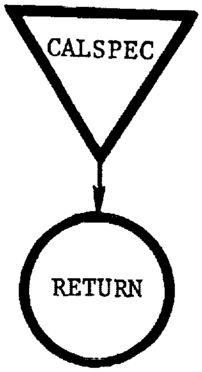


CALE
ØPT : 0
=
≠
PHZF : 0
=
≠
ABNORMAL TERMINATION. SET P1 TO TEN TIMES ITS NOMINAL VALUE
20
IS THIS THE DESIRED PHASE FOR ØPTVAR?
NO
YES
COMPUTE P1
60
Dϕ 120 I = 1,NDEPV
NDEPV : 0
=
RETURN
≠
PHZF : 0
=
≠
ABNORMAL TERMINATION. SET E(I) TO TEN TIMES ITS NOMINAL VALUE
80
IS THIS THE DESIRED PHASE FOR DEPVR(I)?
COMPUTE E(I)
120
RETURN

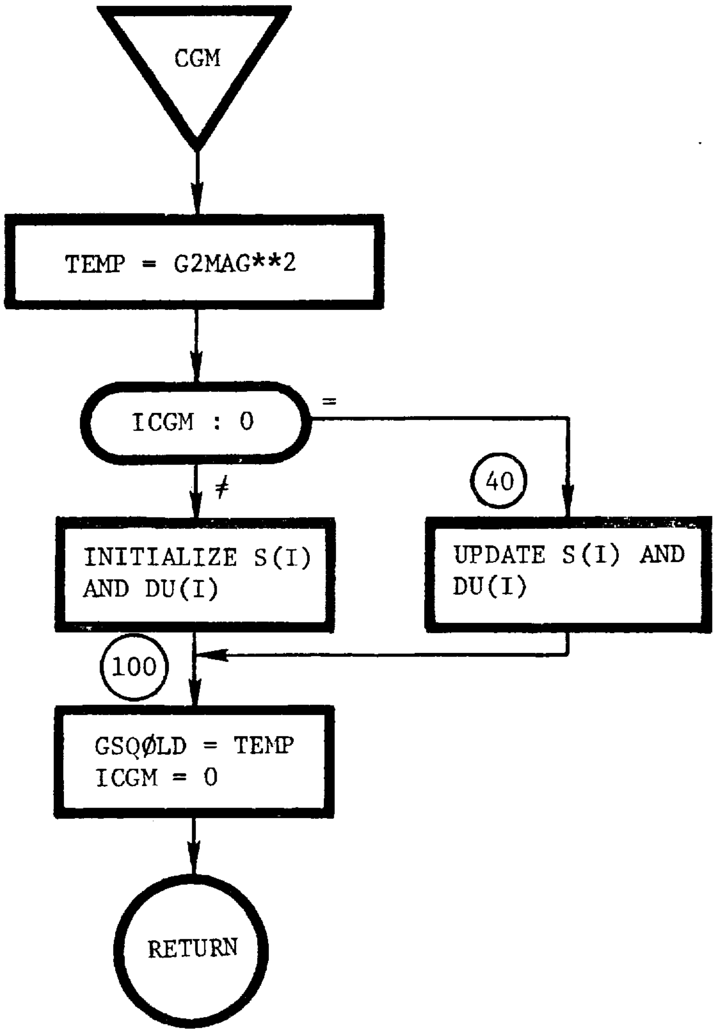
CALES: This rountine calculates the steering errors.


CALES
STEERING ERROR ES(I) = ACTUAL VALUE - DESIRED VALUE OF THE Ith DEPENDENT STEERING VARIABLE
RETURN

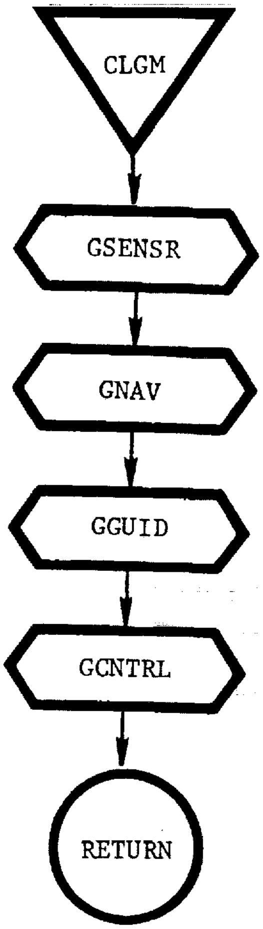
CALSPEC: This routine is a blank routine to be used when special calculations of a temporary nature are required. This routine is called at the end of each integration step from AUXFM.



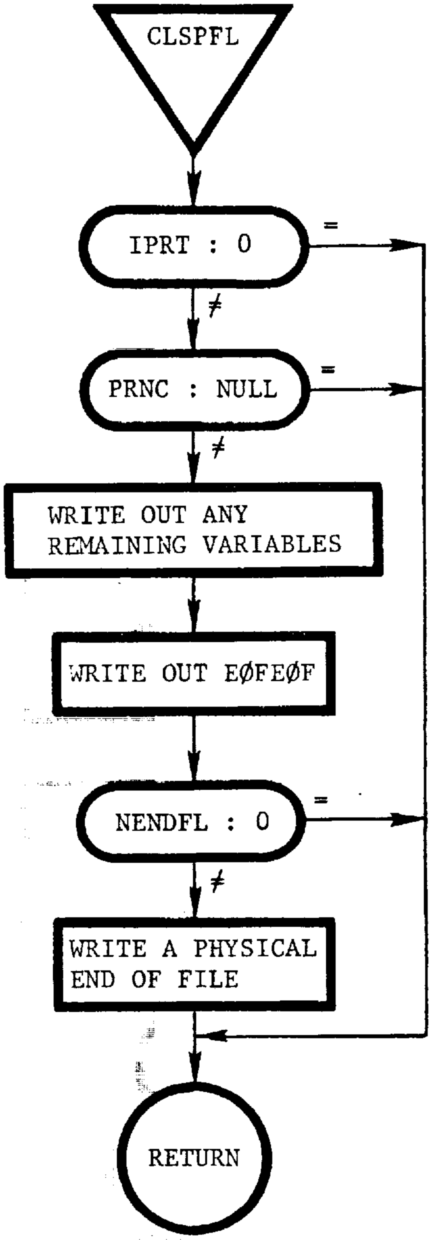
CGM: This routine contains the logic for the conjugate gradient method. It is a second-generation unconstrained optimization technique that has the stability of the steepest-descent method and the convergence properties of the second-order techniques.



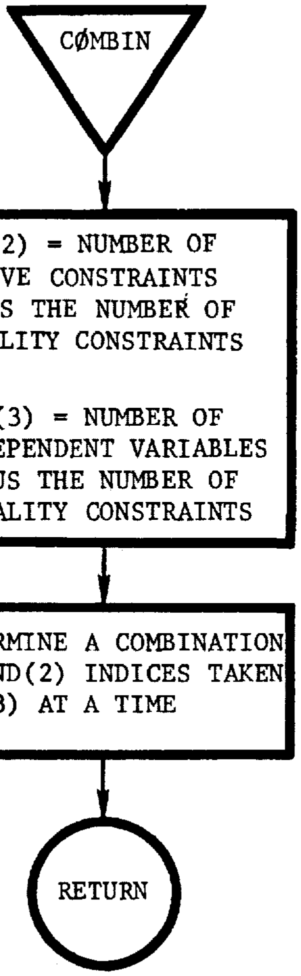
CLGM: This routine contains the executive logic for the closed-loop guidance routines.



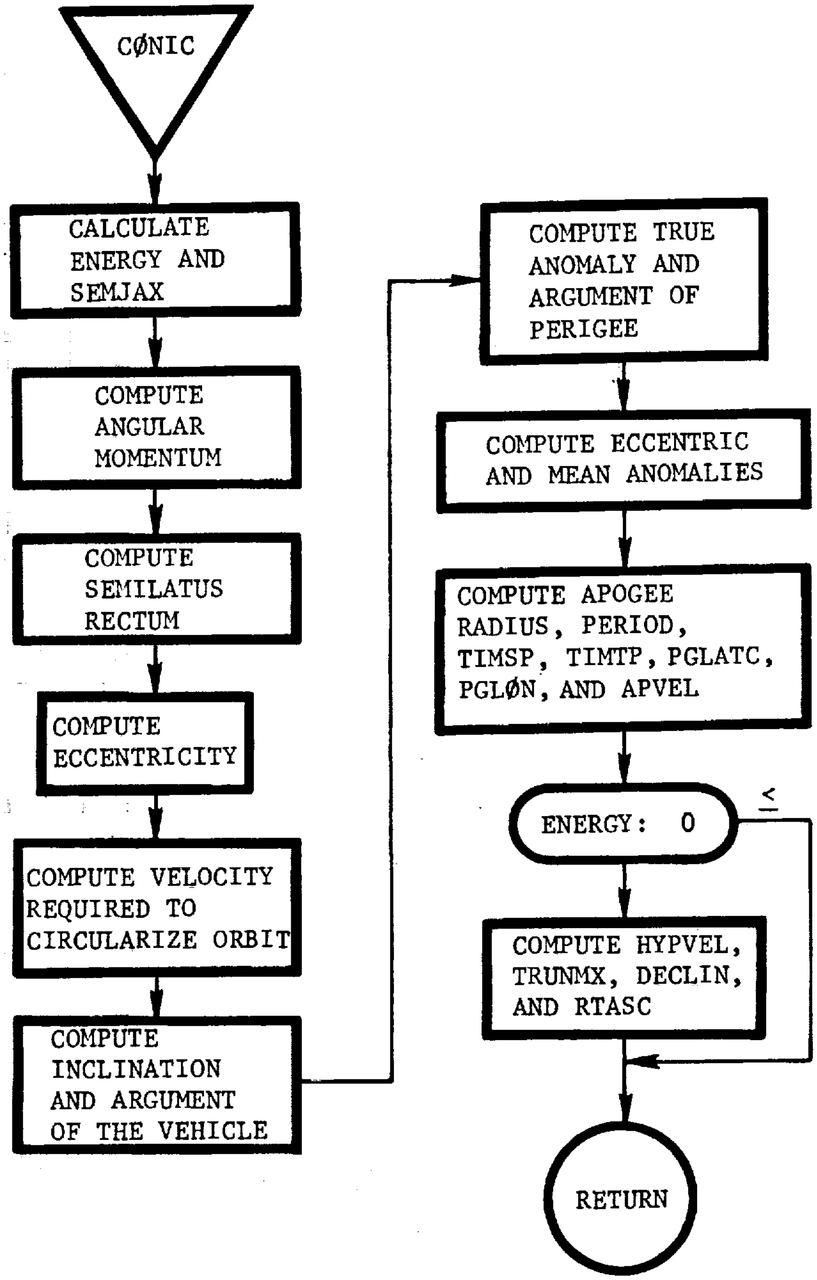
CLSPFL (NENDFL): This routine closes out profile records and writes a physical end of file on the profile tape for each trajectory.



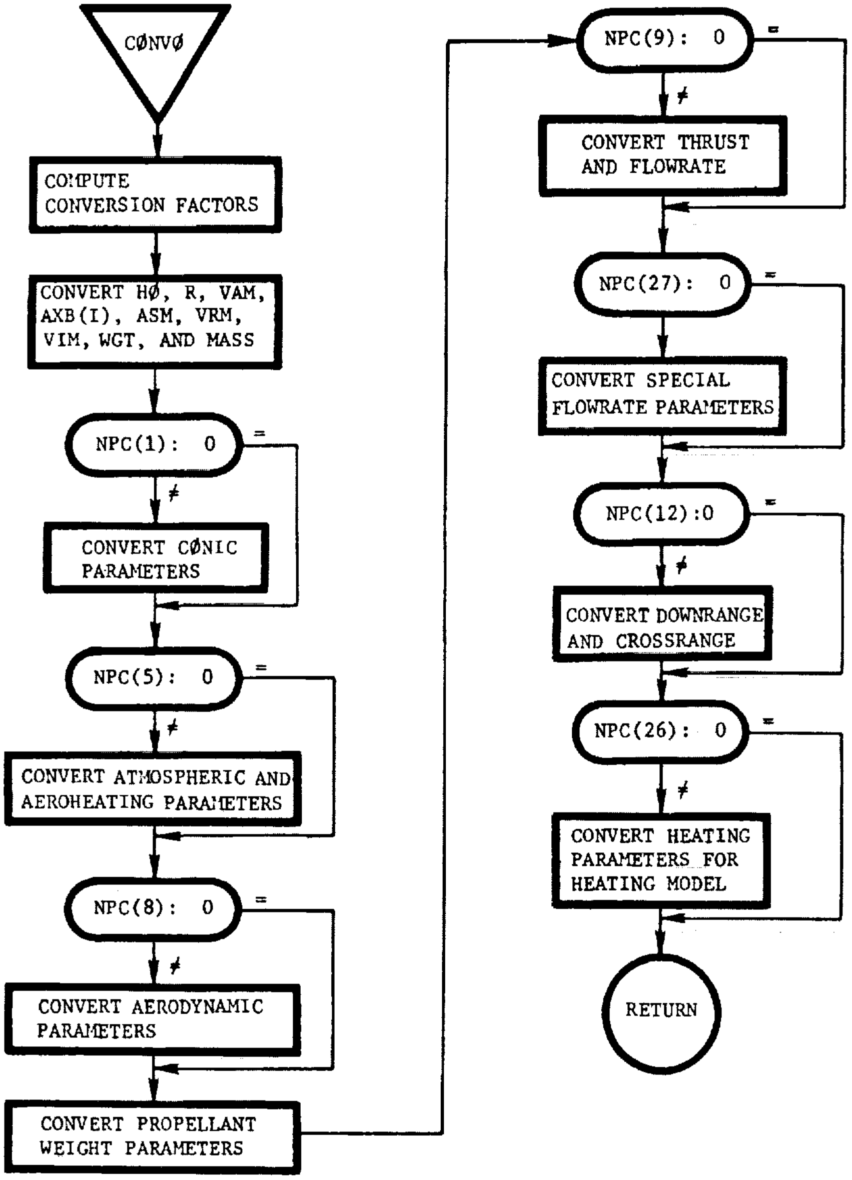
CØMBIN: This routine determines all combinations of the indices of the active constraints. These combinations of constraints are used to determine if any constraints can be dropped, and are used when the number of tight constraints exceeds the number of independent variables.


CØMBIN
IND(2) = NUMBER OF ACTIVE CONSTRAINTS MINUS THE NUMBER OF EQUALITY CONSTRAINTS
IND(3) = NUMBER OF INDEPENDENT VARIABLES MINUS THE NUMBER OF EQUALITY CONSTRAINTS
DETERMINE A COMBINATION OF IND(2) INDICES TAKEN IND(3) AT A TIME
RETURN

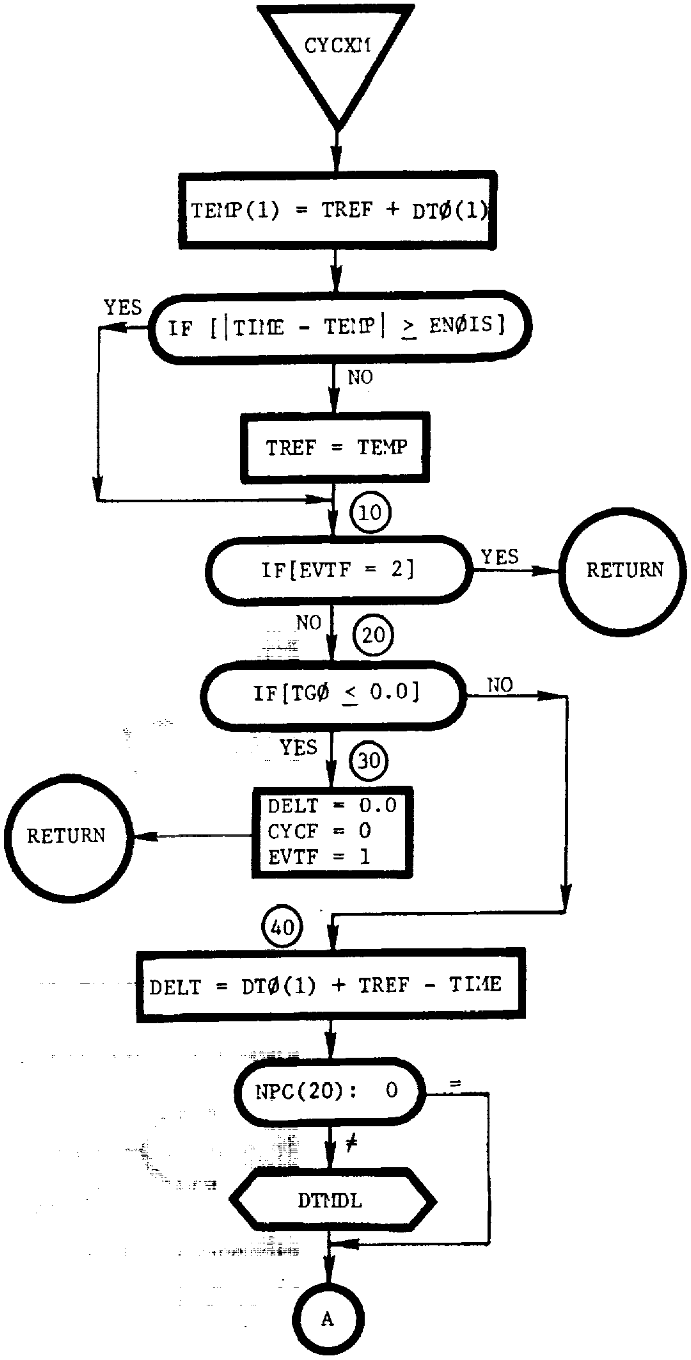
CØNIC: This routine calculates the Keplerian conic for either elliptic or hyperbolic orbits, based on the value of the orbital energy.

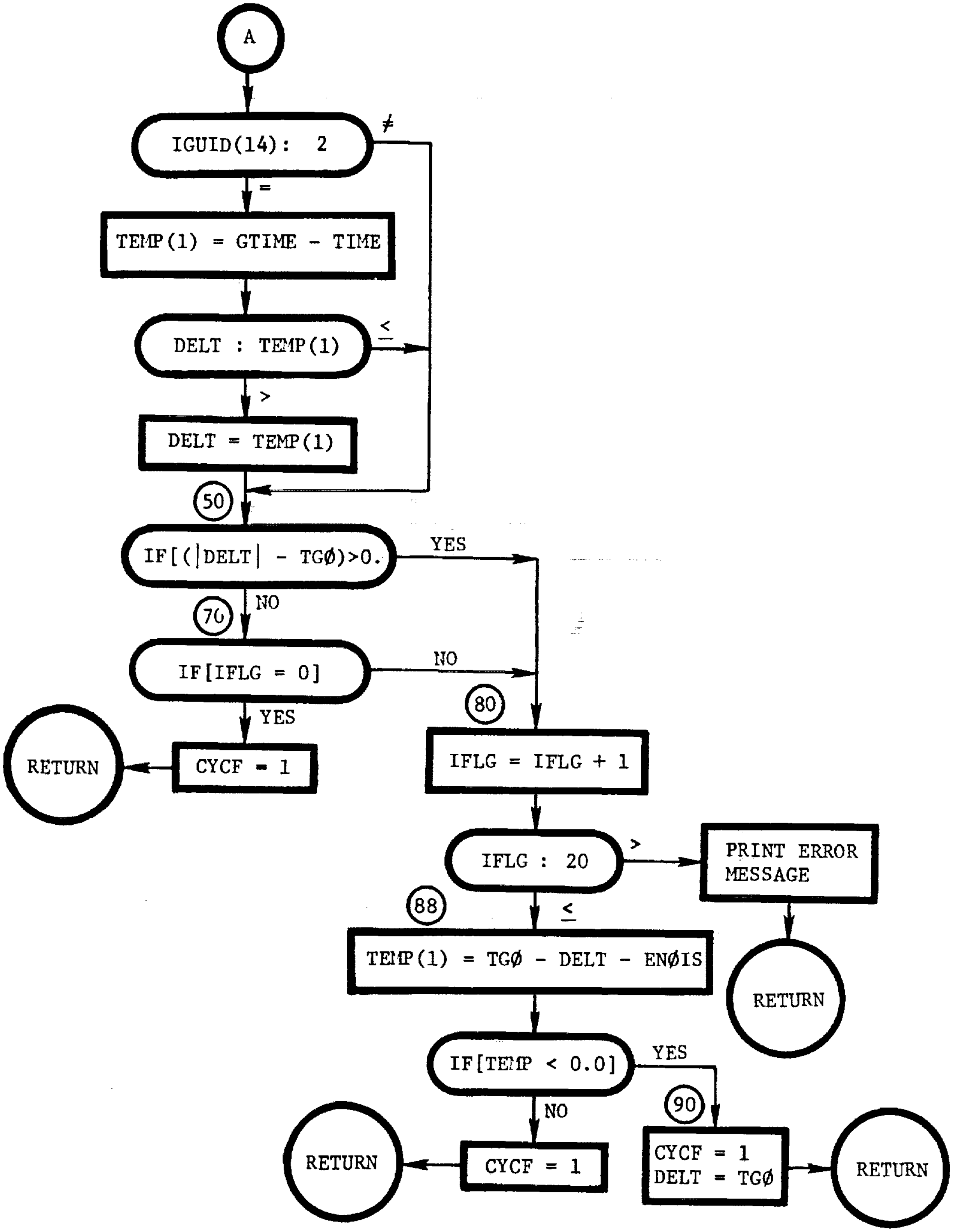

CØNIC
CALCULATE ENERGY AND SEMJAX
COMPUTE ANGULAR MOMENTUM
COMPUTE SEMILATUS RECTUM
COMPUTE ECCENTRICITY
COMPUTE VELOCITY REQUIRED TO CIRCULARIZE ORBIT
COMPUTE INCLINATION AND ARGUMENT OF THE VEHICLE
COMPUTE TRUE ANOMALY AND ARGUMENT OF PERIGEE
COMPUTE ECCENTRIC AND MEAN ANOMALIES
COMPUTE APOGEE RADIUS, PERIOD, TIMSP, TIMTP, PGLATC, PGLØN, AND APVEL
ENERGY: 0
≤
COMPUTE HYPVEL, TRUNMX, DECLIN, AND RTASC
RETURN

CØNVØ: This routine converts the output variables from metric to English, or vice-versa, as required.

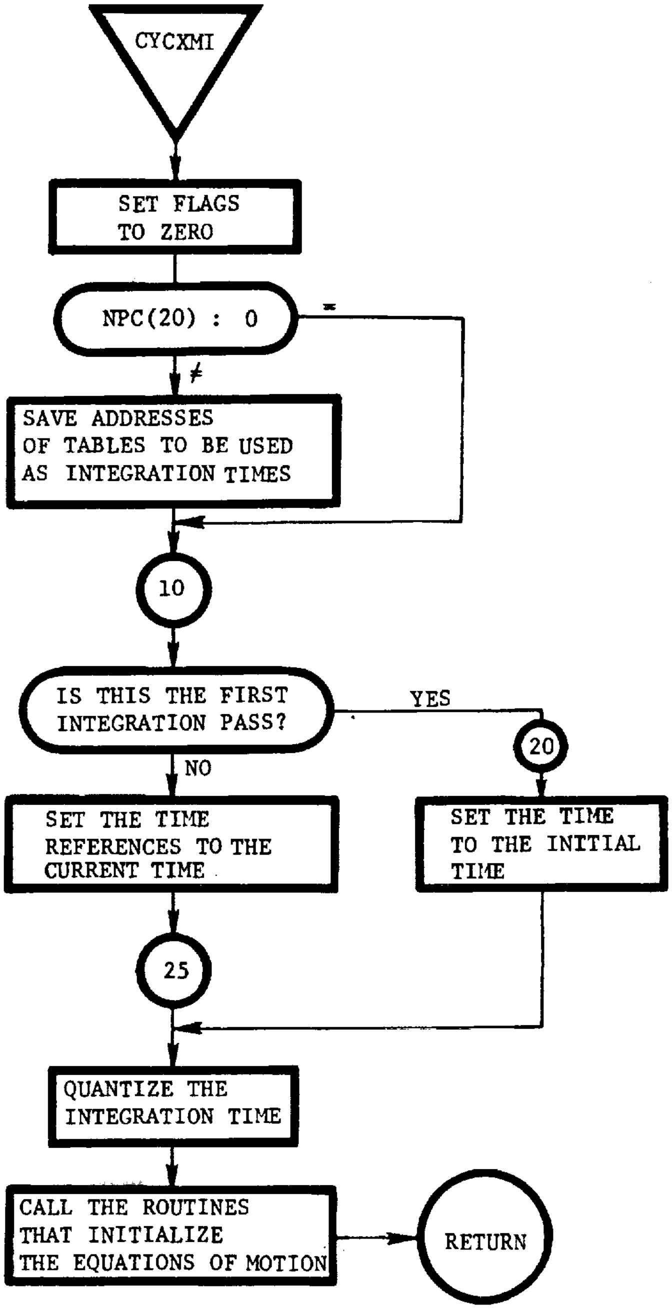

CØNVØ
COMPUTE CONVERSION FACTORS
CONVERT HØ, R, VAM, AXB(I), ASM, VRM, VIM, WGT, AND MASS
NPC(1): 0
=
≠
CONVERT CØNIC PARAMETERS
NPC(5): 0
=
≠
CONVERT ATMOSPHERIC AND AEROHEATING PARAMETERS
NPC(8): 0
=
≠
CONVERT AERODYNAMIC PARAMETERS
CONVERT PROPELLANT WEIGHT PARAMETERS
NPC(9): 0
=
≠
CONVERT THRUST AND FLOWRATE
NPC(27): 0
=
CONVERT SPECIAL FLOWRATE PARAMETERS
NPC(12):0
=
≠
CONVERT DOWNRANGE AND CROSSRANGE
NPC(26): 0
=
≠
CONVERT HEATING PARAMETERS FOR HEATING MODEL
RETURN

CYCXM: This routine performs program-cycling functions, which includes checking for any new phase.




A
IGUID(14): 2
≠
=
TEMP(1) = GTIME - TIME
DELT : TEMP(1)
≤
>
DELT = TEMP(1)
50
IF[(|DELT| - TGØ)>0.
YES
70
NO
IF[IFLG = 0]
NO
YES
RETURN
CYCF = 1
80
IFLG = IFLG + 1
IFLG : 20
>
PRINT ERROR MESSAGE
88
≤
TEMP(1) = TGØ - DELT - ENØIS
RETURN
IF[TEMP < 0.0]
YES
NO
90
RETURN
CYCF = 1
CYCF = 1
DELT = TGØ
RETURN

CYCXMI: This routine initializes the program at the beginning of each new cycle.


CYCXMI
SET FLAGS TO ZERO
NPC(20) : 0
=
≠
SAVE ADDRESSES OF TABLES TO BE USED AS INTEGRATION TIMES
10
IS THIS THE FIRST INTEGRATION PASS?
YES
NO
20
SET THE TIME REFERENCES TO THE CURRENT TIME
SET THE TIME TO THE INITIAL TIME
25
QUANTIZE THE INTEGRATION TIME
CALL THE ROUTINES THAT INITIALIZE THE EQUATIONS OF MOTION
RETURN

DATA: This routine is not called explicitly, but presets the program variables to their stored values at the time overlay (2,0) is called.

```
*DECK,DATA
      SUBROUTINE DATA
C*** DATA
C             DATA - DEFINES COMPUTATIONAL COMMONS
C                    (IV - END) INITIAL DATA VALUES
C*** THE DATA STATEMENTS IN THIS ROUTINE ARE STANDARDIZED
C
C
      COMMON/IV/ IV(2)
C
*CALL AUXVC
*CALL CYCVC
*CALL DPGVC
*CALL DYNVC
*CALL INFVC
*CALL MNMMLT
*CALL HOLINC
*CALL MOTBL
*CALL MOTIC
*CALL MOTVC
*CALL PHZVC
*CALL SPECAL
*CALL TGOVC
*CALL TRACKC
*CALL GUIDIC
*CALL GUIDVC
*CALL TARGVC
C
      COMMON/END/ END
C
C
C
*CALL DYTEM
*CALL LOCAL
C
*CALL KROIC
C
*CALL KROVC
C
                             COMMON/PRO/
     1  PRO(502)
C
      COMMON /GFLAGS/ IGFLAG(10)
      COMMON /GVARS / GVARS(100)
C
      DIMENSION AUXVC1(82)
      DIMENSION AUXVC2(40)
      DIMENSION AUXVC3(23)
      DIMENSION AUXVC4(11)
      DIMENSION DPGVC1(33)
      DIMENSION DPGVC2( 9)
      DIMENSION DPGVC3(77)
      DIMENSION DPGVC4(27)
      DIMENSION DYNVC1(14)
      DIMENSION HOLIN1(51)
      DIMENSION MNMML1(16)
      DIMENSION MOTBL1(304)
      DIMENSION MOTVC1(141)
      DIMENSION MOTVC2(  9)
      DIMENSION MOTVC3(148)
      DIMENSION MOTVC4(26)
      DIMENSION SPECA1(24)
      DIMENSION GUIDV1(35)
      DIMENSION TARGV1(133)
      DIMENSION LOCAL1(145)
      DIMENSION KROIC1(13)
      DIMENSION KROVC1(27)
      DIMENSION KROVC2(8)
      DIMENSION KROVC3(59)
C
      EQUIVALENCE (AUXVC1(1),ALPTOT     )
      EQUIVALENCE (AUXVC2(1),CRRNG      )
      EQUIVALENCE (AUXVC3(1),URX(1)     )
      EQUIVALENCE (AUXVC4(1),DPRGI1     )
      EQUIVALENCE (DPGVC1(1),ALPHA      )
      EQUIVALENCE (DPGVC2(1),HARG(1)    )
      EQUIVALENCE (DPGVC3(1),YAWI       )
      EQUIVALENCE (DPGVC4(1),ENOMS(1)   )
      EQUIVALENCE (DYNVC1(1),DTIMR(1)   )
      EQUIVALENCE (HOLIN1(1),ALPARG     )
      EQUIVALENCE (MNMML1(1),CADPNM     )
      EQUIVALENCE (MOTBL1(1),CST(1)     )
      EQUIVALENCE (MOTVC1(1),AHI        )
      EQUIVALENCE (MOTVC2(1),LONGI      )
      EQUIVALENCE (MOTVC3(1),DENS       )
      EQUIVALENCE (MOTVC4(1),DLID)
      EQUIVALENCE (SPECA1(1),SPECI(1)   )
      EQUIVALENCE (GUIDV1(1),GPXI(1)    )
      EQUIVALENCE (TARGV1(1),ALTAT      )
      EQUIVALENCE (LOCAL1(1),A(1)       )
      EQUIVALENCE (KROIC1(1),TINT     )
      EQUIVALENCE (KROVC1(1),DDINT(1))
      EQUIVALENCE (KROVC2(1),KQQ)
      EQUIVALENCE (KROVC3(1),RNDC)
C
C
      DATA IV(2)   /0     /
      DATA PRO     /502*0/
      DATA IGFLAG /10*0/
      DATA GVARS  /100*0/
C
C
C
C     AUXVC
```

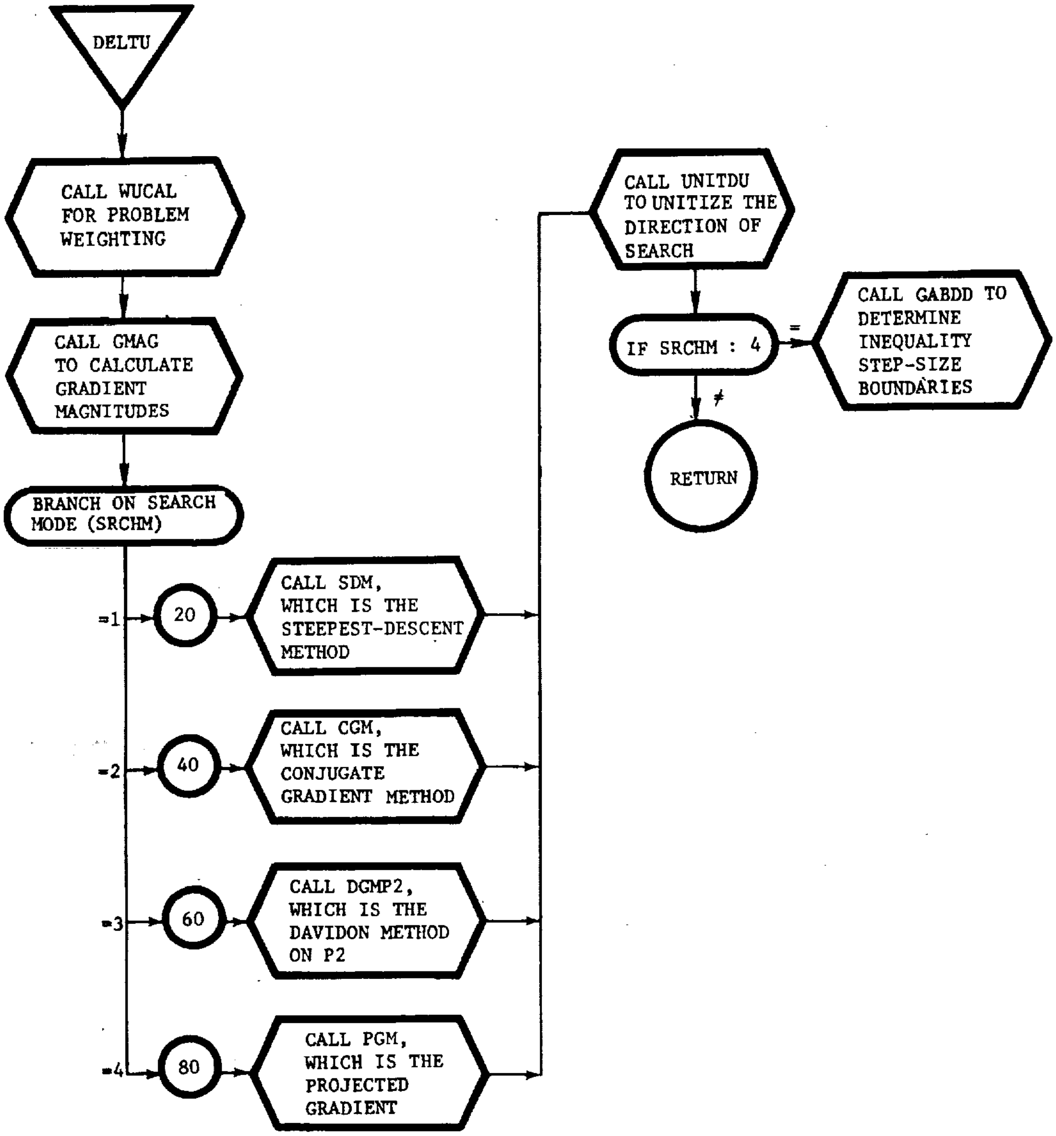
```
C
      DATA AUXVC1/82*0          /
      DATA XMAX  /10*-1.0E10/
      DATA XMIN  /10* 1.0E10/
      DATA AUXVC2 /40*0/
      DATA AUXVC3 /23*0/
      DATA NTRK   /1    /
      DATA XIVE /0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0/
      DATA AUXVC4 /11*0/
C
C     CYCVC
C
      DATA DELT  /0.0           /
      DATA DT    /1.0    /
      DATA DTIME /1.0    /
      DATA DTM   /1.0           /
      DATA DTO   /0.0           /
      DATA ENOIS /1.E-8 /
      DATA TREF  /0.0           /
      DATA IDTAB /6*0           /
      DATA IFLG  /0             /
      DATA CYCF  /0             /
      DATA DELTT /0         /
C
C     DPGVC
C
      DATA DPGVC1/33*0          /
      DATA AB     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA GB     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA IA     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA IB     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA IG     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA IL     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA LB     /1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
      DATA DPGVC2/9*0            /
                             DATA IGUID
     1/0    ,0    ,0    ,0    ,1    ,0    ,0    ,0    ,0    ,0
     2,0    ,2    ,1    ,0    ,0
     3,10*0
     4/
      DATA IVCRT /3*0            /
      DATA IVETA /0              /
      DATA KDG   /3*1.0         /
      DATA KRG   /3*1.0         /
      DATA DPGVC3 /77*0/
      DATA DEPTLS /4*1.0/
      DATA MAXITS /0/
      DATA NITS   /0/
      DATA PERTS  /4*1.E-4/
      DATA DPGVC4 /27*0/
C
C     DYNVC
```

```
C
      DATA DYNVC1/14*0.0        /
      DATA DEBINT / 0     /
      DATA DLTMIN /0/
      DATA DLTMAX /1.0E10/
      DATA EPSINT /1.0/
C
C     HOLINC
C
      DATA HOLIN1 /51*0/
C
C     INFVC
C
      DATA ESNPRT/0             /
      DATA EXTRAP,LPRNT/2*0/
      DATA PINC  /0.0           /
      DATA PRNC  /1HU           /
      DATA FID   /10HUD265 FILE,10H I.D. 0000/
      DATA INFF,IPRNTB,IPRNTR/3*0/
      DATA PSTOP /6HPSTOP /
      DATA TITLE /10*10H          /
      DATA SFID  /0   ,0   /
      DATA MPINC  /0     /
C
C     MNMMLT
C
      DATA ONE    /1.0/
      DATA MNMML1/16*0.0        /
C
C     MOTPL
C
                             DATA MOTPL1
     1/0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     2,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     3,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     4,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     5,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     6,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     7,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     8,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     9,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     0,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     A,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     B,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     C,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     D,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     E,0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1., 0,1.
     F,0,1., 0,1.
     I/
C
C     MOTIC
C
```

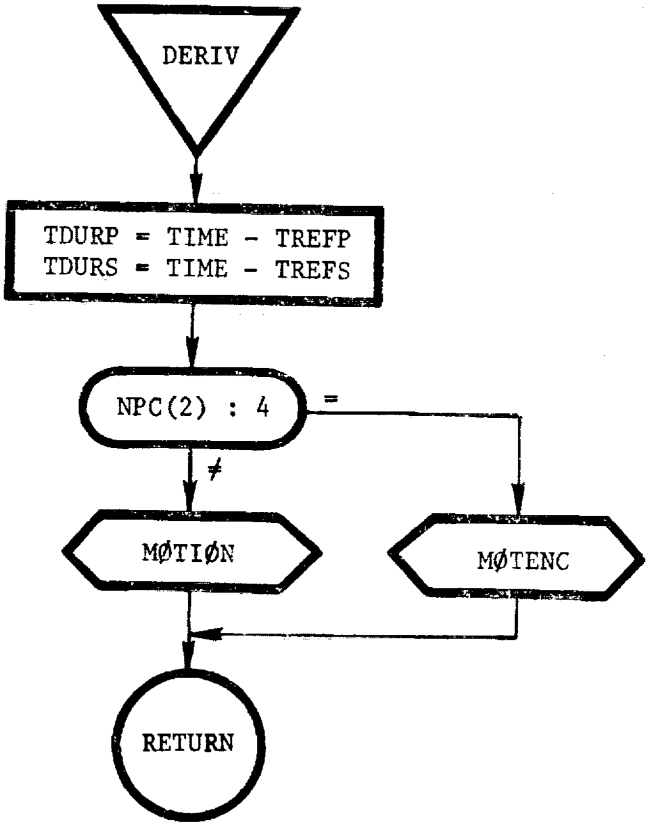
```
      DATA ALTIP /0.0           /
      DATA ALTREF/100.          /
      DATA LATREF/1HU           /
      DATA LONREF/0.0           /
      DATA AZREF /0.0            /
      DATA TIMREF/0.0           /
      DATA ARP    /10*0.0       /
      DATA ASMAX  /3.0           /
      DATA ATMOSK /1.0   ,1.0    /
      DATA AZWB   /180.          /
      DATA AZL    /0.0          /
      DATA LATL   /1HU          /
      DATA LONL   /1HU          /
      DATA CLCDMX/0.0           /
      DATA DETA   /0.0          /
      DATA DESN   /3*0.0        /
      DATA DESNE /0.0           /
      DATA DVIMAG,DVMAR,DVPCT/3*0.0/
      DATA ETAPC /1.0,0.,0.,0.  /
      DATA ETA     /1.0/
      DATA GINT   /10*0.0       /
      DATA GO     /32.174       /
      DATA GXP,GYP,GZP/45*0.0/
      DATA HEATK /1.,17600.,26000./
      DATA ALTITO/0.0           /
      DATA HRAT   /10*0.0       /
      DATA ISPV   /15*1.E11     /
C      1960 FISCHER EARTH MODEL
      DATA J2     /1.0823E-3    /
      DATA J3     /0.0          /
      DATA J4     /0.0          /
      DATA LREF    /1.0          /
      DATA LREFY /0.0           /
C      1960 FISCHER EARTH MODEL
      DATA MU     /1.4076539E+16 /
      DATA OMEGA /7.29211E-5    /
      DATA PGCLAT/0.0           /
      DATA PSL     /1HU          /
      DATA PWPROP/0.0           /
      DATA RHOSL /.0023769      /
      DATA RN      /1.0          /
C      1960 FISCHER EARTH MODEL
      DATA RE      /20925741.    /
      DATA RP      /20855590.    /
      DATA SREF   /0.0          /
      DATA TSL    /1HU          /
      DATA WGTSG /1.E-10        /
      DATA WJETT,WPLD,WPROPI,WEICON/4*0.0/
      DATA XREF   /3*0.0        /
      DATA AEXP    /.64  /
      DATA CINF    /1.0  /
      DATA VINFI   /.007 /
      DATA IENGA /15*1              /
      DATA IENGT /15*1              /
      DATA ITAP  /10*0.0            /
      DATA IWPF  /15*0.0            /
      DATA NENG  /1                 /
      DATA NEQS  /3*0.0             /
      DATA NEWSTG/0.0               /
                            DATA NPC
     1/0    ,1    ,4    ,2    ,2    ,0    ,0    ,1    ,0    ,0
     2,0    ,0    ,0    ,0    ,0    ,0    ,0    ,0    ,1    ,0
     3,15*0
     4/
      DATA GHA   /0.0               /
      DATA GHAS  /180.0             /
      DATA DECL  /0.0               /
      DATA TRPM  /0.0               /
      DATA DVMARR/0.0               /
      DATA DATE  /3*1HU             /
C
C     MOTVC
C
      DATA MOTVC1/141*0             /
      DATA GCLAT /1HU               /
      DATA GDLAT /0                 /
      DATA LONG  /1HU               /
      DATA MOTVC2/9*0               /
      DATA GCRAD /1HU               /
      DATA MOTVC3/148*0             /
      DATA ISV   / 1          /
      DATA ISV3  /0                 /
      DATA DIARP /3*0.0             /
      DATA DRAGP /3*0.0             /
      DATA DRAGPT/0.0               /
      DATA FAXBP /3*0.0             /
      DATA CDP   /3*0.0             /
      DATA DIAMP /3*0.0             /
      DATA DRGPK /3*0.0             /
      DATA DRGPP /3*0.0             /
      DATA VELAP /0.0               /
      DATA DRGPS /3*0.0             /
      DATA IDRGP /3*0               /
      DATA PARIF /3*0.0             /
      DATA MOTVC4 /26*0/
C
C     PHZVC
C
      DATA ALTMAX/1.E20     /
      DATA ALTMIN/-5000.   /
      DATA MAXTIM/1.0E10   /
      DATA EVTF  /0                 /
      DATA FESN  /100               /
      DATA IESN,PHZF,PIF,I4/4*0/
      DATA SAVESN /0/
C
C     SPECAL
C
      DATA SPECA1/24*0           /
C
C     TGOVC
C
      DATA FUXN  /10*0           /
      DATA PCTGO /     .9 /
      DATA SAVE /80*0/
      DATA TGO   /10.0E10/
      DATA TIMX,ESN/2*0/
      DATA IEVNT /10*0           /
      DATA ISZEV /0              /
      DATA NXEVT /3       /
      DATA I5     /0             /
      DATA GUXN /10*0/
      DATA TIMY /0/
C
C
C     TRACKC
C
      DATA CTKLAT /10*0.0         /
      DATA CTKLON /10*0.0         /
      DATA ELEV   /10*0.0         /
      DATA LKA    /10*0.0         /
      DATA LKB    /10*0.0         /
      DATA PGT   /90*0.0          /
      DATA SLNTRG /10*0.0         /
      DATA SLOS1  /10*0           /
      DATA SLOS2  /10*0           /
      DATA SLOS3  /10*0           /
      DATA STKLAT /10*0.0         /
      DATA STKLON /10*0.0         /
      DATA TKLATC /10*0.0         /
      DATA TKRAD  /10*0.0         /
      DATA TRKAZM /10*0.0         /
      DATA TRKGLT /28.22655 ,28.41338 ,26.62278 ,34.8155 ,6*0.0   /
      DATA TRKHIT /49.0     ,34.0     ,45.0     ,22.0    ,6*0     /
      DATA TRKLON /279.40002,279.40723,281.65167,283.64607,6*0.0/
      DATA TRKXRI /30*0.0         /
      DATA JTKFLG /6*0            /
      DATA TRKNAM /10HPATRICK   ,10HCAPE KEN  ,10HGRAND BAHA,
     1            10HALTANT FLD,6*10H          /
      DATA NTRKS  /10             /
      DATA NDUM   /10             /
      DATA ELEMIN/0.0             /
C
C     GUIDIC
C
      DATA DTG    /1.0             /
      DATA GVRI  /10*0          /
      DATA IGF   /6*0           /
C
C     GUIDVC
C
      DATA GTIME /1.E10         /
      DATA GUIDV1 /35*0   /
C
C     TARGVC
C
      DATA TARGV1 /133*0/
C
C     MVEHF(1) TURNS ON THE INTEGRATION OF TARGET VEHICLE
C     MVEHF(2) SPECIFIES TARGET INITIALIZATION
C
C
C     DYTEM
C
      DATA NMAX  /40             /
C
C     LOCAL
C
      DATA LOCAL1/145*0.0       /
      DATA MSKFLG /40000000000000000000B,37777777777777777777B/
      DATA ACOBS /16*0/
      DATA ALTIO /0/
C
C     KROIC
C
      DATA KROIC1 /13*0/
      DATA    GAM         /  1.0,.5,.416666666666667,.375,
     * .348611111111111,  .329861111111111,  .315591931216931,
     * .304224537037037,  .294868000440917,  .286975446428571,
     * .280189596443937,  .274265540031599,  .269028846773649,
     *                            .5,.166666666666667,.125,.105555555555556,
     * 9.375E-2           ,  8.56150793650793E-2,  7.95717592592593E-2,
     * 7.48522927689594E-2,  7.10329861111111E-2,  6.78584998463470E-2,
     * 6.51646205357143E-2,  6.28403190954034E-2,  6.08074792915494E-2/
      DATA GAS/1.0,-.5,-8.33333333333333E-2,-4.16666666666667E-2,
     * -2.63888888888889E-2,  -1.875E-2,
     * -1.42691798941799E-2,  -1.13673941798942E-2,
     * -9.35653659611993E-3,  -7.89255401234568E-3,
     * -6.78584998463471E-3,  -5.92405641233767E-3,
     * -5.23669325795029E-3/
                                  DATA
     1  PT     /   1.,   2.,   4.,   8.
     2         ,  16.,  32.,  64., 128.
     3         , 256., 512.,1024.,2048.
     4         ,4096.,8192.                 /
      DATA (ETP(I),I=1,13)/      3.33333330E-01, 2.50000000E-01,
     1  1.13636360E-01, 6.73076930E-02, 4.60526330E-02, 3.43749980E-02,
     2  2.71381590E-02, 2.22547310E-02, 1.87484580E-02, 1.61123220E-02,
3  1.40603000E-02, 1.24197060E-02, 1.10802170E-02/
 DATA (ETP(I),I=14,26)/     2.00000000E-01, 4.00000000E-01,
1  3.40909090E-01, 2.01923080E-01, 1.38157900E-01, 1.03124990E-01,
2  8.14144780E-02, 6.67641930E-02, 5.62453730E-02, 4.83369670E-02,
3  4.21809010E-02, 3.72591170E-02, 3.32406510E-02/
 DATA (ETP(I),I=27,39)/     1.42857140E-01, 2.85714280E-01,
1  3.42857140E-01, 3.46153840E-01, 2.45614040E-01, 1.87500000F-01,
2  1.50303650E-01, 1.24626490E-01, 1.05873640E-01, 9.15858320E-02,
3  8.03445710E-02, 7.12783130E-02, 6.38220510E-02/
 DATA (ETP(I),I=40,52)/     1.11111110E-01, 2.22222220E-01,
1  2.85714280E-01, 2.53968250E-01, 3.07017540E-01, 2.50000000E-01,
2  2.08755060E-01, 1.78037850E-01, 1.54399060E-01, 1.35682710F-01,
3  1.20516850E-01, 1.07997450E-01, 9.75059080E-02/
 DATA (ETP(I),I=53,65)/     9.09090910E-02, 1.81818180E-01,
1  2.42424240E-01, 2.42424240E-01, 1.73160170E-01, 2.50000000E-01,
2  2.27732800E-01, 2.05428290E-01, 1.85278880E-01, 1.67608050E-01,
3  1.52231820E-01, 1.38853860E-01, 1.27181620E-01/
 DATA (ETP(I),I=66,78)/     7.69230760E-02, 1.53846150E-01,
1  2.09790210E-01, 2.23776220E-01, 1.86480190E-01, 1.11888110E-01,
2  1.91295550E-01, 1.91733070E-01, 1.85278880E-01, 1.75988460E-01,
3  1.65763530E-01, 1.55516330E-01, 1.45680770E-01/
 DATA (ETP(I),I=79,91)/     6.66666660E-02, 1.33333330E-01,
1  1.84615380E-01, 2.05128200E-01, 1.86480190F-01, 1.34265730E-01,
2  6.96192690E-02, 1.39442230E-01, 1.52023690E-01, 1.56434190E-01,
3  1.56012730E-01, 1.52787970E-01, 1.47993160E-01/
 DATA (ETP(I),I=92,104)/    5.88235290E-02, 1.17647060E-01,
1  1.64705880E-01, 1.88235290E-01, 1.80995470E-01, 1.44796380E-01,
2  9.21431500E-02, 4.21225830E-02, 9.77295190E-02, 1.14931240E-01,
3  1.25367380E-01, 1.30961120E-01, 1.33193850E-01/
 DATA (ETP(I),I=105,117)/   5.26315790E-02, 1.05263160E-01,
1  1.48606810E-01, 1.73374610E-01, 1.73374610E-01, 1.48606810E-01,
2  1.06692070E-01, 6.09668970E-02, 2.49410030E-02, 6.63064840E-02,
3  8.35782530E-02, 9.62949390E-02, 1.05153040E-01/
 DATA (ETP(I),I=118,130)/   4.76190480E-02, 9.52380950E-02,
1  1.35338350E-01, 1.60401000E-01, 1.65118680E-01, 1.48606810E-01,
2  1.15583080E-01, 7.54828240E-02, 3.91930050E-02, 1.45159280E-02,
3  4.37790860E-02, 5.88469080E-02, 7.14002090F-02/
 DATA (ETP(I),I=131,143)/   4.34782610E-02, 8.69565210E-02,
1  1.24223600E-01, 1.49068320E-01, 1.56914020E-01, 1.46453090E-01,
2  1.20608430E-01, 8.61488760E-02, 5.16893250E-02, 2.46139640E-02,
3  8.33088030E-03, 2.82465160E-02, 4.03201180E-02/
 DATA (ETP(I),I=144,156)/   4.00000000E-02, 7.99999990E-02,
1  1.14782610E-01, 1.39130430E-01, 1.49068320E-01, 1.43105590E-01,
2  1.23020590E-01, 9.37299770E-02, 6.20271900E-02, 3.44595500E-02,
3  1.51622020E-02, 4.72588120E-03, 1.78691420E-02/
 DATA (ETP(I),I=157,169)/   3.70370370E-02, 7.40740740E-02,
1  1.06666670E-01, 1.30370370F-01, 1.41706920E-01, 1.39130430E-01,
2  1.23671500E-01, 9.89371980E-02, 7.02974820E-02, 4.33935080F-02,
3  2.24625220E-02, 9.18921340E-03, 2.65466170E-03/
 DATA FAC /1.,.5/
 DATA FPP1  /.1      /
 DATA FPP01 /.01     /
```

```
      DATA FPP25 /.25    /
C
C     KROVC
C
      DATA KROVC1 /27*0/
      DATA KBIT2 /98/
      DATA KDC /0/
      DATA KDD /0/
      DATA KDS /0/
      DATA KMAXO /2/
      DATA KQM /0/
      DATA KQMAX /12/
      DATA KROVC2 /8*0/
      DATA RNDINT /2.8E-14/
      DATA KROVC3 /59*0/
C
C     NPC(1) FLAGS CONIC CALCULATION
C     NPC(2) FLAGS INTEGRATION SCHEME
C     NPC(3) FLAGS INITIAL VELOCITY INPUT OPTION
C     NPC(4) FLAGS INITIAL POSITION INPUT OPTION
C     NPC(5) FLAGS ATMOSPHERE MODEL
C     NPC(6) FLAGS ATMOSPHERE WINDS
C     NPC(7) FLAGS ACCELERATION LIMIT BY THROTTLING
C     NPC(8) FLAGS AERODYNAMIC COEFFICIENT TYPE
C     NPC(9) FLAGS ENGINE TYPE
C     NPC(10) FLAGS STATIC TRIM
C     NPC(11) FLAGS ENVIRONMENTAL INEQUALITY CONSTRAINTS
C     NPC(12) FLAGS CROSS AND DOWNRANGE CALCULATION
C     NPC(13) FLAGS PROPELLANT JETTISON OPTION
C     NPC(14) FLAGS HOLD DOWN OPTION
C     NPC(15) FLAGS HEATING RATE
C     NPC(16) FLAGS EARTH MODEL
C     NPC(17) FLAGS MASS FRACTION JETTISON OPTION
C     NPC(18) FLAGS TRAJECTORY TERMINATION
C     NPC(19) FLAGS INPUT CONDITIONS PRINTOUT
C     NPC(20) FLAGS DT MODEL
C     NPC(21) FLAGS FLOWRATE METHOD FOR ROCKET ENGINES
C     NPC(22) FLAGS THROTTLING PARAMETER
C     NPC(23) FLAGS DVMAR CALCULATION
C     NPC(24) FLAGS GENERAL INTEGRATION
C     NPC(25) FLAGS VELOCITY LOSSES
C     NPC(26) FLAGS AERODYNAMIC HEATING INDICATORS
C     NPC(27) FLAGS INDIVIDUAL ENGINE FLOWRATES
C     NPC(28) FLAGS TRACKING STATIONS
C     NPC(29) FLAGS ANALYTIC IMPACT
C     NPC(30) FLAGS WEIGHT AS FUNCTION OF TABLE LOOKUP
C     NPC(31) FLAGS VERNAL EQUANOX CALCULATIONS
C     NPC(32) FLAGS PARACHUTE DRAG OPTION
C     NPC(33) FLAGS BTL COORDINATES COMPUTATIONS
C     NPC(34) - NOT USED
C     NPC(35) - NOT USED
C
      END
```

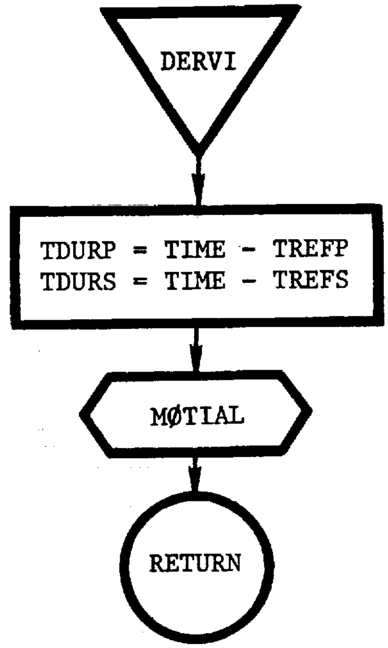
DELTU: This is the main program of overlay (2,5). This routine determines the direction of search, based on the search/optimization mode selected by user input.


DELTU
CALL WUCAL FOR PROBLEM WEIGHTING
CALL GMAG TO CALCULATE GRADIENT MAGNITUDES
BRANCH ON SEARCH MODE (SRCHM)
=1
20
CALL SDM, WHICH IS THE STEEPEST-DESCENT METHOD
=2
40
CALL CGM, WHICH IS THE CONJUGATE GRADIENT METHOD
=3
60
CALL DGMP2, WHICH IS THE DAVIDON METHOD ON P2
=4
80
CALL PGM, WHICH IS THE PROJECTED GRADIENT
CALL UNITDU TO UNITIZE THE DIRECTION OF SEARCH
IF SRCHM : 4
=
CALL GABDD TO DETERMINE INEQUALITY STEP-SIZE BOUNDARIES
≠
RETURN

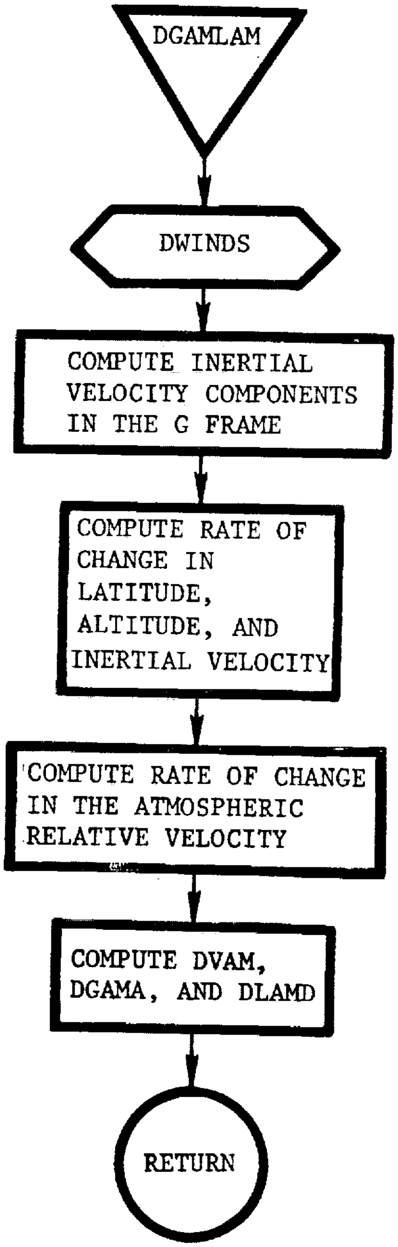
DERIV: This routine updates the time references and calls the computational routines.



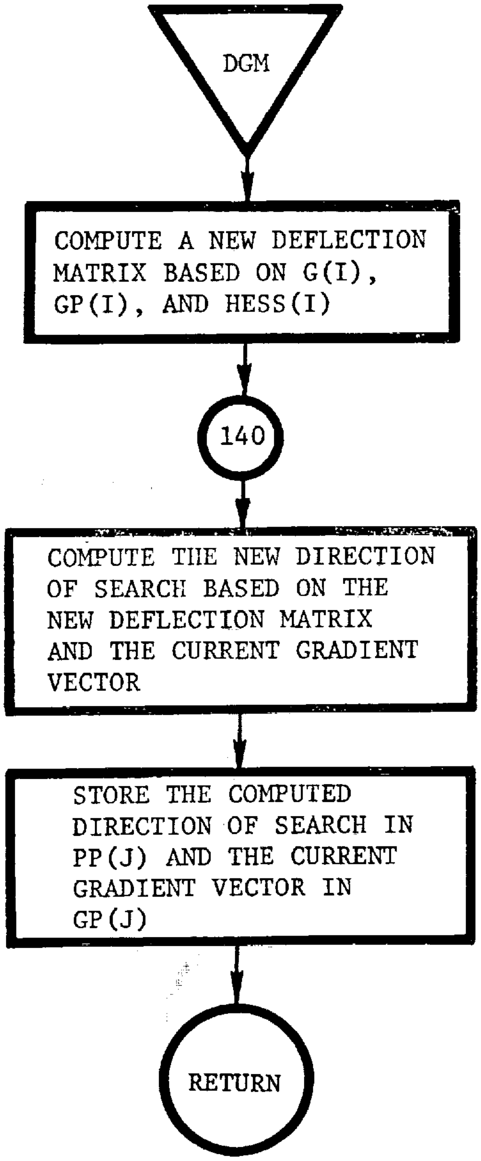
DERVI: This routine initializes the time references and calls the routines to initialize the equations of motion.



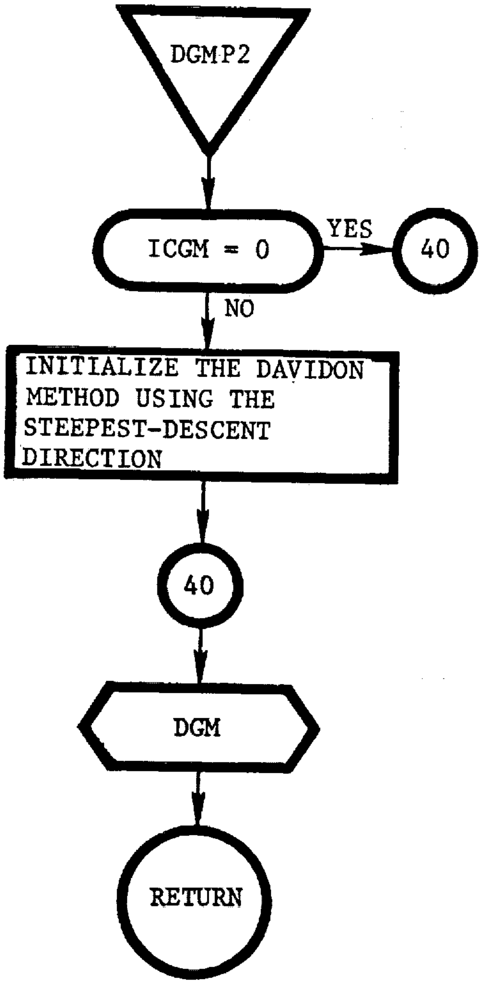
DGAMLAM: This routine computes the rate of change in path angle and azimuth relative to the atmosphere.


DGAMLAM
DWINDS
COMPUTE INERTIAL VELOCITY COMPONENTS IN THE G FRAME
COMPUTE RATE OF CHANGE IN LATITUDE, ALTITUDE, AND INERTIAL VELOCITY
COMPUTE RATE OF CHANGE IN THE ATMOSPHERIC RELATIVE VELOCITY
COMPUTE DVAM, DGAMA, AND DLAMD
RETURN

DGM (G): This routine computes the direction of search based on the Davidon deflected gradient method and is used to minimize a scalar-valued function whose gradient vector is given by G(I).


DGM
COMPUTE A NEW DEFLECTION MATRIX BASED ON G(I), GP(I), AND HESS(I)
140
COMPUTE THE NEW DIRECTION OF SEARCH BASED ON THE NEW DEFLECTION MATRIX AND THE CURRENT GRADIENT VECTOR
STORE THE COMPUTED DIRECTION OF SEARCH IN PP(J) AND THE CURRENT GRADIENT VECTOR IN GP(J)
RETURN

DGMP2: This routine computes the direction of search for minimizing P2 via the Davidon variable metric method.


DGMP2
ICGM = 0
YES
40
NO
INITIALIZE THE DAVIDON METHOD USING THE STEEPEST-DESCENT DIRECTION
40
DGM
RETURN

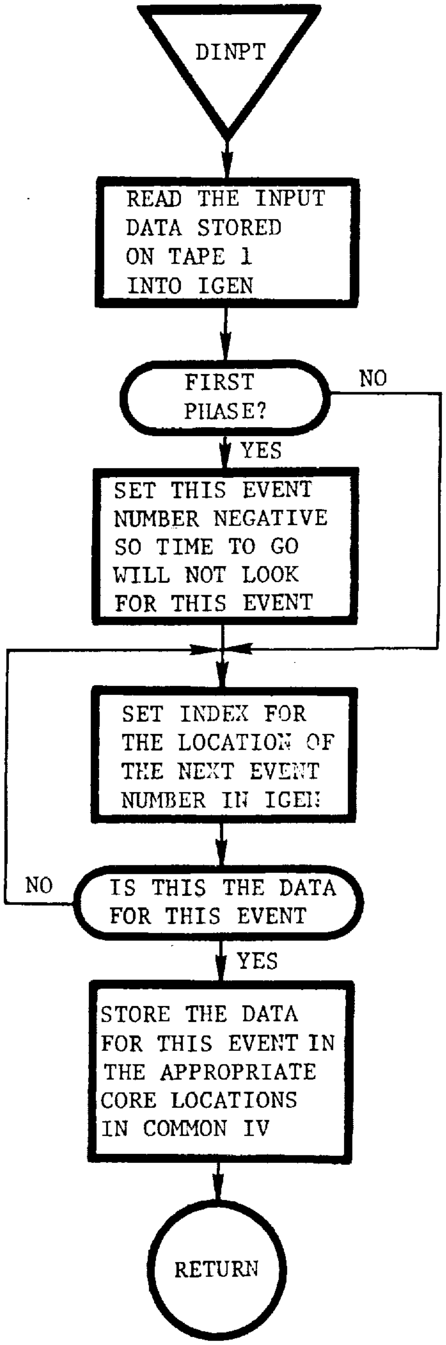
DICT: This routine is not call explicity, but maps the variable names to core locations at the time overlay (1,0) is called.

```
*DECK,DICT
      SUBROUTINE DICT
C*** DICT
C              DICT - DEFINES COMPUTATIONAL COMMONS
C              (IV - END) DICTIONARY VALUES DATA
C*** THE DATA STATEMENTS IN THIS ROUTINE ARE STANDARDIZED
C*** THERE MUST BE AN EVEN NUMBER OF INTEGERS IN COMMONS BETWEEN
C    IV AND END FOR THE UNIVAC 1108 DOUBLE PRECISION VERSION
C
      COMMON/IV/ IV(2)
C
C      C O M P U T A T I O N A L  D A T A  R E G I O N
C
*CALL AUXVC
*CALL CYCVC
*CALL DPGVC
*CALL DYNVC
*CALL INFVC
*CALL MNMMLT
*CALL HOLINC
*CALL MOTBL
*CALL MOTIC
*CALL MOTVC
*CALL PHZVC
*CALL SPECAL
*CALL TGOVC
*CALL TRACKC
*CALL GUIDIC
*CALL GUIDVC
*CALL TARGVC
      COMMON/END/END
C
C***   C O M M O N S  N O T  I N C L U D E D  I N  D I C T I O N A R Y
C
*CALL DYTEM
C
C***   C O M P U T A T I O N A L  D A T A  D I C T I O N A R Y
C
      DIMENSION AUXVC1(126)
      DIMENSION AUXVC2(60)
      DIMENSION CYCVC1(16)
      DIMENSION DPGVC1(126)
      DIMENSION DPGVC2(119)
      DIMENSION DYNVC1(18)
      DIMENSION INFVC1(24)
      DIMENSION MNMML1(17)
      DIMENSION HOLIN1(51)
      DIMENSION MOTIC1(126)
      DIMENSION MOTIC2(126)
      DIMENSION MOTIC3(  1)
      DIMENSION MOTVC1(126)
      DIMENSION MOTVC2(126)
      DIMENSION MOTVC3(110)
      DIMENSION PHZVC1(10)
      DIMENSION SPECA1(24)
      DIMENSION TGOVC1(118)
      DIMENSION TRACK1(126)
      DIMENSION TRACK2(126)
      DIMENSION TRACK3( 61)
      DIMENSION GUIDI1(17)
      DIMENSION GUIDV1(37)
      DIMENSION TARGV1(126)
      DIMENSION TARGV2(7)
C
      EQUIVALENCE (AUXVC1(1),ALPTOT    )
      EQUIVALENCE (AUXVC2(1),XR(3)     )
      EQUIVALENCE (CYCVC1(1),DELT      )
      EQUIVALENCE (DPGVC1(1),ALPHA     )
      EQUIVALENCE (DPGVC2(1),KDG(2)    )
      EQUIVALENCE (DYNVC1(1),DTIMR(1)  )
      EQUIVALENCE (INFVC1(1),ESNPRT    )
      EQUIVALENCE (MNMML1(1),ONE       )
      EQUIVALENCE (HOLIN1(1),ALPARG    )
      EQUIVALENCE (MOTIC1(1),ALTIP     )
      EQUIVALENCE (MOTIC2(1),LREFY     )
      EQUIVALENCE (MOTIC3(1),DATE(3)   )
      EQUIVALENCE (MOTVC1(1),AHI       )
      EQUIVALENCE (MOTVC2(1),GXI(3)    )
      EQUIVALENCE (MOTVC3(1),XLO(3)    )
      EQUIVALENCE (PHZVC1(1),ALTMAX    )
      EQUIVALENCE (SPECA1(1),SPECI(1)  )
      EQUIVALENCE (TGOVC1(1),FUXN(1)    )
      EQUIVALENCE (TRACK1(1),CTKLAT(1))
      EQUIVALENCE (TRACK2(1) ,PGT(77)   )
      EQUIVALENCE (TRACK3(1),TRKLON(3))
      EQUIVALENCE (GUIDI1(1),DTG       )
      EQUIVALENCE (GUIDV1(1),GTIME     )
      EQUIVALENCE (TARGV1(1),ALTAT     )
      EQUIVALENCE (TARGV2(1),DAXPT(3)  )
C
C
C
                             DATA AUXVC1
     1 /6HALPTOT,6HALTA   ,6HALTP   ,6HANGMOM,6HAPORAD,6HAPVEL ,6HARGP
     2 ,6HARGV  ,6HDECLIN,6HDRAG   ,6HLIFT   ,6HDGENV ,6HDPRNG1,6HDPRNG2
     3 ,6HDWNRNG,6HDVCIR ,6HDVEXS ,6HECCAN ,6HECCEN ,6HENERGY,6HGDLTIP
     4 ,6HHYPVEL,6HINC    ,6HINCPCH,6HINCYAW,6HIPNULL,6HIYNULL,6HIRANGE
     5 ,6HIRAGNE,6HLAN    ,6HLONGIP,6HMEAAN ,6HPERIOD,6HPGERAD,6HPGLON
     6 ,6HPGVEL ,6HQALTOT,6HREYNO ,6HRIP1   ,6HRIP2   ,6HRIP3   ,6HRTASC
     7 ,6HSEMJAX,6HSIDEAL,6HSSVIDL,6HTHTP   ,6HTHTPL  ,6HTIMIP  ,6HTIMSP
     8 ,6HTIMTP ,6HTLHT1 ,6HTLHT2 ,6HTLHT3 ,6HTLHT4 ,6HTLHT5 ,6HTLHT6
     9 ,6HTLHT7 ,6HTLHT8 ,6HTLHT9 ,6HTLHT10,6HTLPWT ,6HTRUAN ,6HTRUNMX
     0 ,6HTTLISP,6HUBAR   ,6HVIP1   ,6HVIP2   ,6HVIP3   ,6HVMU    ,6HWTP1
     A ,6HWTP2  ,6HWTP3  ,6HWTP4  ,6HWTP5  ,6HWTP6  ,6HWTP7  ,6HWTP8
     B ,6HWTP9   ,6HWTP10 ,6HXISAV1,6HXISAV2,6HXISAV3,6HXMAX1 ,6HXMAX2
     C ,6HXMAX3  ,6HXMAX4 ,6HXMAX5 ,6HXMAX6 ,6HXMAX7 ,6HXMAX8 ,6HXMAX9
     D ,6HXMAX10,6HXMIN1 ,6HXMIN2 ,6HXMIN3 ,6HXMIN4 ,6HXMIN5 ,6HXMIN6
     E ,6HXMIN7  ,6HXMIN8 ,6HXMIN9 ,6HXMIN10,6HCRRNG ,6HYXMN1 ,6HYXMN2
     F ,6HYXMN3  ,6HYXMN4 ,6HYXMN5 ,6HYXMN6 ,6HYXMN7 ,6HYXMN8 ,6HYXMN9
     G ,6HYXMN10,6HYXMX1 ,6HYXMX2 ,6HYXMX3 ,6HYXMX4 ,6HYXMX5 ,6HYXMX6
     H ,6HYXMX7  ,6HYXMX8 ,6HYXMX9 ,6HYXMX10,6HMACHDT,6HXR     ,6HYR
     I /
C
                                 DATA AUXVC2
     1 /6HZR     ,6HXVE   ,6HYVE   ,6HZVE   ,6HVXVE  ,6HVYVE  ,6HVZVE
     2 ,6HXSVE   ,6HYSVE  ,6HZSVE  ,6HXSI   ,6HYSI   ,6HZSI   ,6HSHADF
     3 ,6HSCONE  ,6HSCLOCK,6HXIVE11,6HXIVE12,6HXIVE13,6HXIVE21,6HXIVE22
     4 ,6HXIVE23,6HXIVE31,6HXIVE32,6HXIVE33,6HURX   ,6HURY   ,6HURZ
     5 ,6HUTX    ,6HUTY   ,6HUTZ   ,6HUNX   ,6HUNY   ,6HUNZ   ,6HRAS
     6 ,6HLANVE  ,6HVCIRC ,6HGULIE ,6HRASGM ,6HBTLX  ,6HBTLY  ,6HBTLZ
     7 ,6HBTLXD  ,6HBTLYD ,6HBTLZD ,6HZETA1 ,6HZETA2 ,6HZETA3 ,6HNTRK
     8 ,6HDPRGI1,6HDPRGI2,6HGCLTIP,6HGENV1 ,6HGENV2 ,6HPGENV ,6HRGENV
     9 ,6HSGENV  ,6HDTRUAN,6HDYNPD ,6HBETAS
     I /
C
C
C
                                 DATA CYCVC1
     1 /6HDELT   ,6HDT    ,6HDTIME ,6HDTM   ,6HDTO   ,6HENOIS ,6HTREF
     2 ,6HIDTAB1,6HIDTAB2,6HIDTAB3,6HIDTAB4,6HIDTAB5,6HIDTAB6,6HIFLG
     3 ,6HCYCF   ,6HDELTT
     4 /
C
C
C
                                 DATA DPGVC1
     1 /6HALPHA  ,6HBETA  ,6HBNKANG,6HALPPC1,6HALPPC2,6HALPPC3,6HALPPC4
     2 ,6HBETPC1,6HBETPC2,6HBETPC3,6HBETPC4,6HBNKPC1,6HBNKPC2,6HBNKPC3
     3 ,6HBNKPC4,6HALPDOT,6HBETDOT,6HBNKDOT,6HDALPHA,6HDBETA ,6HDBANK
     4 ,6HDYAW   ,6HDPITCH,6HDROLL ,6HAB1   ,6HAB2   ,6HAB3   ,6HAB4
     5 ,6HAB5    ,6HAB6   ,6HAB7   ,6HAB8   ,6HAB9   ,6HGB1   ,6HGB2
     6 ,6HGB3    ,6HGB4   ,6HGB5   ,6HGB6   ,6HGB7   ,6HGB8   ,6HGB9
     7 ,6HIA1    ,6HIA2   ,6HIA3   ,6HIA4   ,6HIA5   ,6HIA6   ,6HIA7
     8 ,6HIA8    ,6HIA9   ,6HIB11  ,6HIB12  ,6HIB13  ,6HIB21  ,6HIB22
     9 ,6HIB23   ,6HIB31  ,6HIB32  ,6HIB33  ,6HIG1   ,6HIG2   ,6HIG3
     0 ,6HIG4    ,6HIG5   ,6HIG6   ,6HIG7   ,6HIG8   ,6HIG9   ,6HIL1
     A ,6HIL2    ,6HIL3   ,6HIL4   ,6HIL5   ,6HIL6   ,6HIL7   ,6HIL8
     B ,6HIL9    ,6HLB1   ,6HLB2   ,6HLB3   ,6HLB4   ,6HLB5   ,6HLB6
     C ,6HLB7    ,6HLB8   ,6HLB9   ,6HHARG1 ,6HHARG2 ,6HHARG3 ,6HHERROR
     D ,6HHERROR,6HHERROR,6HIDGF1 ,6HIDGF2 ,6HIDGF3 ,6HIGUID1,6HIGUID2
     E ,6HIGUID3,6HIGUID4,6HIGUID5,6HIGUID6,6HIGUID7,6HIGUID8,6HIGUID9
     F ,6HIGUID1,6HIGUID1,6HIGUID1,6HIGUID1,6HIGUID1,6HIGUID1,6HIGUID1
     G ,6HIGUID1,6HIGUID1,6HIGUID1,6HIGUID2,6HIGUID2,6HIGUID2,6HIGUID2
     H ,6HIGUID2,6HIGUID2,6HIVCRT1,6HIVCRT2,6HIVCRT3,6HIVETA ,6HKDG1
     I /
                                 DATA DPGVC2
     1 /6HKDG2  ,6HKDG3  ,6HKRG1  ,6HKRG2  ,6HKRG3  ,6HYAWI  ,6HPITI
     2 ,6HROLI  ,6HYAWR  ,6HPITR  ,6HROLR  ,6HROLPC1,6HROLPC2,6HROLPC3
     3 ,6HROLPC4,6HYAWPC1,6HYAWPC2,6HYAWPC3,6HYAWPC4,6HPITPC1,6HPITPC2
     4 ,6HPITPC3,6HPITPC4,6HROLBD ,6HPITBD ,6HYAWBD ,6HYAWID ,6HPITID
     5 ,6HROLID ,6HYAWRD ,6HPITRD ,6HROLRD ,6HXYOMA1,6HXYOMA2,6HXYOMA3
     6 ,6HXYOMB1,6HXYOMB2,6HXYOMB3,6HXYOMS1,6HXYOMS2,6HXYOMS3,6HXYOM11
     7 ,6HXYOM12,6HXYOM13,6HXYOM21,6HXYOM22,6HXYOM23,6HXYOM31,6HXYOM32
     8 ,6HXYOM33,6HXYOME1,6HXYOME2,6HXYOME3,6HGUIDXH,6HGUIDYH,6HGUIDZH
     9 ,6HALPHI ,6HBETAI ,6HBANKI ,6HIIV11 ,6HIIV12 ,6HIIV13 ,6HIIV21
     0 ,6HIIV22 ,6HIIV23 ,6HIIV31 ,6HIIV32 ,6HIIV33 ,6HIVB11 ,6HIVE12
     A ,6HIVE13 ,6HIVB21 ,6HIVB22 ,6HIVB23 ,6HIVB31 ,6HIVB32 ,6HIVB33
     B ,6HNDEPVS,6HDPVLS1,6HDPVLS2,6HDPVLS3,6HDPVLS4,6HDPTLS1,6HDPTLS2
     C ,6HDPTLS3,6HDPTLS4,6HMAXITS,6HNITS  ,6HPERTS1,6HPERTS2,6HPERTS3
     D ,6HPERTS4,6HENOMS1,6HENOMS2,6HENOMS3,6HENOMS4,6HES1   ,6HES2
     E ,6HES3   ,6HES4   ,6HUNOMS1,6HUNOMS2,6HUNOMS3,6HUNOMS4,6HUS1
     F ,6HUS2   ,6HUS3   ,6HUS4   ,6HENORMS,6HISTEPS,6HNAMVS1,6HNAMVS2
     G ,6HNAMVS3,6HNAMVS4,6HNAMVS5,6HNAMVS6,6HNAMVS7,6HNAMVS8,6HGSAITS
     I /
C
C
C
                          DATA DYNVC1
     1 /6HDTIMR1,6HDTIMR2,6HDTIMR3,6HDTIMR4,6HTDURP ,6HTIMES ,6HTIMRF1
     2 ,6HTIMRF2,6HTIMRF3,6HTIMRF4,6HTREFP ,6HTREFS ,6HNDISC ,6HNPASS
     3 ,6HERRINT,6HDLTMIN,6HDLTMAX,6HEPSINT
     4 /
C
C
C
                          DATA INFVC1
     1 /6HESNPRT,6HEXTRAP,6HLPRNT ,6HPINC  ,6HPRNC  ,6HFID1  ,6HFID2
     2 ,6HINFF  ,6HIPRNTB,6HIPRNTR,6HPSTOP ,6HTITLE1,6HTITLE2,6HTITLE3
     3 ,6HTITLE4,6HTITLE5,6HTITLE6,6HTITLE7,6HTITLE8,6HTITLE9,6HTITLE1
     4 ,6HSFID1 ,6HSFID2 ,6HNPINC
     5/
C
C
C
                          DATA MNMML1
     1 /6HONE   ,6HCADPNM,6HCADYNM,6HCANM  ,6HCDDPNM,6HCDDYNM,6HCDNM
     2 ,6HCLDPNM,6HCLNM  ,6HCMANM ,6HCMDPNM,6HCNANM ,6HCNDPNM,6HCWBNM
     3 ,6HCWDYNM,6HCYBNM ,6HCYDYNM
     4 /
C
C
C
                          DATA HOLIN1
     1 /6HALPARG,6HBETARG,6HBNKARG,6HETAARG,6HDGF1  ,6HDGF2  ,6HDGF3
     2 ,6HGDERV1,6HGDERV2,6HGDERV3,6HGDERV4,6HGDFRV5,6HGDERV6,6HGDERV7
     3 ,6HGDERV8,6HGDERV9,6HGDERV0,6HMONF1 ,6HMONF2 ,6HMONF3 ,6HMONX1
     4 ,6HMONX2 ,6HMONX3 ,6HMONX4 ,6HMONX5 ,6HMONX6 ,6HMONX7 ,6HMONX8
     5 ,6HMONX9 ,6HMONX10,6HMONY1 ,6HMONY2 ,6HMONY3 ,6HMONY4 ,6HMONY5
     6 ,6HMONY6 ,6HMONY7 ,6HMONY8 ,6HMONY9 ,6HMONY10,6HYAWARG,6HPITARG
     7 ,6HROLARG,6HDPVRS1,6HDPVRS2,6HDPVRS3,6HDPVRS4,6HINVRS1,6HINVRS2
     8 ,6HINVRS3,6HINVRS4
     8 /
C
C
C
      DATA IV(2)   /6H*     /
C
C
      DATA CST    /6HCST   ,6HCSM   /
      DATA ATEMT  /6HATEMT ,6HATEMM /
      DATA PREST  /6HPREST ,6HPRESM /
      DATA VWUT   /6HVWUT  ,6HVWUM  /
      DATA VWVT   /6HVWVT  ,6HVWVM  /
      DATA VWWT   /6HVWWT  ,6HVWWM  /
      DATA AZWT   /6HAZWT  ,6HAZWM  /
      DATA VWT    /6HVWT   ,6HVWM   /
      DATA ALPHAT/6HALPHAT,6HALPHAM/
      DATA BETAT  /6HBETAT ,6HBETAM /
      DATA BANKT  /6HBANKT ,6HBANKM /
      DATA YAWT   /6HYAWT  ,6HYAWM  /
      DATA PITT   /6HPITT  ,6HPITM  /
      DATA ROLT   /6HROLT  ,6HROLM  /
      DATA CDT    /6HCDT   ,6HCDM   /
      DATA CLT    /6HCLT   ,6HCLM   /
      DATA CAT    /6HCAT   ,6HCAM   /
      DATA CNAT   /6HCNAT  ,6HCNAM  /
      DATA CYBT   /6HCYBT  ,6HCYBM  /
      DATA CMAT   /6HCMAT  ,6HCMAM  /
      DATA CWBT   /6HCWBT  ,6HCWBM  /
      DATA CMDPT  /6HCMDPT ,6HCMDPM /
      DATA CADPT  /6HCADPT ,6HCADPM /
      DATA CNDPT  /6HCNDPT ,6HCNDPM /
      DATA CWDYT  /6HCWDYT ,6HCWDYM /
      DATA CADYT  /6HCADYT ,6HCADYM /
      DATA CYDYT  /6HCYDYT ,6HCYDYM /
      DATA XCGT   /6HXCGT  ,6HXCGM  /
      DATA YCGT   /6HYCGT  ,6HYCGM  /
      DATA ZCGT   /6HZCGT  ,6HZCGM  /
                             DATA TVCIT
     1/6HTVC1T ,6HTVC1M ,6HTVC2T ,6HTVC2M ,6HTVC3T ,6HTVC3M
     2,6HTVC4T ,6HTVC4M ,6HTVC5T ,6HTVC5M ,6HTVC6T ,6HTVC6M
     3,6HTVC7T ,6HTVC7M ,6HTVC8T ,6HTVC8M ,6HTVC9T ,6HTVC9M
     4,6HTVC10T,6HTVC10M,6HTVC11T,6HTVC11M,6HTVC12T,6HTVC12M
     5,6HTVC13T,6HTVC13M,6HTVC14T,6HTVC14M,6HTVC15T,6HTVC15M
     6 /
                             DATA WDIT
     1/6HWD1T  ,6HWD1M  ,6HWD2T  ,6HWD2M  ,6HWD3T  ,6HWD3M
     2,6HWD4T  ,6HWD4M  ,6HWD5T  ,6HWD5M  ,6HWD6T  ,6HWD6M
     3,6HWD7T  ,6HWD7M  ,6HWD8T  ,6HWD8M  ,6HWD9T  ,6HWD9M
     4,6HWD10T ,6HWD10M ,6HWD11T ,6HWD11M ,6HWD12T ,6HWD12M
```

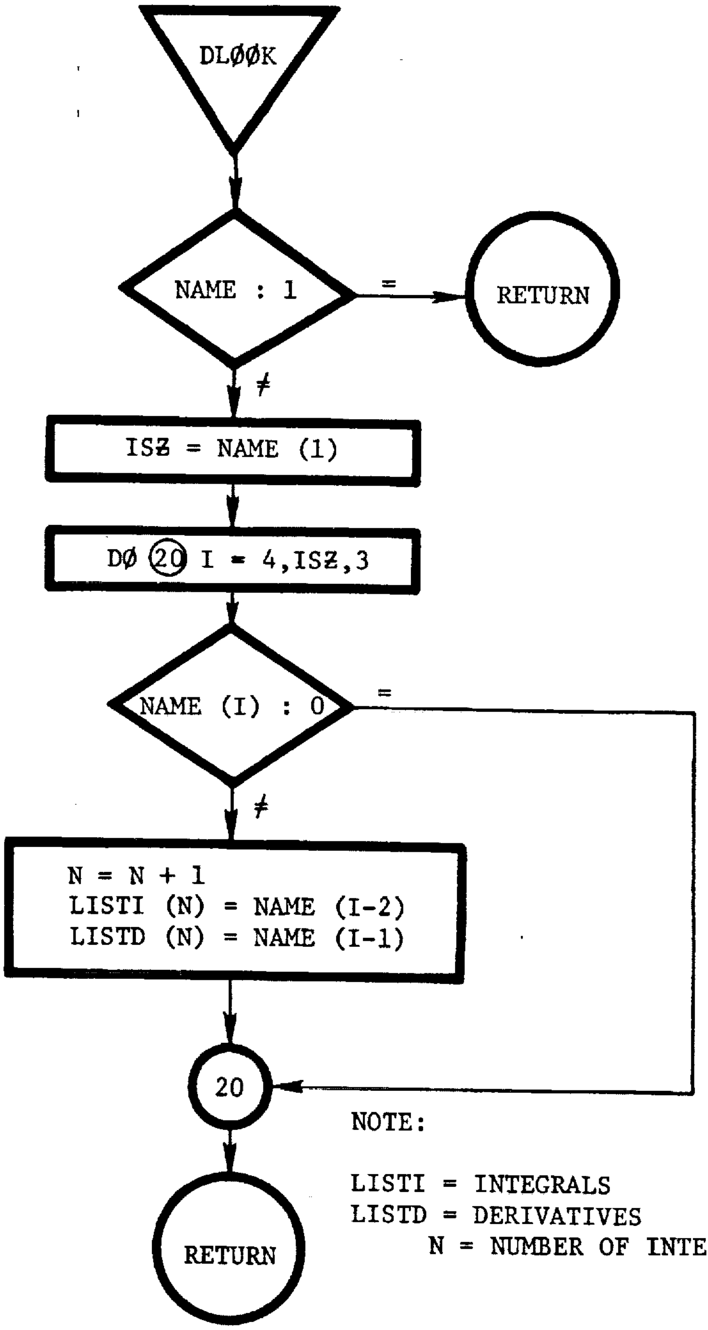
```
5,6HWD13T ,6HWD13M ,6HWD14T ,6HWD14M ,6HWD15T ,6HWD15M
6 /
                          DATA AEIT
1/6HAE1T  ,6HAE1M   ,6HAE2T   ,6HAE2M   ,6HAE3T   ,6HAE3M
2,6HAE4T  ,6HAE4M   ,6HAE5T   ,6HAE5M   ,6HAE6T   ,6HAE6M
3,6HAE7T  ,6HAE7M   ,6HAE8T   ,6HAE8M   ,6HAE9T   ,6HAE9M
7,6HAE10T ,6HAE10M  ,6HAE11T  ,6HAE11M  ,6HAE12T  ,6HAE12M
5,6HAE13T ,6HAE13M  ,6HAE14T  ,6HAE14M  ,6HAE15T  ,6HAE15M
6 /
 DATA FL1T  /6HFL1T  ,6HFL1M  /
 DATA FL2T  /6HFL2T  ,6HFL2M  /
 DATA FL3T  /6HFL3T  ,6HFL3M  /
 DATA XREFT /6HXREFT ,6HXREFM /
 DATA YREFT /6HYREFT ,6HYREFM /
 DATA ZREFT /6HZREFT ,6HZREFM /
 DATA CLDPT /6HCLDPT ,6HCLDPM /
 DATA CDDPT /6HCDDPT ,6HCDDPM /
 DATA CDDYT /6HCDDYT ,6HCDDYM /
 DATA DENST /6HDENST ,6HDENSM /
 DATA HTRTT /6HHTRTT ,6HHTRTM /
 DATA ETAT  /6HETAT  ,6HETAM  /
 DATA CAOT  /6HCAOT  ,6HCAOM  /
 DATA CNOT  /6HCNOT  ,6HCNOM  /
 DATA CYOT  /6HCYOT  ,6HCYOM  /
                          DATA PIIT,YIIT
1/6HPI1T  ,6HPI1M   ,6HPI2T   ,6HPI2M   ,6HPI3T   ,6HPI3M
2,6HPI4T  ,6HPI4M   ,6HPI5T   ,6HPI5M   ,6HPI6T   ,6HPI6M
3,6HPI7T  ,6HPI7M   ,6HPI8T   ,6HPI8M   ,6HPI9T   ,6HPI9M
4,6HPI10T ,6HPI10M  ,6HPI11T  ,6HPI11M  ,6HPI12T  ,6HPI12M
5,6HPI13T ,6HPI13M  ,6HPI14T  ,6HPI14M  ,6HPI15T  ,6HPI15M
6,6HYI1T  ,6HYI1M   ,6HYI2T   ,6HYI2M   ,6HYI3T   ,6HYI3M
7,6HYI4T  ,6HYI4M   ,6HYI5T   ,6HYI5M   ,6HYI6T   ,6HYI6M
8,6HYI7T  ,6HYI7M   ,6HYI8T   ,6HYI8M   ,6HYI9T   ,6HYI9M
9,6HYI10T ,6HYI10M  ,6HYI11T  ,6HYI11M  ,6HYI12T  ,6HYI12M
0,6HYI13T ,6HYI13M  ,6HYI14T  ,6HYI14M  ,6HYI15T  ,6HYI15M /
 DATA CDOT  /6HCDOT  ,6HCDOM  /
 DATA CLOT  /6HCLOT  ,6HCLOM  /
 DATA CWOT  /6HCWOT  ,6HCWOM  /
 DATA CMOT  /6HCMOT  ,6HCMOM  /
 DATA WUA1T /6HWUA1T ,6HWUA1M /
 DATA WUA2T /6HWUA2T ,6HWUA2M /
 DATA DENKT /6HDENKT ,6HDENKM /
 DATA GDF1T /6HGDF1T ,6HGDF1M /
 DATA GDF2T /6HGDF2T ,6HGDF2M /
 DATA GDF3T /6HGDF3T ,6HGDF3M /
 DATA GNOM1T/6HGNOM1T,6HGNOM1M/
 DATA GNOM2T/6HGNOM2T,6HGNOM2M/
 DATA GNOM3T/6HGNOM3T,6HGNOM3M/
 DATA GNMX1T/6HGNMX1T,6HGNMX1M/
 DATA GNMX2T/6HGNMX2T,6HGNMX2M/
 DATA GNMX3T/6HGNMX3T,6HGNMX3M/
 DATA GNMN1T/6HGNMN1T,6HGNMN1M/
```

```
      DATA GNMN2T/6HGNMN2T,6HGNMN2M/
      DATA GNMN3T/6HGNMN3T,6HGNMN3M/
      DATA GENV1T/6HGENV1T,6HGENV1M/
      DATA GENV2T/6HGENV2T,6HGENV2M/
      DATA FMASST/6HFMASST,6HFMASSM/
      DATA ZLALPT/6HZLALPT,6HZLALPM/
      DATA CAIOT /6HCAIOT ,6HCAIOM /
      DATA WGT1T /6HWGT1T ,6HWGT1M /
      DATA WGT2T /6HWGT2T ,6HWGT2M /
      DATA WGTD1T/6HWGTD1T,6HWGTD1M/
      DATA WGTD2T/6HWGTD2T,6HWGTD2M/
      DATA CDP1T /6HCDP1T ,6HCDP1M ,6HCDP2T ,6HCDP2M ,6HCDP3T ,6HCDP3M /
      DATA HTRT1T/6HHTRT1T,6HHTRT1M/
C
C
C
                             DATA MOTIC1
     1 /6HALTIP ,6HALTREF,6HLATREF,6HLONREF,6HAZREF ,6HTIMREF,6HARP1
     2 ,6HARP2  ,6HARP3  ,6HARP4  ,6HARP5  ,6HARP6  ,6HARP7  ,6HARP8
     3 ,6HARP9  ,6HARP10 ,6HASMAX ,6HATMSK1,6HATMSK2,6HAZWB  ,6HAZL
     4 ,6HLATL  ,6HLONL  ,6HCLCDMX,6HDETA  ,6HDESN1 ,6HDESN2 ,6HDESN3
     5 ,6HDESNE ,6HDVIMAG,6HDVMAR ,6HDVPCT ,6HETAPC1,6HETAPC2,6HETAPC3
     6 ,6HETAPC4,6HETA   ,6HGINT1 ,6HGINT2 ,6HGINT3 ,6HGINT4 ,6HGINT5
     7 ,6HGINT6 ,6HGINT7 ,6HGINT8 ,6HGINT9 ,6HGINT10,6HGO    ,6HGXP1
     8 ,6HGXP2  ,6HGXP3  ,6HGXP4  ,6HGXP5  ,6HGXP6  ,6HGXP7  ,6HGXP8
     9 ,6HGXP9  ,6HGXP10 ,6HGXP11 ,6HGXP12 ,6HGXP13 ,6HGXP14 ,6HGXP15
     0 ,6HGYP1  ,6HGYP2  ,6HGYP3  ,6HGYP4  ,6HGYP5  ,6HGYP6  ,6HGYP7
     A ,6HGYP8  ,6HGYP9  ,6HGYP10 ,6HGYP11 ,6HGYP12 ,6HGYP13 ,6HGYP14
     B ,6HGYP15 ,6HGZP1  ,6HGZP2  ,6HGZP3  ,6HGZP4  ,6HGZP5  ,6HGZP6
     C ,6HGZP7  ,6HGZP8  ,6HGZP9  ,6HGZP10 ,6HGZP11 ,6HGZP12 ,6HGZP13
     D ,6HGZP14 ,6HGZP15 ,6HHEATK1,6HHEATK2,6HHEATK3,6HALTITO,6HHRAT1
     E ,6HHRAT2 ,6HHRAT3 ,6HHRAT4 ,6HHRAT5 ,6HHRAT6 ,6HHRAT7 ,6HHRAT8
     F ,6HHRAT9 ,6HHRAT10,6HISPV1 ,6HISPV2 ,6HISPV3 ,6HISPV4 ,6HISPV5
     G ,6HISPV6 ,6HISPV7 ,6HISPV8 ,6HISPV9 ,6HISPV10,6HISPV11,6HISPV12
     H ,6HISPV13,6HISPV14,6HISPV15,6HJ2    ,6HJ3    ,6HJ4    ,6HLREF
     I /
                             DATA MOTIC2
     1 /6HLREFY ,6HMU    ,6HOMEGA ,6HPGCLAT,6HPSL   ,6HPWPROP,6HRHOSL
     2 ,6HRN    ,6HRE    ,6HRP    ,6HSREF  ,6HTSL   ,6HWGTSG ,6HWJETT
     3 ,6HWPLD  ,6HWPROPI,6HWEICON,6HXREF  ,6HYREF  ,6HZREF  ,6HAEXP
     4 ,6HCINF  ,6HVINFI ,6HIENGA1,6HIENGA2,6HIENGA3,6HIENGA4,6HIENGA5
     5 ,6HIENGA6,6HIENGA7,6HIENGA8,6HIENGA9,6HIENGA1,6HIENGA1,6HIENGA1
     6 ,6HIENGA1,6HIENGA1,6HIENGA1,6HIENGT1,6HIENGT2,6HIENGT3,6HIENGT4
     7 ,6HIENGT5,6HIENGT6,6HIENGT7,6HIENGT8,6HIENGT9,6HIENGT1,6HIENGT1
     8 ,6HIENGT1,6HIENGT1,6HIENGT1,6HIENGT1,6HITAP1 ,6HITAP2 ,6HITAP3
     9 ,6HITAP4 ,6HITAP5 ,6HITAP6 ,6HITAP7 ,6HITAP8 ,6HITAP9 ,6HITAP10
     0 ,6HIWPF1 ,6HIWPF2 ,6HIWPF3 ,6HIWPF4 ,6HIWPF5 ,6HIWPF6 ,6HIWPF7
     A ,6HIWPF8 ,6HIWPF9 ,6HIWPF10,6HIWPF11,6HIWPF12,6HIWPF13,6HIWPF14
     B ,6HIWPF15,6HNENG  ,6HNEQS1 ,6HNEQS2 ,6HNEQS3 ,6HNEWSTG,6HNPC1
     C ,6HNPC2  ,6HNPC3  ,6HNPC4  ,6HNPC5  ,6HNPC6  ,6HNPC7  ,6HNPC8
     D ,6HNPC9  ,6HNPC10 ,6HNPC11 ,6HNPC12 ,6HNPC13 ,6HNPC14 ,6HNPC15
     E ,6HNPC16 ,6HNPC17 ,6HNPC18 ,6HNPC19 ,6HNPC20 ,6HNPC21 ,6HNPC22
     F ,6HNPC23 ,6HNPC24 ,6HNPC25 ,6HNPC26 ,6HNPC27 ,6HNPC28 ,6HNPC29
     G ,6HNPC30 ,6HNPC31 ,6HNPC32 ,6HNPC33 ,6HNPC34 ,6HNPC35 ,6HNPC36
     H ,6HGHA   ,6HGHAS  ,6HDECL  ,6HTRPM  ,6HDVMARR,6HDATE1 ,6HDATE2
     I /
                             DATA MOTIC3
     1 /6HDATE3
     I /
C
C
C
                             DATA MOTVC1
     1 /6HAHI   ,6HAHID  ,6HAMXB  ,6HAMYB  ,6HAMZB  ,6HASM   ,6HASMG
     2 ,6HATEM  ,6HAXB   ,6HAYB   ,6HAZB   ,6HASXI  ,6HASYI  ,6HASZI
     3 ,6HAXI   ,6HAYI   ,6HAZI   ,6HAXL1  ,6HAXL2  ,6HAXL3  ,6HCA
     4 ,6HCN    ,6HCD    ,6HCL    ,6HCDDP  ,6HCDDY  ,6HCLDP  ,6HCM
     5 ,6HCMDP  ,6HCADP  ,6HCNDP  ,6HCWDY  ,6HCADY  ,6HCYDY  ,6HCY
     6 ,6HCW    ,6HCIP1  ,6HCIP2  ,6HCIP3  ,6HCIP4  ,6HCIP5  ,6HCIP6
     7 ,6HCIP7  ,6HCIP8  ,6HCIP9  ,6HCIP10 ,6HCIP11 ,6HCIP12 ,6HCIP13
     8 ,6HCIP14 ,6HCIP15 ,6HCIY1  ,6HCIY2  ,6HCIY3  ,6HCIY4  ,6HCIY5
     9 ,6HCIY6  ,6HCIY7  ,6HCIY8  ,6HCIY9  ,6HCIY10 ,6HCIY11 ,6HCIY12
     0 ,6HCIY13 ,6HCIY14 ,6HCIY15 ,6HCS    ,6HDAX   ,6HDAY   ,6HDAZ
     A ,6HDE0   ,6HDE1   ,6HDE2   ,6HDE3   ,6HDFVAL1,6HDFVAL2,6HDFVAL3
     B ,6HDFLP  ,6HDFLY  ,6HDFVLH1,6HDFVLH2,6HDFVLH3,6HGAMAD ,6HAZVAD
     C ,6HDMASS ,6HHEATRT,6HDUA   ,6HDVA   ,6HDWA   ,6HVELAD ,6HDVWH1
     D ,6HDVWH2 ,6HDVWH3 ,6HDLRD  ,6HTVLRD ,6HATLD  ,6HGLRD  ,6HVIDLD
     E ,6HDVX   ,6HDVY   ,6HDVZ   ,6HDX    ,6HDY    ,6HDZ    ,6HE0
     F ,6HE1    ,6HE2    ,6HE3    ,6HETAL  ,6HFAXB  ,6HFAYB  ,6HFAZB
     G ,6HFMXB  ,6HFMYB  ,6HFMZB  ,6HFTXB  ,6HFTYB  ,6HFTZB  ,6HFVAL1
     H ,6HFVAL2 ,6HFVAL3 ,6HGAMMAA,6HGAMMAI,6HGAMMAR,6HGXI   ,6HGYI
     I /
                             DATA MOTVC2
     1 /6HGZI   ,6HH     ,6HHTBT  ,6HHTBTD ,6HHTLF  ,6HHTLFD ,6HHTTP
     2 ,6HHTTPD ,6HHTRT  ,6HHTRTD ,6HHTURB ,6HHTURBD,6HAZVELA,6HAZVELI
     3 ,6HAZVELR,6HGCLAT ,6HGDLAT ,6HLONG  ,6HLONGI ,6HMACH  ,6HMASS
     4 ,6HPJETTS,6HPRES  ,6HPWDOT ,6HDYNP  ,6HQALPHA,6HTLHEAT,6HGCRAD
     5 ,6HDENS  ,6HRSO   ,6HRS    ,6HSIP1  ,6HSIP2  ,6HSIP3  ,6HSIP4
     6 ,6HSIP5  ,6HSIP6  ,6HSIP7  ,6HSIP8  ,6HSIP9  ,6HSIP10 ,6HSIP11
     7 ,6HSIP12 ,6HSIP13 ,6HSIP14 ,6HSIP15 ,6HSIY1  ,6HSIY2  ,6HSIY3
     8 ,6HSIY4  ,6HSIY5  ,6HSIY6  ,6HSIY7  ,6HSIY8  ,6HSIY9  ,6HSIY10
     9 ,6HSIY11 ,6HSIY12 ,6HSIY13 ,6HSIY14 ,6HSIY15 ,6HTHRUST,6HTIME
     0 ,6HTMXB  ,6HTMYB  ,6HTMZB  ,6HTTMXB ,6HTTMYB ,6HTTMZB ,6HTVAC
     A ,6HU     ,6HV     ,6HW     ,6HUA    ,6HVA    ,6HWA    ,6HVELA
     B ,6HVAXI  ,6HVAYI  ,6HVAZI  ,6HUB    ,6HVB    ,6HWB    ,6HVELI
     C ,6HDLR   ,6HTVLR  ,6HATL   ,6HGLR   ,6HVIDEAL,6HUR    ,6HVR
     D ,6HWR    ,6HVELR  ,6HVRXI  ,6HVRYI  ,6HVRZI  ,6HUW    ,6HVW
     E ,6HWW    ,6HVWXI  ,6HVWYI  ,6HVWZI  ,6HVXI   ,6HVYI   ,6HVZI
     F ,6HVXL   ,6HVYL   ,6HVZL   ,6HVXLO  ,6HVYLO  ,6HVZLO  ,6HWDOT
     G ,6HWEIGHT,6HWJETTM,6HWPROP ,6HXCG   ,6HYCG   ,6HZCG   ,6HXI
     H ,6HYI    ,6HZI    ,6HXL    ,6HYL    ,6HZL    ,6HXLO   ,6HYLO
     I /
                             DATA MOTVC3
     1 /6HZLO   ,6HYAWRH ,6HDCLV  ,6HDCDV  ,6HVINV  ,6HAE1   ,6HAE2
     2 ,6HAE3   ,6HAE4   ,6HAE5   ,6HAE6   ,6HAE7   ,6HAE8   ,6HAE9
     3 ,6HAE10  ,6HAE11  ,6HAE12  ,6HAE13  ,6HAE14  ,6HAE15  ,6HWD1
     4 ,6HWD2   ,6HWD3   ,6HWD4   ,6HWD5   ,6HWD6   ,6HWD7   ,6HWD8
     5 ,6HWD9   ,6HWD10  ,6HWD11  ,6HWD12  ,6HWD13  ,6HWD14  ,6HWD15
     6 ,6HTHR1  ,6HTHR2  ,6HTHR3  ,6HTHR4  ,6HTHR5  ,6HTHR6  ,6HTHR7
     7 ,6HTHR8  ,6HTHR9  ,6HTHR10 ,6HTHR11 ,6HTHR12 ,6HTHR13 ,6HTHR14
     8 ,6HTHR15 ,6HISV   ,6HISV3  ,6HDIARP1,6HDIARP2,6HDIARP3,6HDRAGP1
     9 ,6HDRAGP2,6HDRAGP3,6HDRAGPT,6HFAXBP1,6HFAXBP2,6HFAXBP3,6HCDP1
     0 ,6HCDP2  ,6HCDP3  ,6HDIAMP1,6HDIAMP2,6HDIAMP3,6HDRGPK1,6HDRGPK2
     A ,6HDRGPK3,6HDRGPP1,6HDRGPP2,6HDRGPP3,6HVELAP ,6HDRGPS1,6HDRGPS2
     B ,6HDRGPS3,6HIDRGP1,6HIDRGP2,6HIDRGP3,6HPARIF1,6HPARIF2,6HPARIF3
     C ,6HDLID  ,6HTVLID ,6HGLID  ,6HDLI   ,6HTVLI  ,6HGLI   ,6HHTRT1I
     D ,6HHTRT1 ,6HTIMEO ,6HTTIME ,6HAXG   ,6HAYG   ,6HAZG   ,6HAHORIZ
     E ,6HAVERT ,6HIAEROH,6HROVET1,6HROVET2,6HROVET3,6HROVET4,6HROVET5
     F ,6HROVET6,6HROVET7,6HROVET8,6HROVET9,6HROVET0
     I /
C
C
C
                        DATA PHZVC1
     1 /6HALTMAX,6HALTMIN,6HMAXTIM,6HEVTF  ,6HFESN  ,6HIESN  ,6HPHZF
     2 ,6HPIF   ,6HI4    ,6HSAVESN
     3 /
C
C
C
                        DATA SPECA1
     1 /6HSPECI1,6HSPECI2,6HSPECI3,6HSPECI4,6HSPECI5,6HSPECI6,6HSPECI7
     2 ,6HSPECI8,6HSPECI9,6HSPECV1,6HSPECV2,6HSPECV3,6HSPECV4,6HSPECV5
     3 ,6HSPECV6,6HSPECV7,6HSPECV8,6HSPECV9,6HNSPEC1,6HNSPEC2,6HNSPEC3
     4 ,6HNSPEC4,6HNSPEC5,6HNSPEC6
     5 /
C
C
C
                        DATA TGOVC1
     1 /6HFUXN1 ,6HFUXN2 ,6HFUXN3 ,6HFUXN4 ,6HFUXN5 ,6HFUXN6 ,6HFUXN7
     2 ,6HFUXN8 ,6HFUXN9 ,6HFUXN10,6HPCTGO ,6HSAVE1 ,6HSAVE2 ,6HSAVE3
     3 ,6HSAVE4 ,6HSAVE5 ,6HSAVE6 ,6HSAVE7 ,6HSAVE8 ,6HSAVE9 ,6HSAVE10
     4 ,6HSAVE11,6HSAVE12,6HSAVE13,6HSAVE14,6HSAVE15,6HSAVE16,6HSAVE17
     5 ,6HSAVE18,6HSAVE19,6HSAVE20,6HSAVE21,6HSAVE22,6HSAVE23,6HSAVE24
     6 ,6HSAVE25,6HSAVE26,6HSAVE27,6HSAVE28,6HSAVE29,6HSAVE30,6HSAVE31
     7 ,6HSAVE32,6HSAVE33,6HSAVE34,6HSAVE35,6HSAVE36,6HSAVE37,6HSAVE38
     8 ,6HSAVE39,6HSAVE40,6HSAVE41,6HSAVE42,6HSAVE43,6HSAVE44,6HSAVE45
     9 ,6HSAVE46,6HSAVE47,6HSAVE48,6HSAVE49,6HSAVE50,6HSAVE51,6HSAVE52
     0 ,6HSAVE53,6HSAVE54,6HSAVE55,6HSAVE56,6HSAVE57,6HSAVE58,6HSAVE59
     A ,6HSAVE60,6HSAVE61,6HSAVE62,6HSAVE63,6HSAVE64,6HSAVE65,6HSAVE66
     B ,6HSAVE67,6HSAVE68,6HSAVE69,6HSAVE70,6HSAVE71,6HSAVE72,6HSAVE73
     C ,6HSAVE74,6HSAVE75,6HSAVE76,6HSAVE77,6HSAVE78,6HSAVE79,6HSAVE80
     D ,6HTGO   ,6HTIMX  ,6HESN
     C ,6HIEVNT1,6HIEVNT2,6HIEVNT3,6HIEVNT4,6HIEVNT5,6HIEVNT6,6HIEVNT7
     D ,6HIEVNT8,6HIEVNT9,6HIENV10,6HISZEV ,6HNXEVT ,6HI5    ,6HGUXN1
     E ,6HGUXN2 ,6HGUXN3 ,6HGUXN4 ,6HGUXN5 ,6HGUXN6 ,6HGUXN7 ,6HGUXN8
     F ,6HGUXN9 ,6HGUXN10,6HTIMY
     I /
C
C
C
                   DATA TRACK1
     1 /6HCTKLT1,6HCTKLT2,6HCTKLT3,6HCTKLT4,6HCTKLT5,6HCTKLT6,6HCTKLT7
     2 ,6HCTKLT8,6HCTKLT9,6HCTKLT0,6HCTKLN1,6HCKTLN2,6HCKTLN3,6HCKTLN4
     3 ,6HCTKLN5,6HCTKLN6,6HCTKLN7,6HCTKLN8,6HCTKLN9,6HCTKLN0,6HELEV1
     4 ,6HELEV2 ,6HELEV3 ,6HELEV4 ,6HELEV5 ,6HELEV6 ,6HELEV7 ,6HELEV8
     5 ,6HELEV9 ,6HELEV10,6HLKA1  ,6HLKA2  ,6HLKA3  ,6HLKA4  ,6HLKA5
     6 ,6HLKA6  ,6HLKA7  ,6HLKA8  ,6HLKA9  ,6HLKA10 ,6HLKB1  ,6HLKB2
     7 ,6HLKB3  ,6HLKB4  ,6HLKB5  ,6HLKB6  ,6HLKB7  ,6HLKB8  ,6HLKB9
     8 ,6HLKB10 ,6HPGT1  ,6HPGT2  ,6HPGT3  ,6HPGT4  ,6HPGT5  ,6HPGT6
     9 ,6HPGT7  ,6HPGT8  ,6HPGT9  ,6HPGT10 ,6HPGT11 ,6HPGT12 ,6HPGT13
     0 ,6HPGT14 ,6HPGT15 ,6HPGT16 ,6HPGT17 ,6HPGT18 ,6HPGT19 ,6HPGT20
     A ,6HPGT21 ,6HPGT22 ,6HPGT23 ,6HPGT24 ,6HPGT25 ,6HPGT26 ,6HPGT27
     B ,6HPGT28 ,6HPGT29 ,6HPGT30 ,6HPGT31 ,6HPGT32 ,6HPGT33 ,6HPGT34
     C ,6HPGT35 ,6HPGT36 ,6HPGT37 ,6HPGT38 ,6HPGT39 ,6HPGT40 ,6HPGT41
     D ,6HPGT42 ,6HPGT43 ,6HPGT44 ,6HPGT45 ,6HPGT46 ,6HPGT47 ,6HPGT48
     E ,6HPGT49 ,6HPGT50 ,6HPGT51 ,6HPGT52 ,6HPGT53 ,6HPGT54 ,6HPGT55
     F ,6HPGT56 ,6HPGT57 ,6HPGT58 ,6HPGT59 ,6HPGT60 ,6HPGT61 ,6HPGT62
     G ,6HPGT63 ,6HPGT64 ,6HPGT65 ,6HPGT66 ,6HPGT67 ,6HPGT68 ,6HPGT69
     H ,6HPGT70 ,6HPGT71 ,6HPGT72 ,6HPGT73 ,6HPGT74 ,6HPGT75 ,6HPGT76
     I /
C
                   DATA TRACK2
     1 /6HPGT77 ,6HPGT78 ,6HPGT79 ,6HPGT80 ,6HPGT81 ,6HPGT82 ,6HPGT83
     2 ,6HPGT84 ,6HPGT85 ,6HPGT86 ,6HPGT87 ,6HPGT88 ,6HPGT89 ,6HPGT90
     3 ,6HSLTRG1,6HSLTRG2,6HSLTRG3,6HSLTRG4,6HSLTRG5,6HSLTRG6,6HSLTRG7
     4 ,6HSLTRG8,6HSLTRG9,6HSLTRG0,6HSLOS11,6HSLOS12,6HSLOS13,6HSLOS14
     5 ,6HSLOS15,6HSLOS16,6HSLOS17,6HSLOS18,6HSLOS19,6HSLOS10,6HSLOS21
     6 ,6HSLOS22,6HSLOS23,6HSLOS24,6HSLOS25,6HSLOS26,6HSLOS27,6HSLOS28
     7 ,6HSLOS29,6HSLOS20,6HSLOS31,6HSLOS32,6HSLOS33,6HSLOS34,6HSLOS35
     8 ,6HSLOS36,6HSLOS37,6HSLOS38,6HSLOS39,6HSLOS30,6HSTKLT1,6HSTKLT2
     9 ,6HSTKLT3,6HSTKLT4,6HSTKLT5,6HSTKLT6,6HSTKLT7,6HSTKLT8,6HSTKLT9
     0 ,6HSTKLT0,6HSTKLN1,6HSTKLN2,6HSTKLN3,6HSTKLN4,6HSTKLN5,6HSTKLN6
     A ,6HSTKLN7,6HSTKLN8,6HSTKLN9,6HSTKLN0,6HTKLTC1,6HTKLTC2,6HTKLTC3
     B ,6HTKLTC4,6HTKLTC5,6HTKLTC6,6HTKLTC7,6HTKLTC8,6HTKLTC9,6HTKLTC0
     C ,6HTKRAD1,6HTKRAD2,6HTKRAD3,6HTKRAD4,6HTKRAD5,6HTKRAD6,6HTKRAD7
     D ,6HTKRAD8,6HTKRAD9,6HTKRAD0,6HTKAZM1,6HTKAZM2,6HTKAZM3,6HTKAZM4
     E ,6HTKAZM5,6HTKAZM6,6HTKAZM7,6HTKAZM8,6HTKAZM9,6HTKAZM0,6HTRKLT1
     F ,6HTRKLT2,6HTRKLT3,6HTRKLT4,6HTRKLT5,6HTRKLT6,6HTRKLT7,6HTRKLT8
     G ,6HTRKLT9,6HTRKLT0,6HTRKHT1,6HTRKHT2,6HTRKHT3,6HTRKHT4,6HTRKHT5
     H ,6HTRKHT6,6HTRKHT7,6HTRKHT8,6HTRKHT9,6HTRKHT0,6HTRKLN1,6HTRKLN2
     I /
C
                   DATA TRACK3
     1 /6HTRKLN3,6HTRKLN4,6HTRKLN5,6HTRKLN6,6HTRKLN7,6HTRKLN8,6HTRKLN9
     2 ,6HTRKLN0,6HTKRX1 ,6HTKRX2 ,6HTKRX3 ,6HTKRX4 ,6HTKRX5 ,6HTKRX6
     3 ,6HTKRX7 ,6HTKRX8 ,6HTKRX9 ,6HTKRX10,6HTKRX11,6HTKRX12,6HTKRX12
     4 ,6HTKRX13,6HTKRX14,6HTKRX15,6HTKRX16,6HTKRX17,6HTKRX18,6HTKRX19
     5 ,6HTKRX20,6HTKRX21,6HTKRX22,6HTKRX23,6HTKRX24,6HTKRX25,6HTKRX26
     6 ,6HTKRX27,6HTKRX28,6HTKRX29,6HTKRX30,6HTRKFL1,6HTRKFL2,6HTRKFL3
     7 ,6HTRKFL5,6HTRKFL6,6HTRKFL7,6HTRKFL8,6HTRKFL9,6HTRKFL0,6HTRKNM1
     8 ,6HTRKNM2,6HTRKNM3,6HTRKNM4,6HTRKNM5,6HTRKNM6,6HTRKNM7,6HTRKNM8
     9 ,6HTRKNM9,6HTRKNM0,6HNTRKS ,6HNDUM   ,6HFLEMIN
     I /
C
C
C
C
                                 DATA GUIDI1
     1/6HDTG    ,6HGVRI1 ,6HGVRI2 ,6HGVRI3 ,6HGVRI4 ,6HGVRI5 ,6HGVRI6
     2,6HGVRI7 ,6HGVRI8 ,6HGVRI9 ,6HGVRI10,6HIGF1  ,6HIGF2  ,6HIGF3
     3,6HIGF4  ,6HIGF5  ,6HIGF6
     4/
                                 DATA GUIDV1
     1/6HGTIME ,6HDGTIME,6HGPXI  ,6HGPYI  ,6HGPZI  ,6HGPVXI ,6HGPVYI
     2,6HGPVZI ,6HGPAXI ,6HGPAYI ,6HGPAZI ,6HGALPHA,6HGBETA ,6HGBANK
     3,6HGYAWR ,6HGPITR ,6HGROLR ,6HGROLI ,6HGYAWI ,6HGPITI ,6HGASM
     4,6HGPASYI,6HGPASYI,6HGPASZI,6HGTHRST,6HGWGT  ,6HGWDOT ,6HGVRC1
     5,6HGVRC2 ,6HGVRC3 ,6HGVRC4 ,6HGVRC5 ,6HGVRC6 ,6HGVRC7 ,6HGVRC8
     6,6HGVRC9 ,6HGVRC10
     7/
C
C
C
                                 DATA TARGV1
     1/6HALTAT ,6HALTPT ,6HANGMOT,6HAPORT ,6HARGPT ,6HARGVT ,6HDECLT
     2,6HECCENT,6HENRGYT,6HHYPVT ,6HINCT  ,6HLANT  ,6HPERIDT,6HPGERT
     3,6HPGLONT,6HPGCLTT,6HRTASCT,6HSEMAXT,6HTRUANT,6HTRUMXT,6HETR1
     4,6HETR2  ,6HETR3  ,6HETR4  ,6HETR5  ,6HETR6  ,6HETR7  ,6HETR8
     5,6HETR9  ,6HDXRT  ,6HDYRT  ,6HDZRT  ,6HDVXRT ,6HDVYRT ,6HDVZRT
     6,6HDXI   ,6HDYI   ,6HDZI   ,6HDVXI  ,6HDVYI  ,6HDVZI  ,6HDAXI
     7,6HDAYI  ,6HDAZI  ,6HXIT   ,6HYIT   ,6HZIT   ,6HVXIT  ,6HVYIT
     8,6HVZIT  ,6HAXIT  ,6HAYIT  ,6HAZIT  ,6HGCRADT,6HVELIT ,6HTDXR
     9,6HTDYR  ,6HTDZR  ,6HTDVXR ,6HTDVYR ,6HTDVZR ,6HDVCIRT,6HECCANT
     0,6HMEAANT,6HTIMSPT,6HTIMTPT,6HPGVELT,6HAPVELT,6HGTAXI ,6HGTAYI
     A,6HGTAZI ,6HGTVXI ,6HGTVYI ,6HGTVZI ,6HGTXI  ,6HGTYI  ,6HGTZI
     B,6HMVEHF1,6HMVEHF2,6HMVEHF3,6HMVEHF4,6HMVEHF5,6HMVEHF6,6HMVEHF7
     C,6HMVEHF8,6HMVEHF9,6HMVEHF0,6HAZVIT ,6HGAMIT ,6HIGT1  ,6HIGT2
     D,6HIGT3  ,6HIGT4  ,6HIGT5  ,6HIGT6  ,6HIGT7  ,6HIGT8  ,6HIGT9
     E,6HDRT   ,6HVCIRCT,6HLANVET,6HGCLATT,6HLONGT ,6HXLTO  ,6HYLTO
     F,6HZLTO  ,6HVXLTO ,6HVYLTO ,6HVZLTO ,6HXLT   ,6HYLT   ,6HZLT
     G,6HVXLT  ,6HVYLT  ,6HVZLT  ,6HAXLT  ,6HAYLT  ,6HAZLT  ,6HDXPT
     H,6HDYPT  ,6HDZPT  ,6HDVXPT ,6HDVYPT ,6HDVZPT ,6HDAXPT ,6HDAYPT
     I/
                                 DATA TARGV2
     1/6HDAZPT ,6HDVVXI ,6HDVVYI ,6HDVVZI ,6HDVVXIT,6HDVVYIT,6HDVVZIT
     I/
      END
```

DINPT: This routine reads the previously stored input data from the disc and locates the data for the current phase.


DINPT
READ THE INPUT DATA STORED ON TAPE 1 INTO IGEN
FIRST PHASE?
NO
YES
SET THIS EVENT NUMBER NEGATIVE SO TIME TO GO WILL NOT LOOK FOR THIS EVENT
SET INDEX FOR THE LOCATION OF THE NEXT EVENT NUMBER IN IGEN
NO
IS THIS THE DATA FOR THIS EVENT
YES
STORE THE DATA FOR THIS EVENT IN THE APPROPRIATE CORE LOCATIONS IN COMMON IV
RETURN

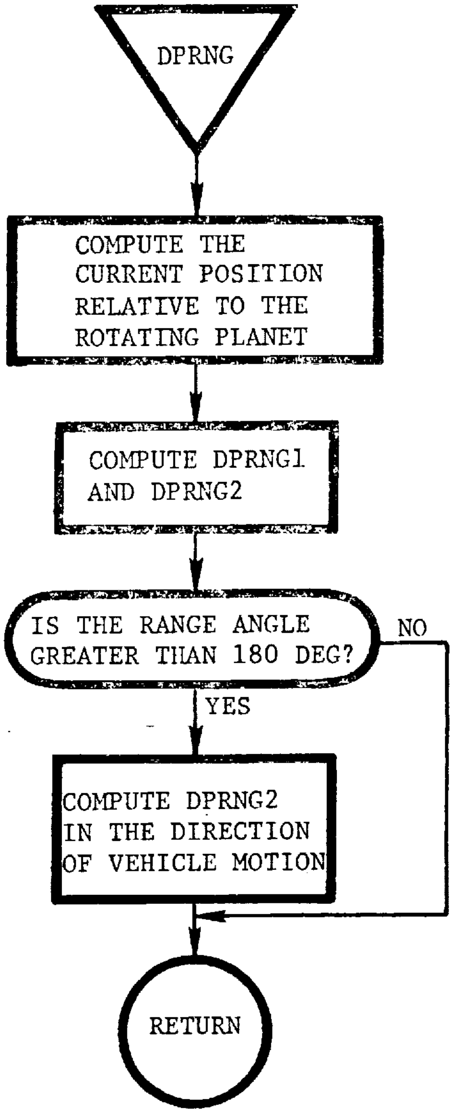
DLØØK (NAME): This routine sets the core addresses of the variables (NAME) to be integrated into the integration list.



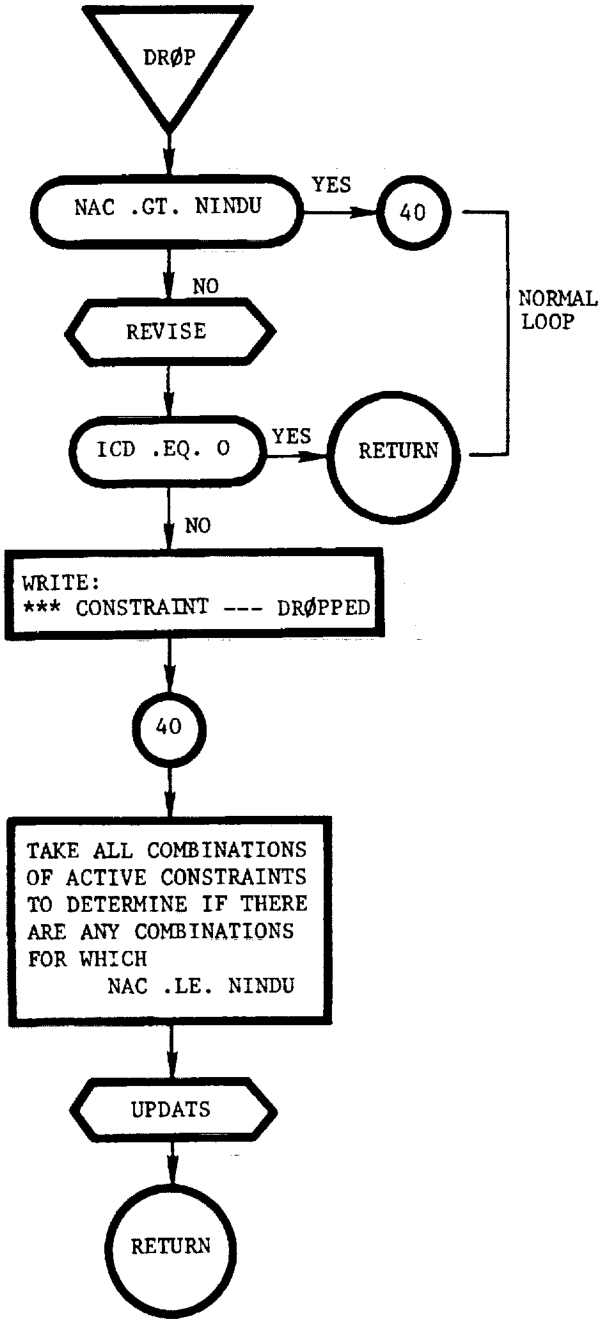
NOTE:

LISTI = INTEGRALS
LISTD = DERIVATIVES
N = NUMBER OF INTEGRALS

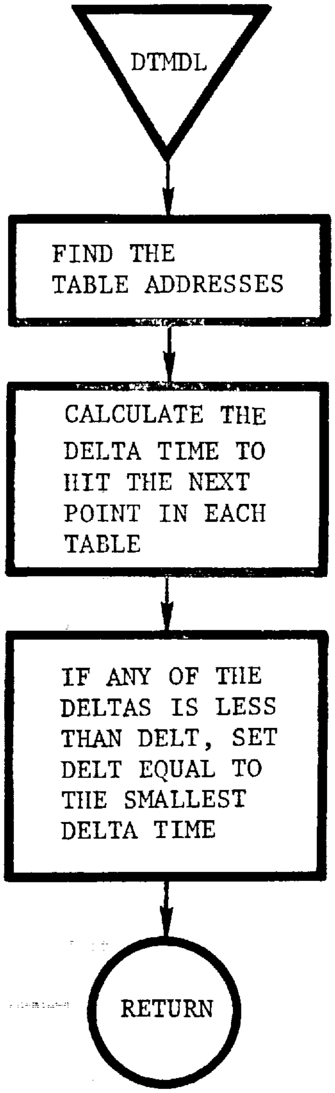
DPRNG: This routine calculates the range based on the dot product of the initial position vector of the vehicle and the impact point vector with the oblate planet.


DPRNG
COMPUTE THE CURRENT POSITION RELATIVE TO THE ROTATING PLANET
COMPUTE DPRNG1 AND DPRNG2
IS THE RANGE ANGLE GREATER THAN 180 DEG?
NO
YES
COMPUTE DPRNG2 IN THE DIRECTION OF VEHICLE MOTION
RETURN

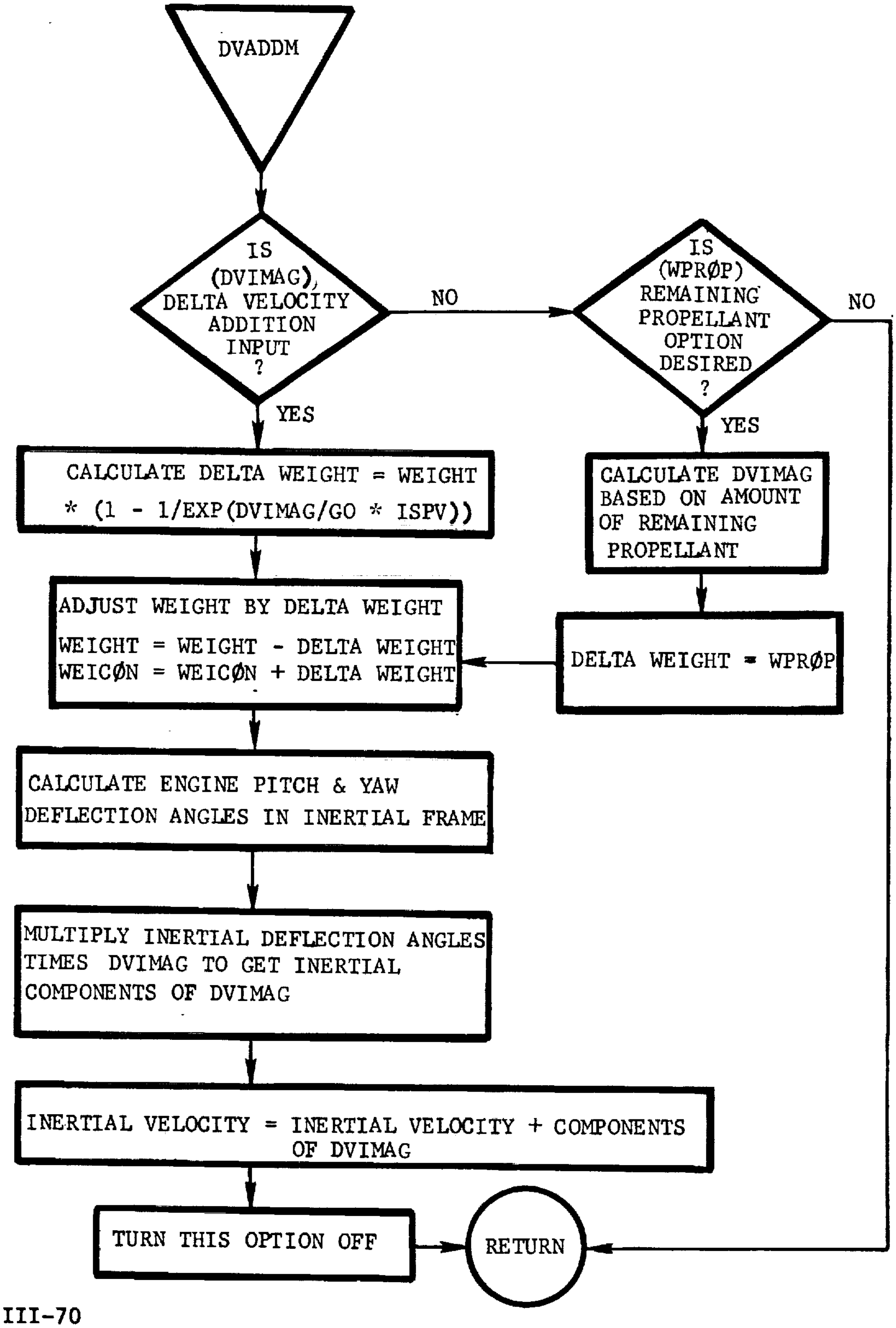
DRØP: This subroutine drops tight constraints.


DRØP
NAC .GT. NINDU
YES
40
NO
NORMAL LOOP
REVISE
ICD .EQ. 0
YES
RETURN
NO
WRITE:
*** CONSTRAINT --- DRØPPED
40
TAKE ALL COMBINATIONS OF ACTIVE CONSTRAINTS TO DETERMINE IF THERE ARE ANY COMBINATIONS FOR WHICH
NAC .LE. NINDU
UPDATS
RETURN

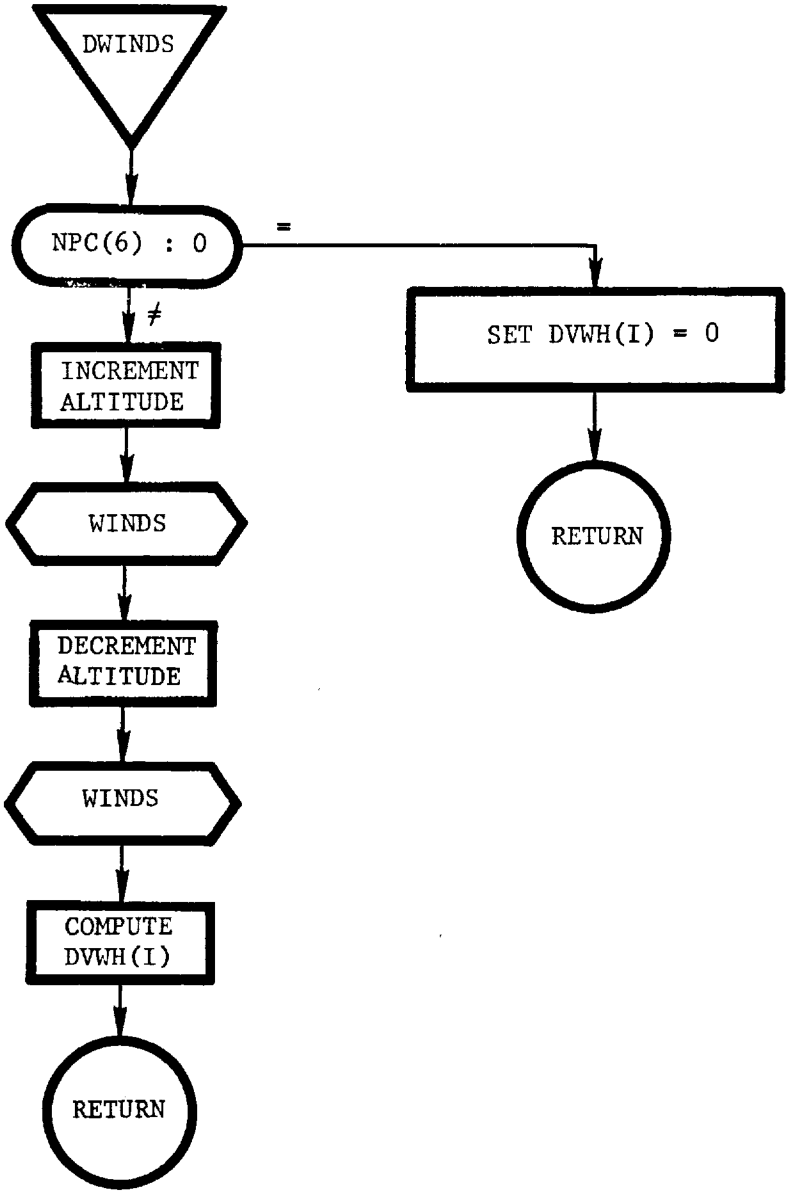
DTMDL: This routine checks the user-specified tables to ensure that the next integration step size is less than or equal to the next time point in any of the tables.


DTMDL
FIND THE TABLE ADDRESSES
CALCULATE THE DELTA TIME TO HIT THE NEXT POINT IN EACH TABLE
IF ANY OF THE DELTAS IS LESS THAN DELT, SET DELT EQUAL TO THE SMALLEST DELTA TIME
RETURN

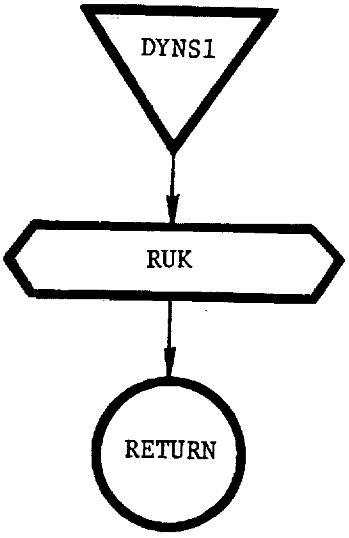
DVADDM: This routine adds instantaneous delta inertial velocity at the beginning of any phase.


DVADDM
IS (DVIMAG), DELTA VELOCITY ADDITION INPUT ?
NO
YES
IS (WPRØP) REMAINING PROPELLANT OPTION DESIRED ?
NO
YES
CALCULATE DELTA WEIGHT = WEIGHT * (1 - 1/EXP(DVIMAG/GO * ISPV))
CALCULATE DVIMAG BASED ON AMOUNT OF REMAINING PROPELLANT
ADJUST WEIGHT BY DELTA WEIGHT
WEIGHT = WEIGHT - DELTA WEIGHT
WEICØN = WEICØN + DELTA WEIGHT
DELTA WEIGHT = WPRØP
CALCULATE ENGINE PITCH & YAW DEFLECTION ANGLES IN INERTIAL FRAME
MULTIPLY INERTIAL DEFLECTION ANGLES TIMES DVIMAG TO GET INERTIAL COMPONENTS OF DVIMAG
INERTIAL VELOCITY = INERTIAL VELOCITY + COMPONENTS OF DVIMAG
TURN THIS OPTION OFF
RETURN

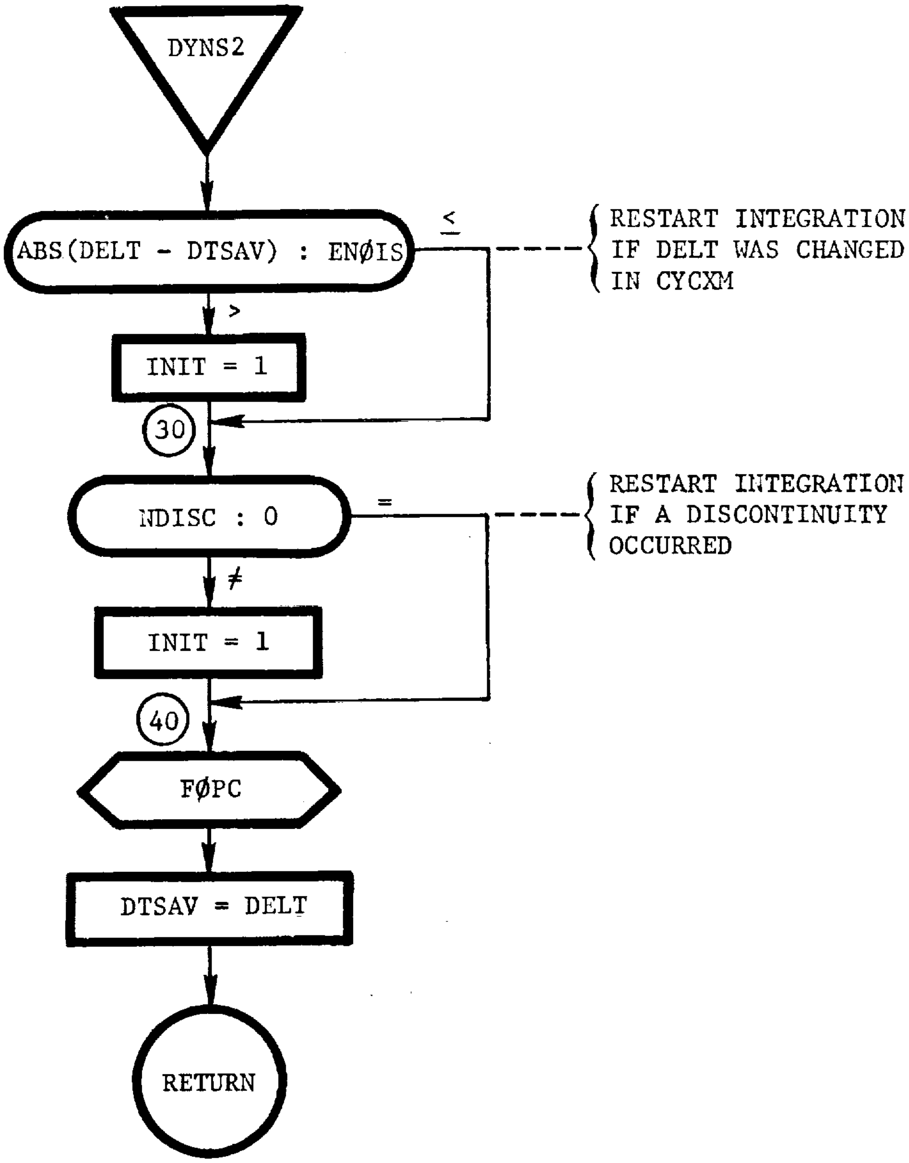
DWINDS: This routine calculates the rate of change in the wind with respect to the altitude above the surface.



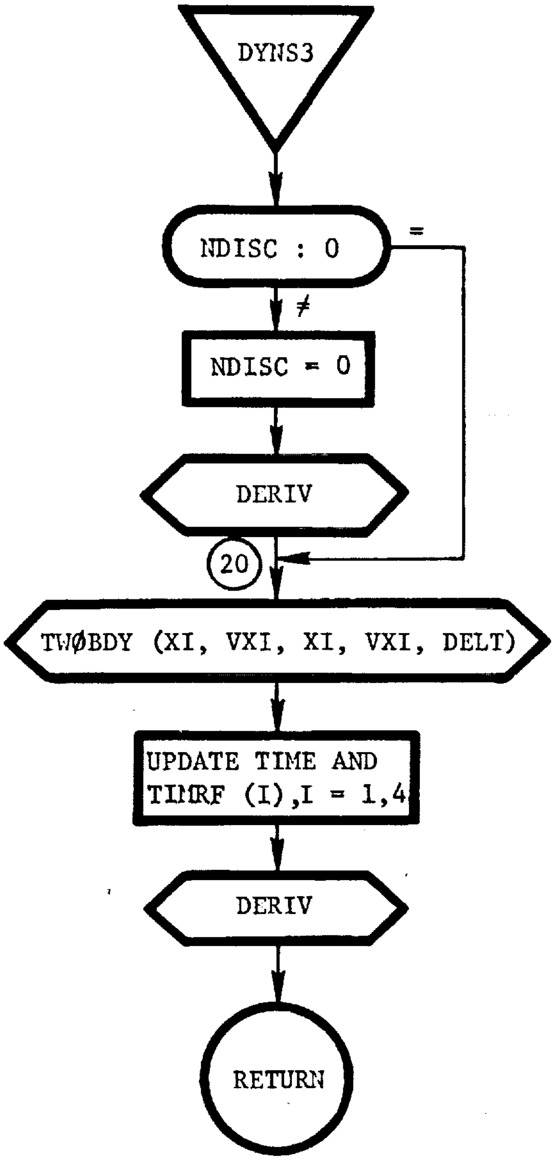
DYNS1: This routine integrates the equations of motion using the Runge-Kutta method.



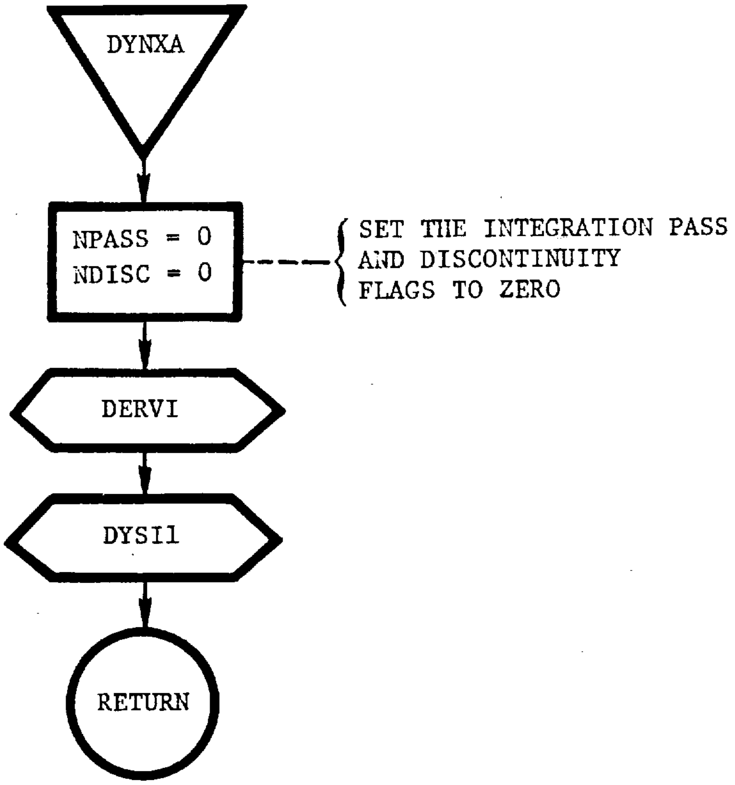
DYNS2: This routine integrates the equations of motion using the predictor-corrector method.


DYNS2
ABS(DELT - DTSAV) : ENØIS
≤
>
RESTART INTEGRATION IF DELT WAS CHANGED IN CYCXM
INIT = 1
30
NDISC : 0
=
≠
RESTART INTEGRATION IF A DISCONTINUITY OCCURRED
INIT = 1
40
FØPC
DTSAV = DELT
RETURN

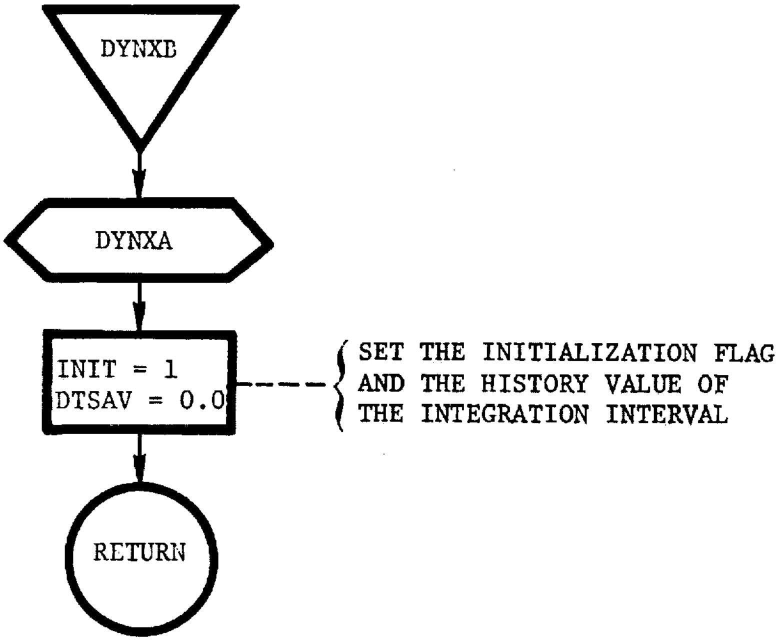
DYNS3: This routine integrates the equations of motion using Laplace's Method of integrating orbits about a spherical planet.


DYNS3
NDISC : 0
=
≠
NDISC = 0
DERIV
20
TWØBDY (XI, VXI, XI, VXI, DELT)
UPDATE TIME AND TIMRF (I),I = 1,4
DERIV
RETURN

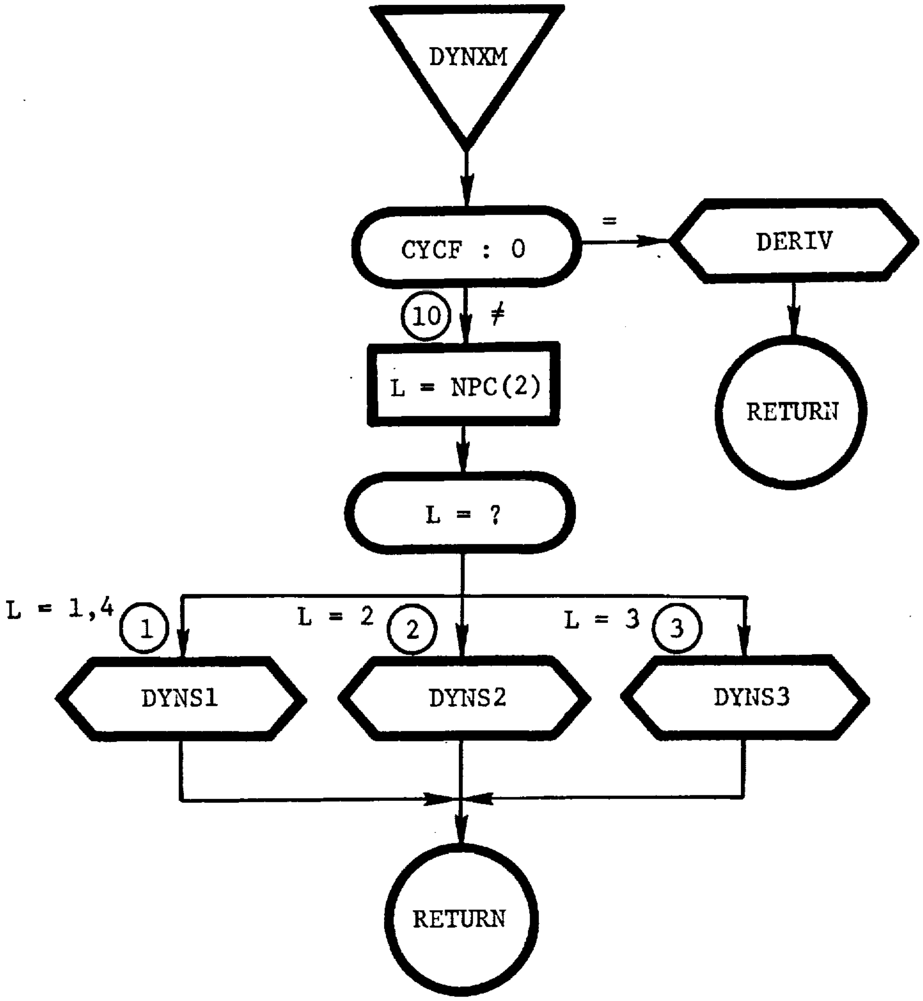
DYNXA: This routine initializes the fourth-order Runge-Kutta integration scheme.



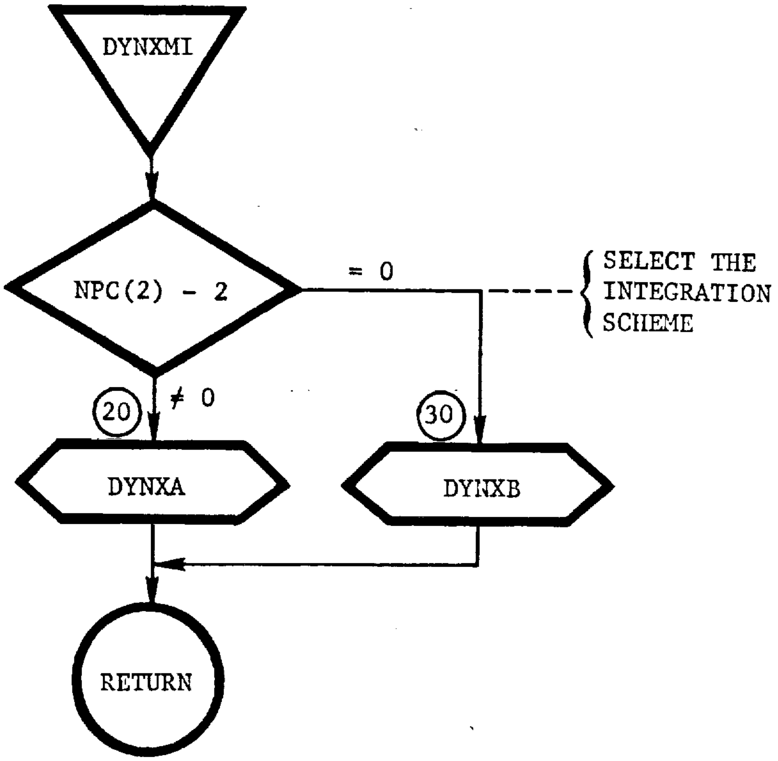
DYNXB: This routine initializes the fourth-order predictor-corrector integration scheme.


DYNXB
DYNXA
INIT = 1
DTSAV = 0.0
SET THE INITIALIZATION FLAG AND THE HISTORY VALUE OF THE INTEGRATION INTERVAL
RETURN

DYNXM: This routine determines which integration scheme is to be used.



DYNXMI: This routine selects the integration scheme to be used.

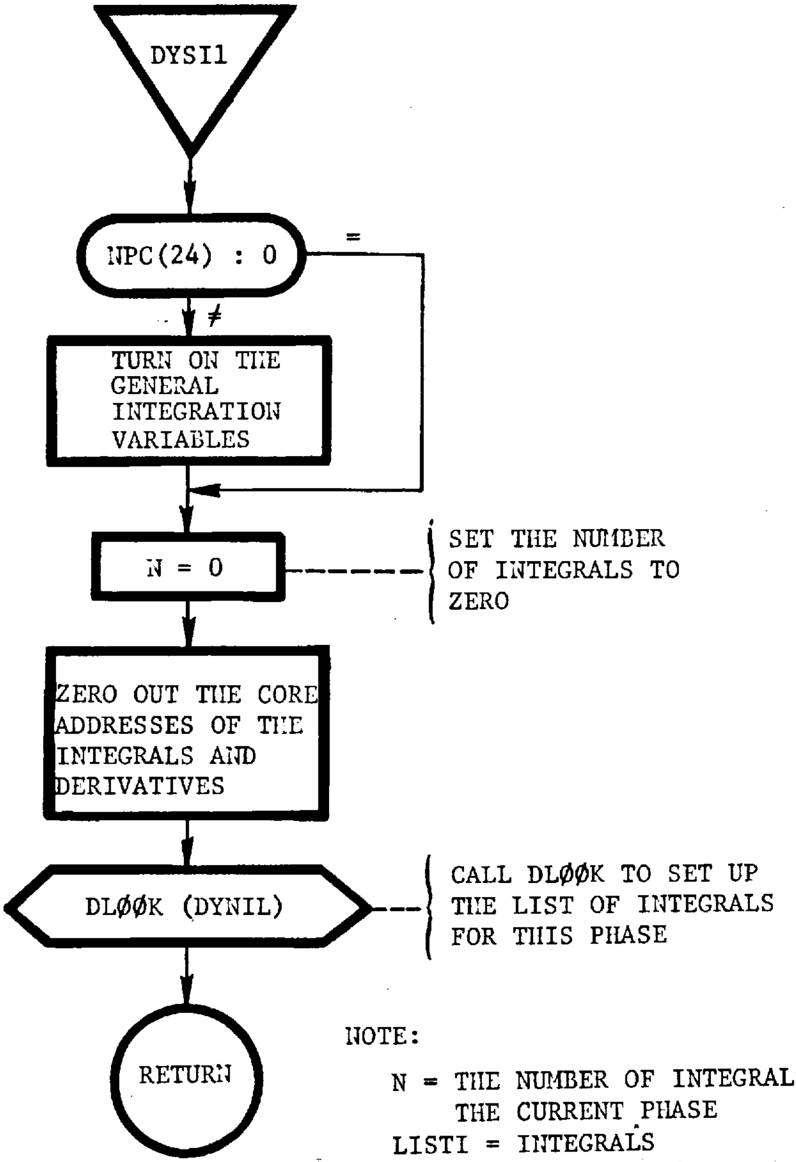


NOTE:

DYNXA - FOURTH-ORDER RUNGE-KUTTA INTEGRATION
- LAPLACE'S METHOD OF INTEGRATION
- ENCKE'S METHOD OF INTEGRATION

DYNXB - FOURTH-ORDER PREDICTOR-CORRECTOR INTEGRATION

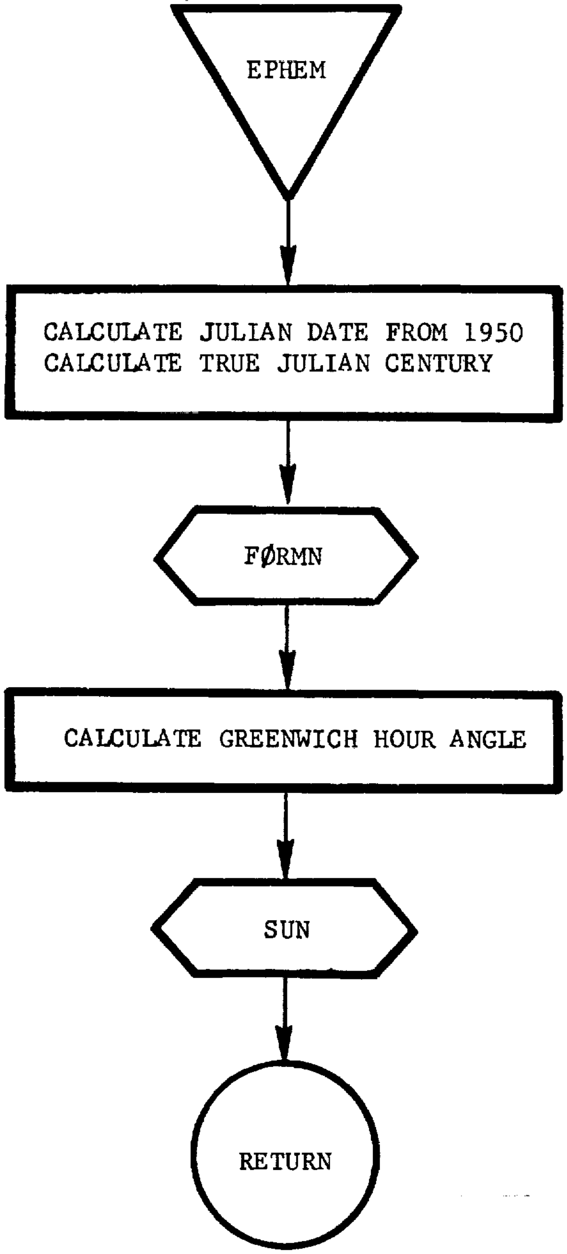
DYSI1: This routine sets the integration list.


DYSI1
NPC(24) : 0
=
≠
TURN ON THE GENERAL INTEGRATION VARIABLES
N = 0
SET THE NUMBER OF INTEGRALS TO ZERO
ZERO OUT THE CORE ADDRESSES OF THE INTEGRALS AND DERIVATIVES
DLØØK (DYNIL)
CALL DLØØK TO SET UP THE LIST OF INTEGRALS FOR THIS PHASE
RETURN

NOTE:

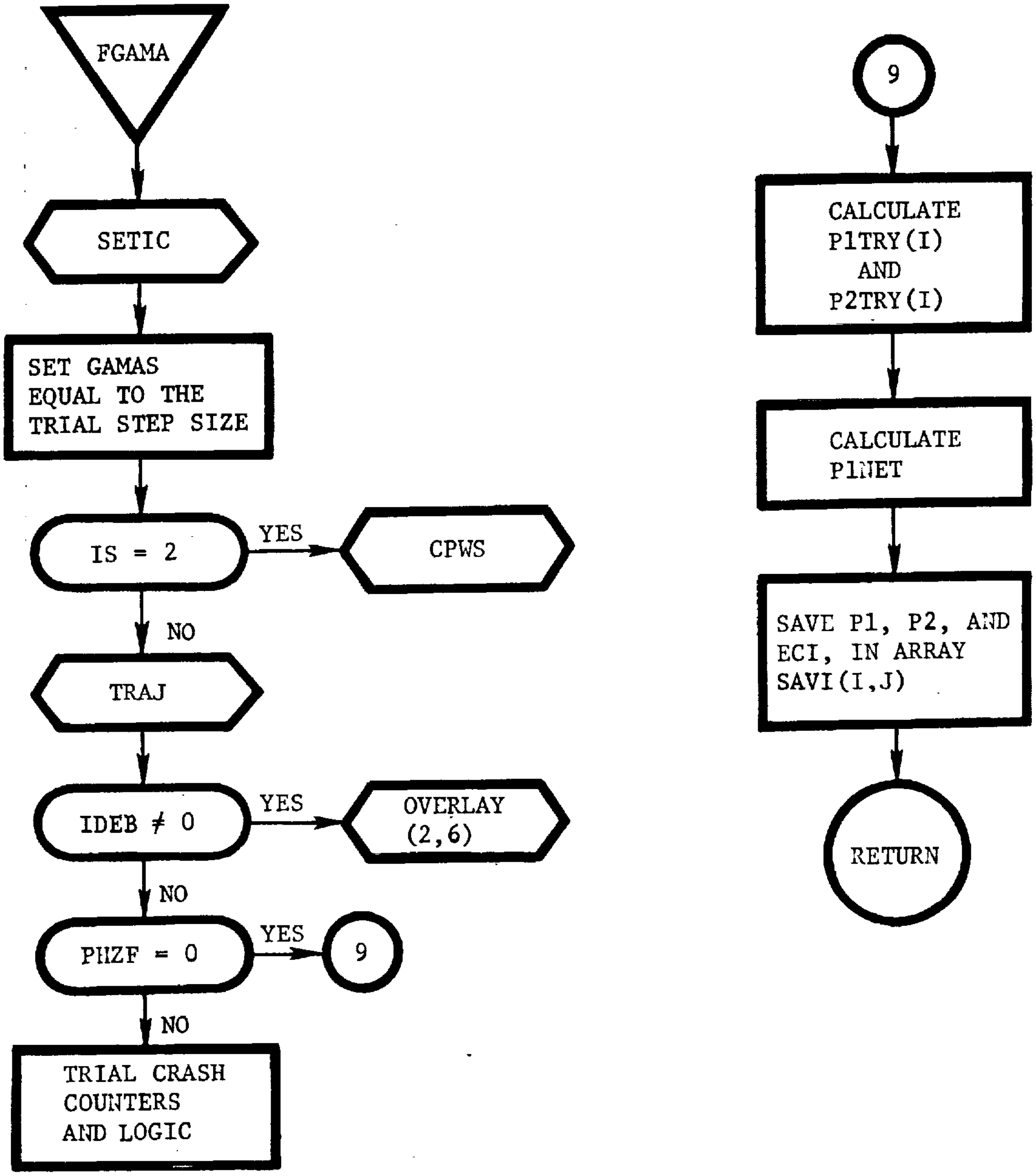
N = THE NUMBER OF INTEGRALS FOR
THE CURRENT PHASE
LISTI = INTEGRALS
LISTD = DERIVATIVES

EPHEM: This routine calculates the Greenwich Hour Angle and the right ascension and declination of the sun.

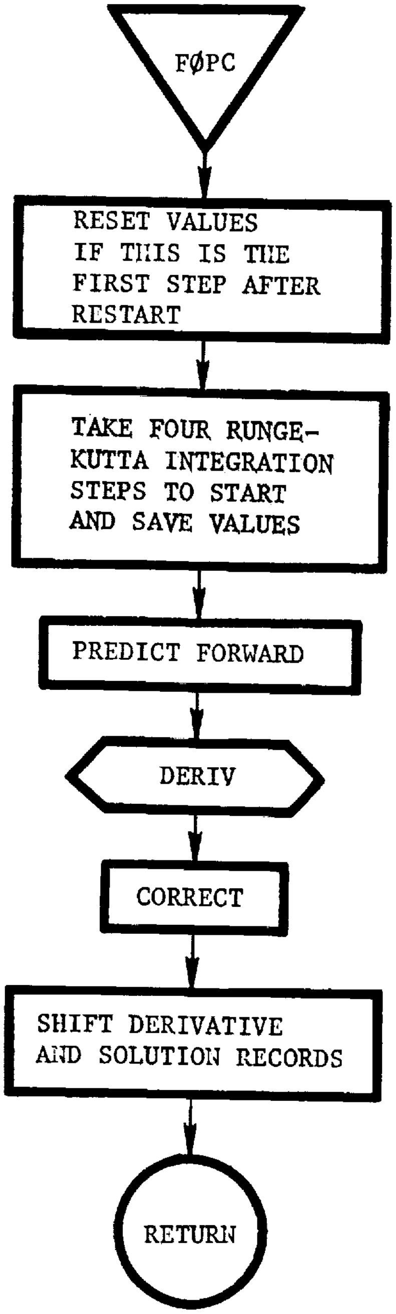

EPHEM
CALCULATE JULIAN DATE FROM 1950
CALCULATE TRUE JULIAN CENTURY
FØRMN
CALCULATE GREENWICH HOUR ANGLE
SUN
RETURN

FGAMA(IS): This subroutine calculates the values of P1 and P2 associated with a particular GAMA in the direction of search by changing the controls according to the equation

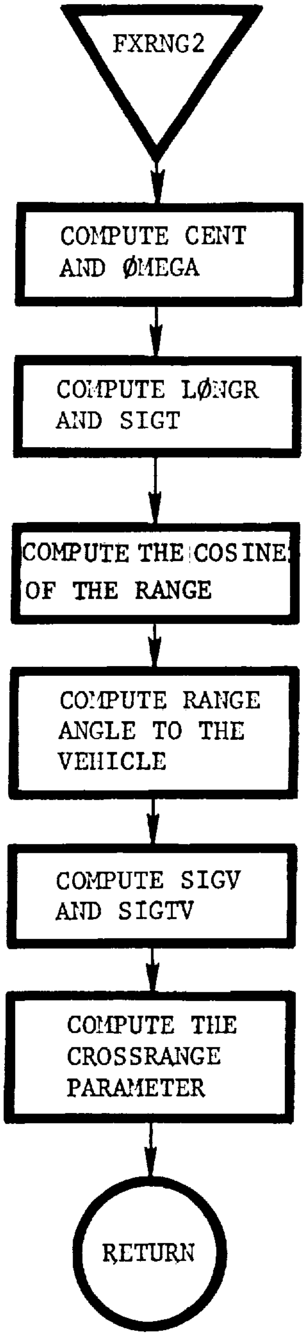
U(I) = U(I) + GAMA * DU(I)


FGAMA
SETIC
SET GAMAS EQUAL TO THE TRIAL STEP SIZE
IS = 2
YES
CPWS
NO
TRAJ
IDEB ≠ 0
YES
OVERLAY (2,6)
NO
PHZF = 0
YES
9
NO
TRIAL CRASH COUNTERS AND LOGIC
9
CALCULATE P1TRY(I) AND P2TRY(I)
CALCULATE P1NET
SAVE P1, P2, AND ECI, IN ARRAY SAVI(I,J)
RETURN

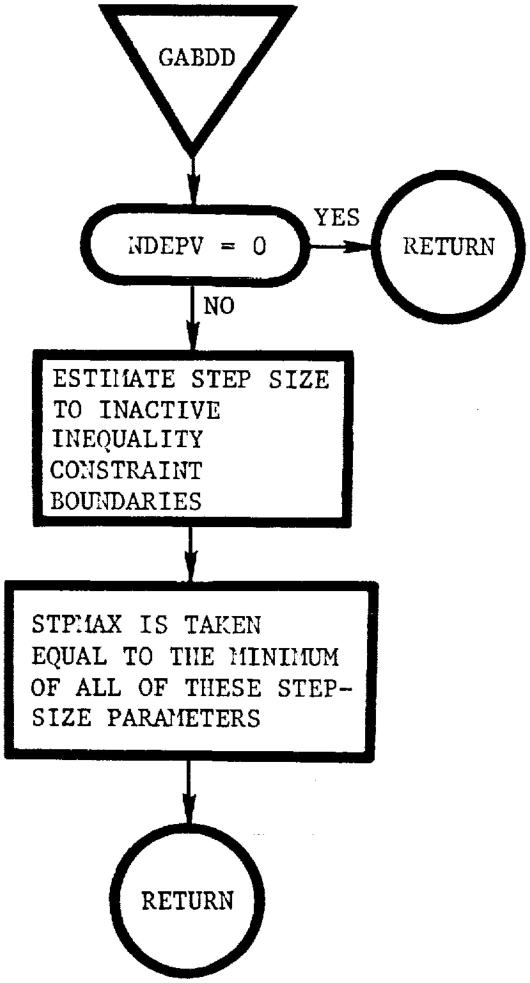
FØPC: This routine contains the fourth-order predictor-corrector integration algorithm.


FØPC
RESET VALUES IF THIS IS THE FIRST STEP AFTER RESTART
TAKE FOUR RUNGE-KUTTA INTEGRATION STEPS TO START AND SAVE VALUES
PREDICT FORWARD
DERIV
CORRECT
SHIFT DERIVATIVE AND SOLUTION RECORDS
RETURN

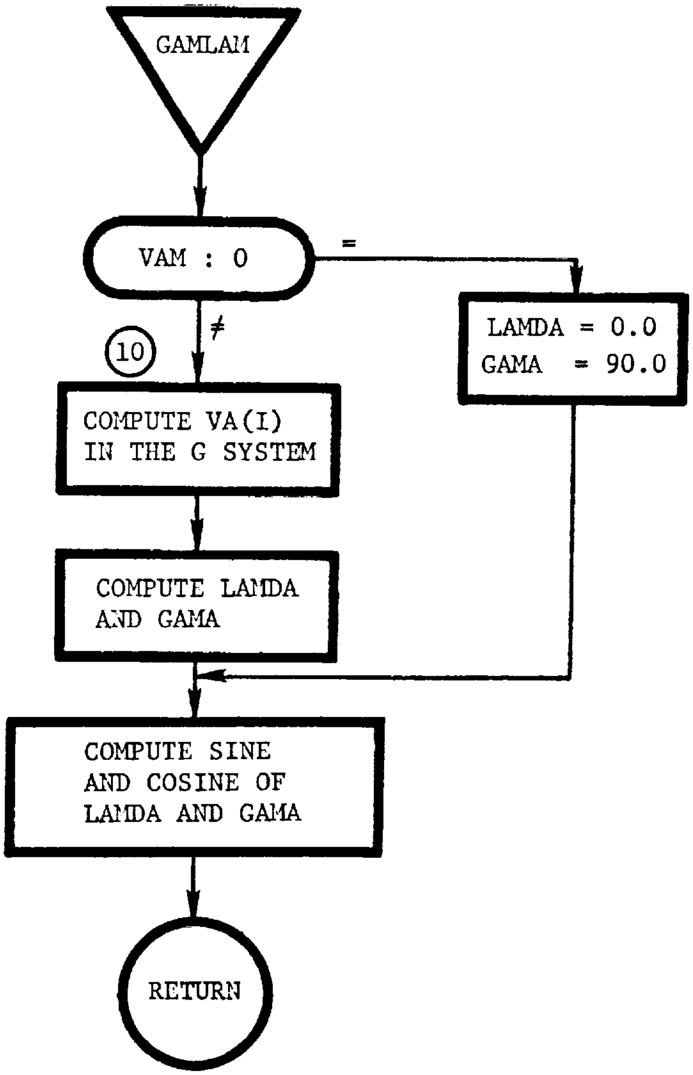
FXRNG2(T): This routine calculates the crossrange parameter used in the iteration scheme required to determine the crossrange relative to a reference ground track.


FXRNG2
COMPUTE CENT AND ØMEGA
COMPUTE LØNGR AND SIGT
COMPUTE THE COSINE OF THE RANGE
COMPUTE RANGE ANGLE TO THE VEHICLE
COMPUTE SIGV AND SIGTV
COMPUTE THE CROSSRANGE PARAMETER
RETURN

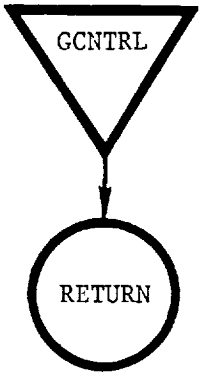
GABDD: This routine computes the maximum step size in the direction of search, STPMAX. STPMAX is computed as the minimum step needed to make one of the inactive inequality constraints become active. This prediction is based on a linearization of the inactive constraints.


GABDD
NDEPV = 0
YES
RETURN
NO
ESTIMATE STEP SIZE TO INACTIVE INEQUALITY CONSTRAINT BOUNDARIES
STPMAX IS TAKEN EQUAL TO THE MINIMUM OF ALL OF THESE STEP-SIZE PARAMETERS
RETURN

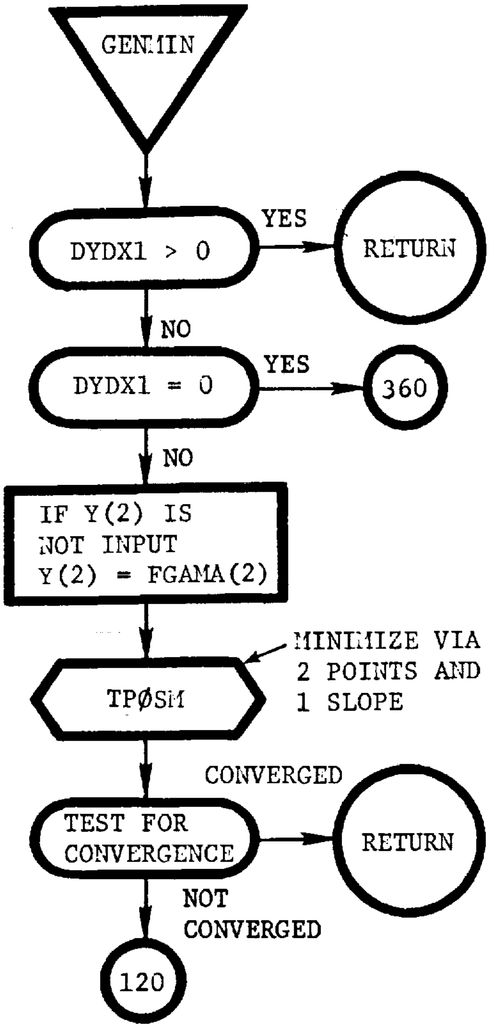
GAMLAM: This routine calculates the flight path angle and azimuth angle of the atmospheric relative velocity vector.

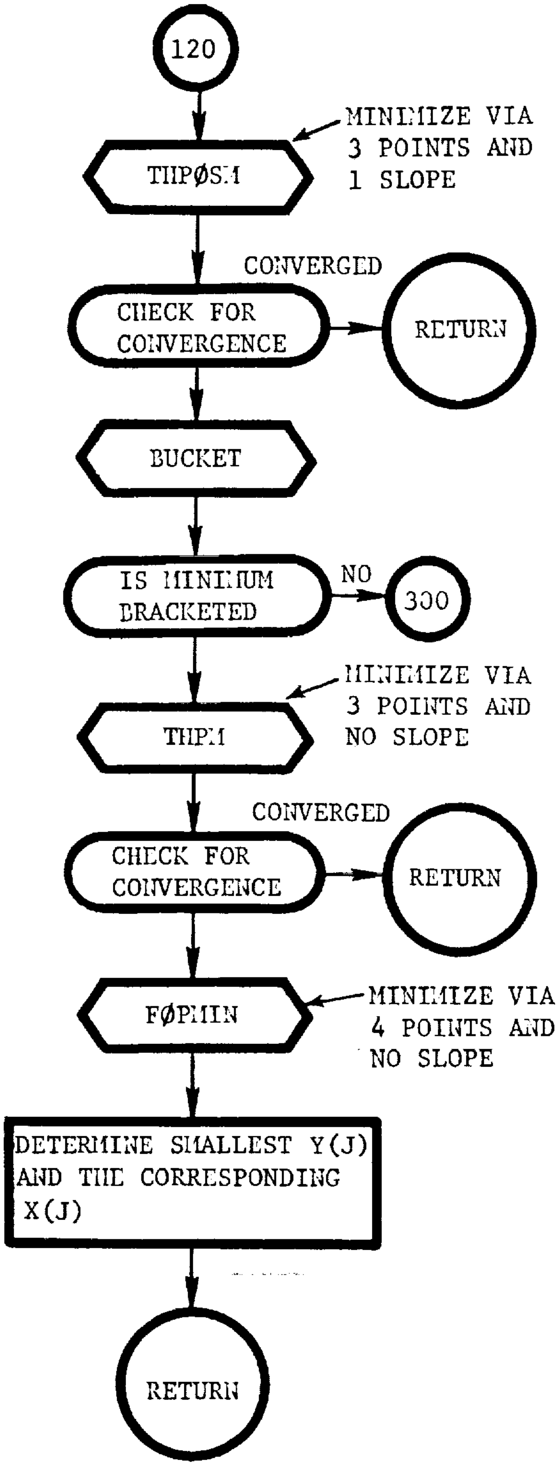

GAMLAM
VAM : 0
=
≠
10
LAMDA = 0.0
GAMA = 90.0
COMPUTE VA(I) IN THE G SYSTEM
COMPUTE LAMDA AND GAMA
COMPUTE SINE AND COSINE OF LAMDA AND GAMA
RETURN

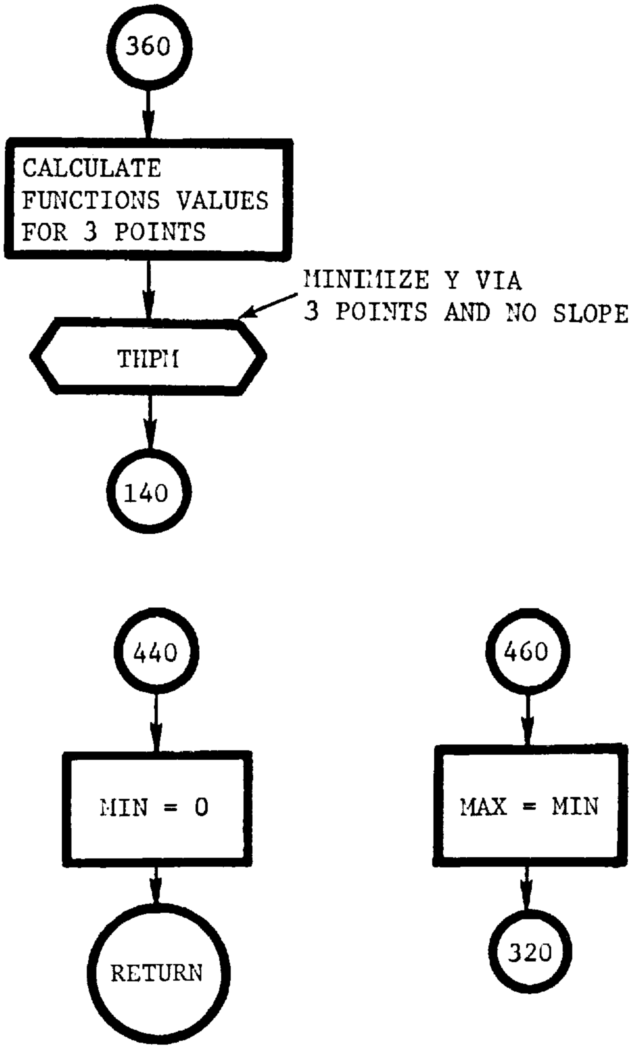
GCNTRL: This is a blank routine used for simulating the hardware lags and errors associated with implementing the closed-loop guidance steering commands.



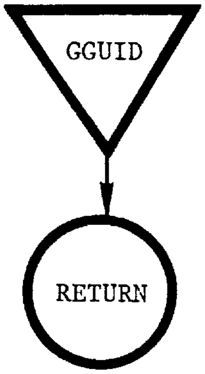
GENMIN: This routine finds the minimum of the function y, given the pairs (X, Y) and the slope at X = 0. If the pairs (X, Y) are not given, then GENMIN generates these pairs automatically. The minimization is based on approximations to the functions that are made using quadratic and cubic polynomials. The analytically calculated minimums of these polynomials are used as estimates to the actual minimum values of y.



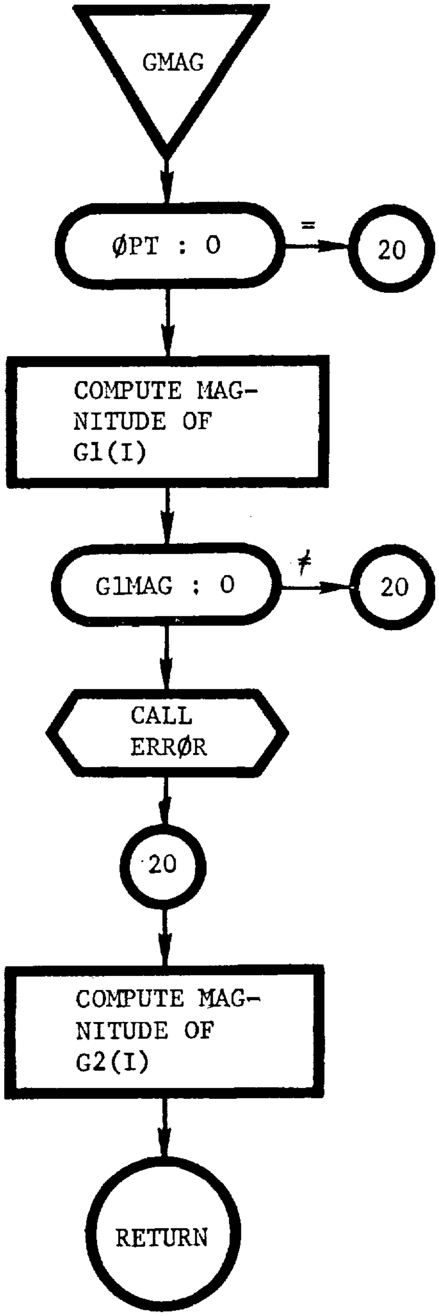

120
THPØSM
MINIMIZE VIA
3 POINTS AND
1 SLOPE
CONVERGED
CHECK FOR
CONVERGENCE
RETURN
BUCKET
IS MINIMUM
BRACKETED
NO
300
THPM
MINIMIZE VIA
3 POINTS AND
NO SLOPE
CONVERGED
CHECK FOR
CONVERGENCE
RETURN
FØPMIN
MINIMIZE VIA
4 POINTS AND
NO SLOPE
DETERMINE SMALLEST Y(J)
AND THE CORRESPONDING
X(J)
RETURN


360
CALCULATE FUNCTIONS VALUES FOR 3 POINTS
MINIMIZE Y VIA 3 POINTS AND NO SLOPE
THPM
140
440
MIN = 0
RETURN
460
MAX = MIN
320

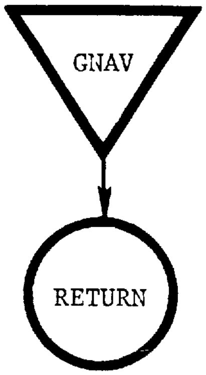
GGUID: This is a blank routine that is to be used for simulating the closed-loop guidance equations being analyzed.



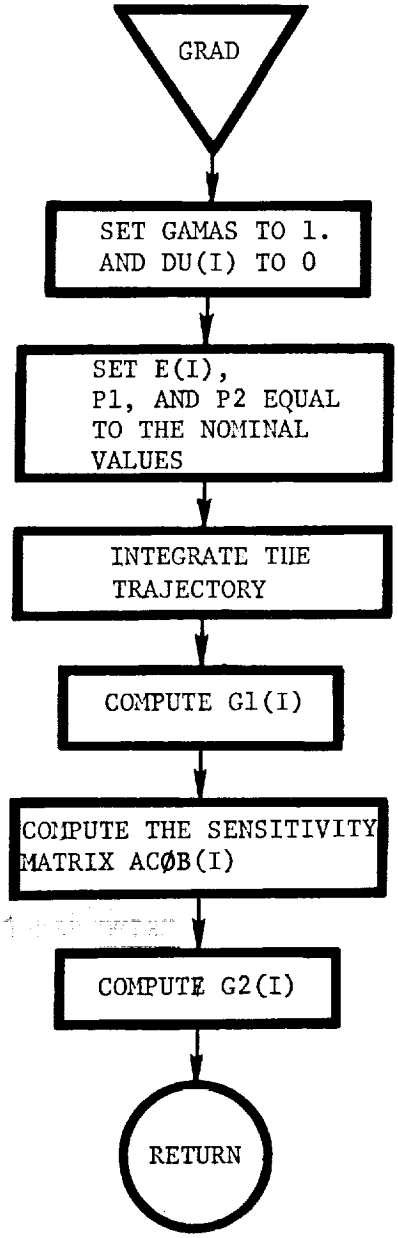
GMAG: This routine computes the magnitude of the gradient vectors G1(I) and G2(I).



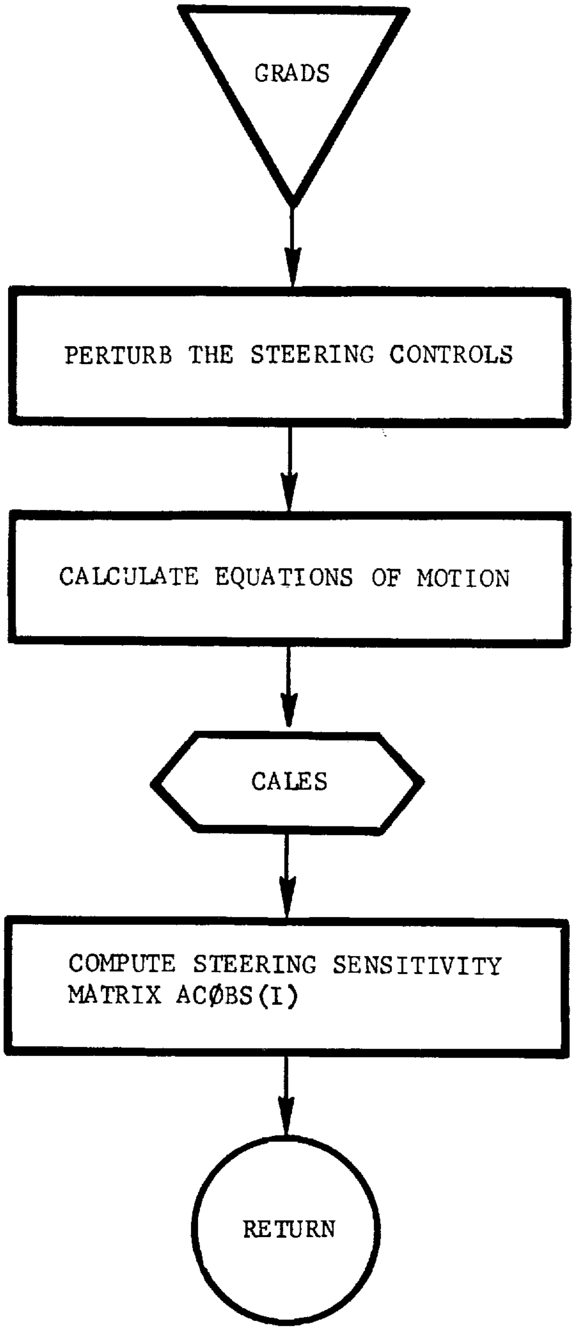
GNAV: This is a blank routine that is to be used for simulating the navigation equations for the closed-loop guidance system being analyzed.



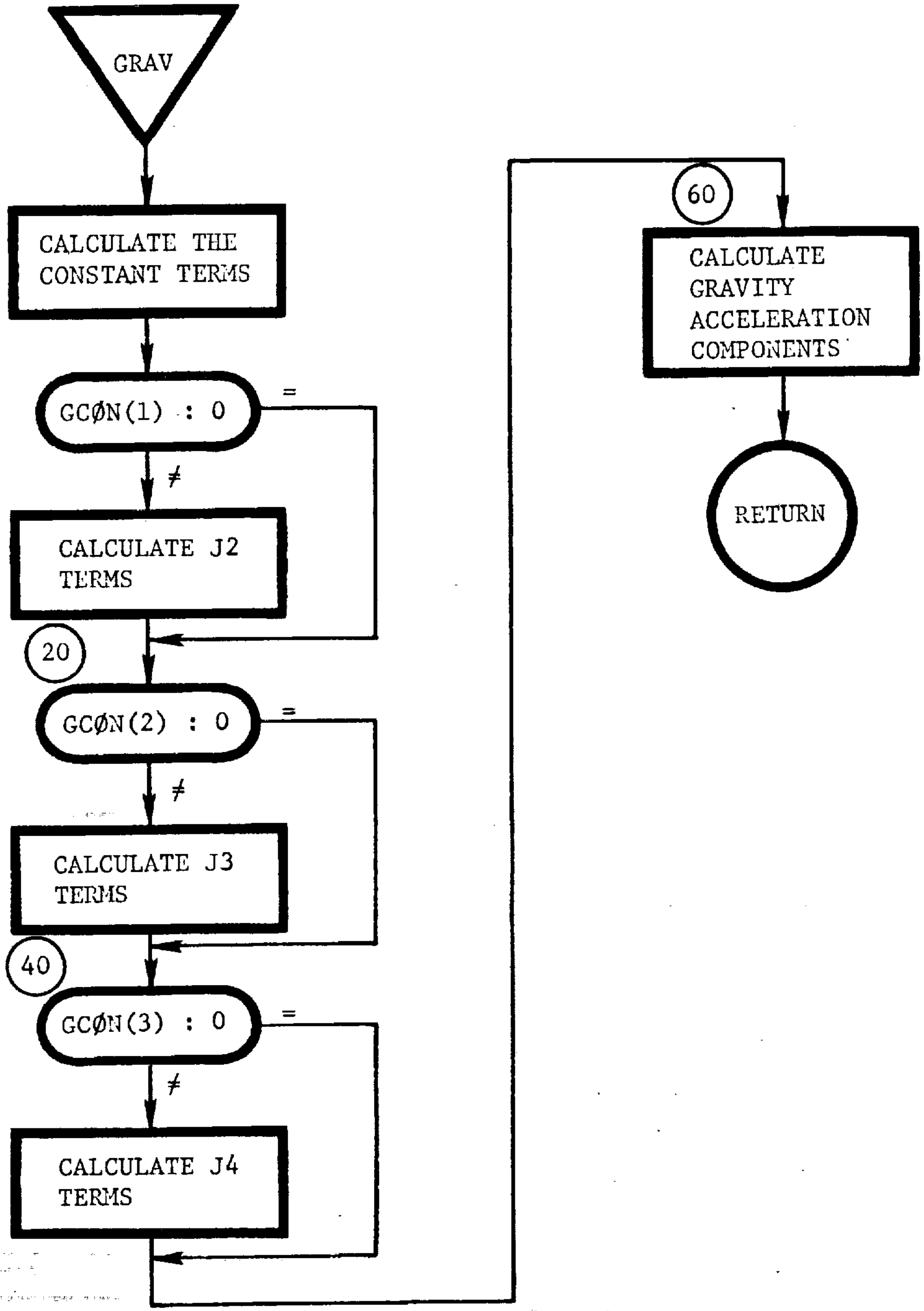
GRAD: This routine computes the gradients of the penalty functions (G1 and G2) with respect to the control parameters.


GRAD
SET GAMAS TO 1. AND DU(I) TO 0
SET E(I), P1, AND P2 EQUAL TO THE NOMINAL VALUES
INTEGRATE THE TRAJECTORY
COMPUTE G1(I)
COMPUTE THE SENSITIVITY MATRIX ACØB(I)
COMPUTE G2(I)
RETURN

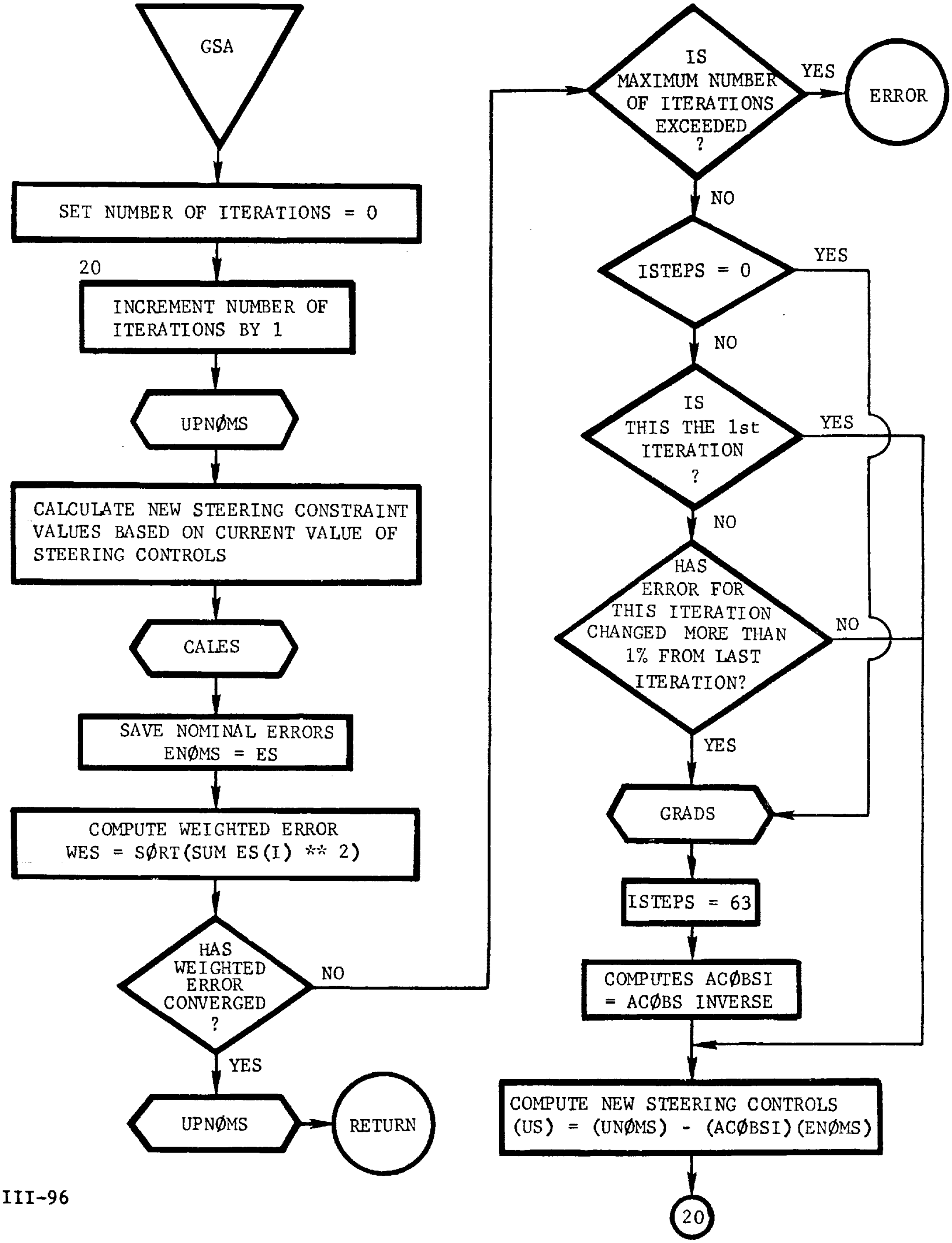
GRADS: This routine computes the gradients of the steering constraints with respect to the steering control parameters.


GRADS
PERTURB THE STEERING CONTROLS
CALCULATE EQUATIONS OF MOTION
CALES
COMPUTE STEERING SENSITIVITY MATRIX ACØBS(I)
RETURN

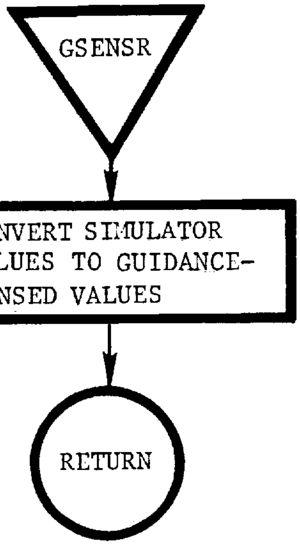
GRAV: This routine calculates the gravity acceleration vector.



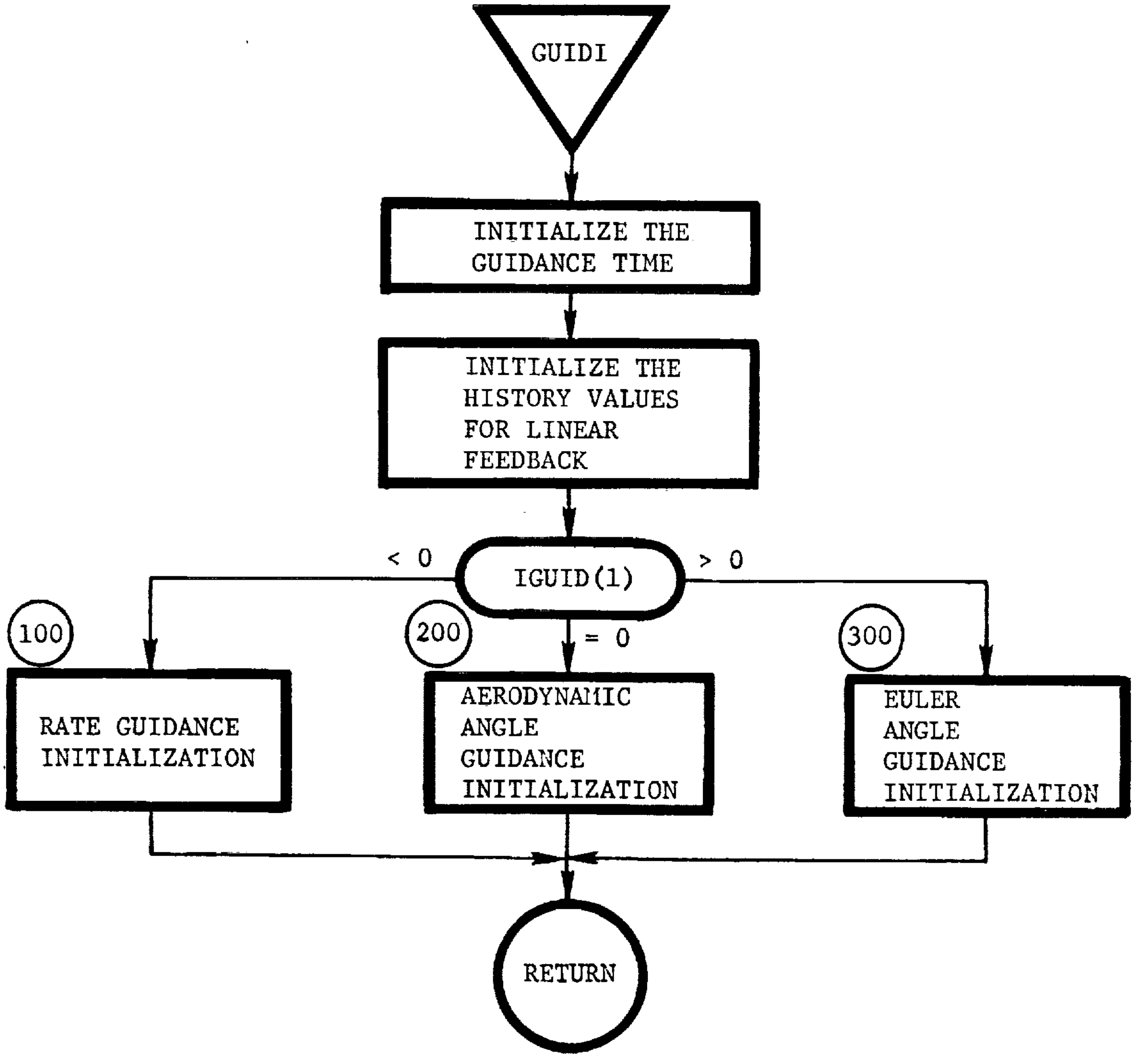
GSA: This routine (Generalized Steering Algorithm) computes the necessary values of steering control variables to yield the desired values of the steering constraints.


GSA
SET NUMBER OF ITERATIONS = 0
20
INCREMENT NUMBER OF ITERATIONS BY 1
UPNØMS
CALCULATE NEW STEERING CONSTRAINT VALUES BASED ON CURRENT VALUE OF STEERING CONTROLS
CALES
SAVE NOMINAL ERRORS ENØMS = ES
COMPUTE WEIGHTED ERROR WES = SØRT(SUM ES(I) ** 2)
HAS WEIGHTED ERROR CONVERGED ?
NO
YES
UPNØMS
RETURN
IS MAXIMUM NUMBER OF ITERATIONS EXCEEDED ?
YES
ERROR
NO
ISTEPS = 0
YES
NO
IS THIS THE 1st ITERATION ?
YES
NO
HAS ERROR FOR THIS ITERATION CHANGED MORE THAN 1% FROM LAST ITERATION?
NO
YES
GRADS
ISTEPS = 63
COMPUTES ACØBSI = ACØBS INVERSE
COMPUTE NEW STEERING CONTROLS (US) = (UNØMS) - (ACØBSI)(ENØMS)
20

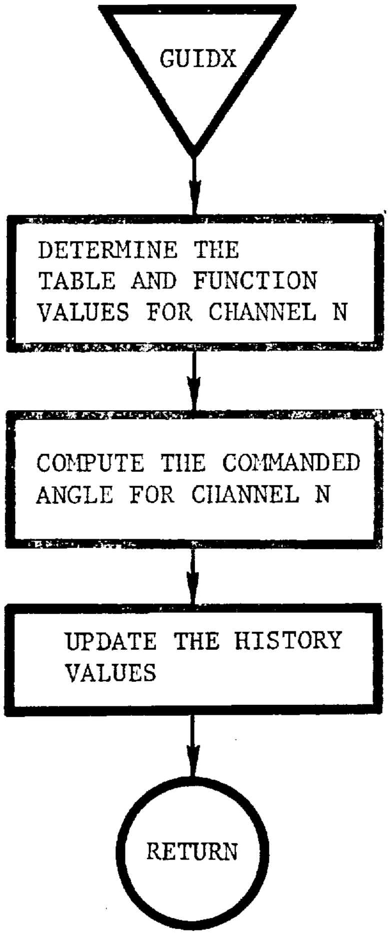
GSENSR: This routine is to be used for simulating the interface between the trajectory simulator (real world) and the guidance sensor (hardware detected) being analyzed.



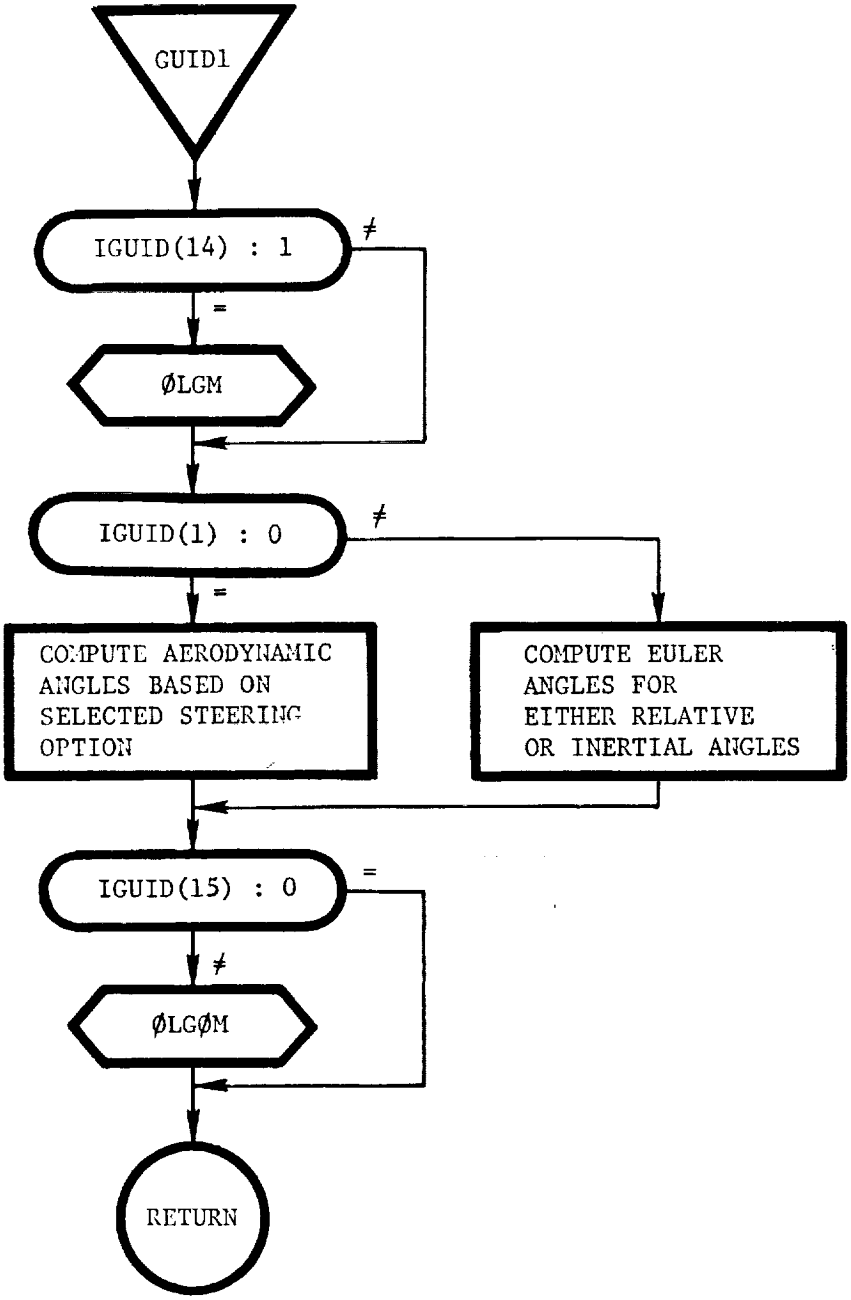
GUIDI: This routine initializes the guidance (steering) parameters based on the option selected by the user via the IGUID array.


GUIDI
INITIALIZE THE GUIDANCE TIME
INITIALIZE THE HISTORY VALUES FOR LINEAR FEEDBACK
IGUID(1)
< 0
> 0
= 0
100
200
300
RATE GUIDANCE INITIALIZATION
AERODYNAMIC ANGLE GUIDANCE INITIALIZATION
EULER ANGLE GUIDANCE INITIALIZATION
RETURN

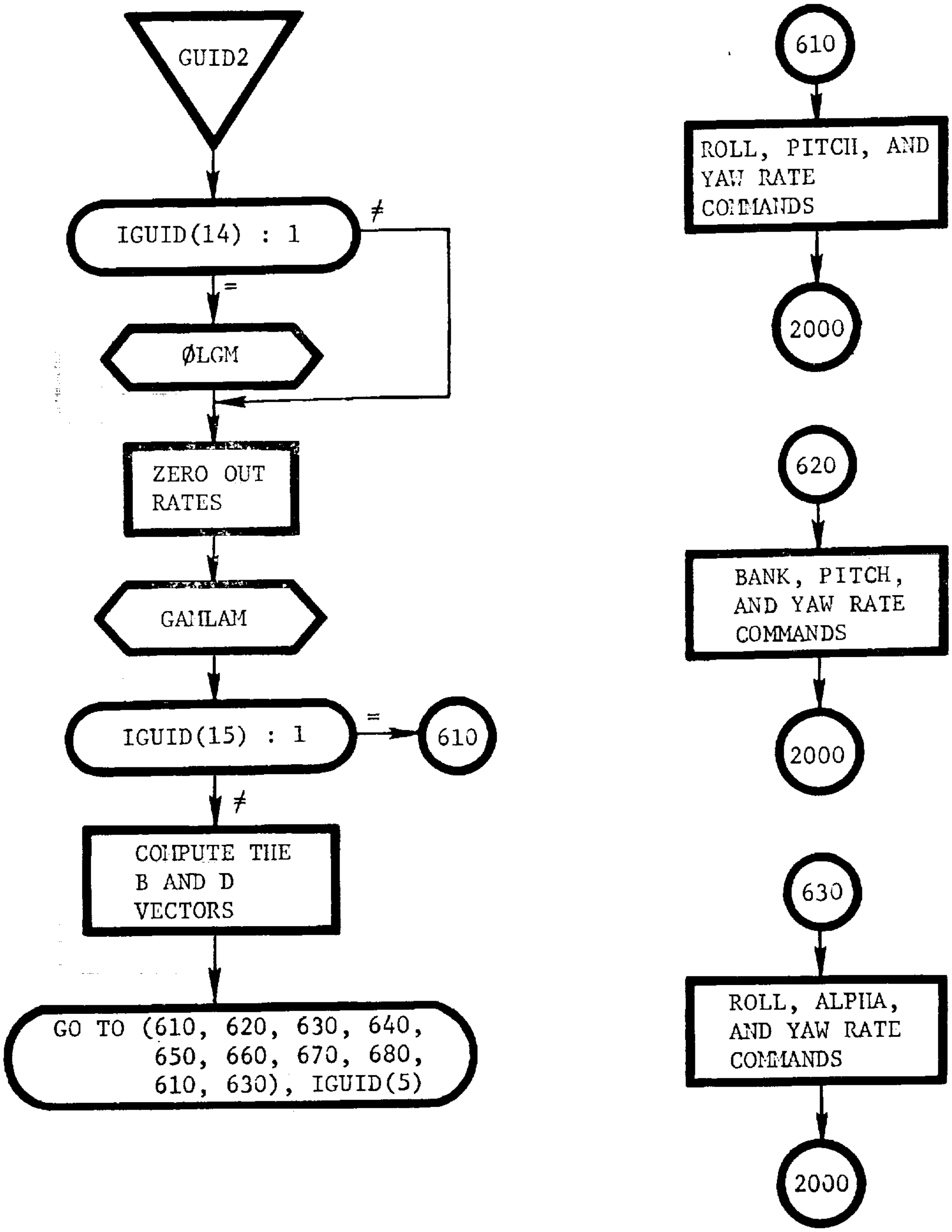
GUIDX (N): This function returns as the commanded angle for channel N, where N=1,2,3, based on the generalized linear feedback guidance (steering) algorithm.

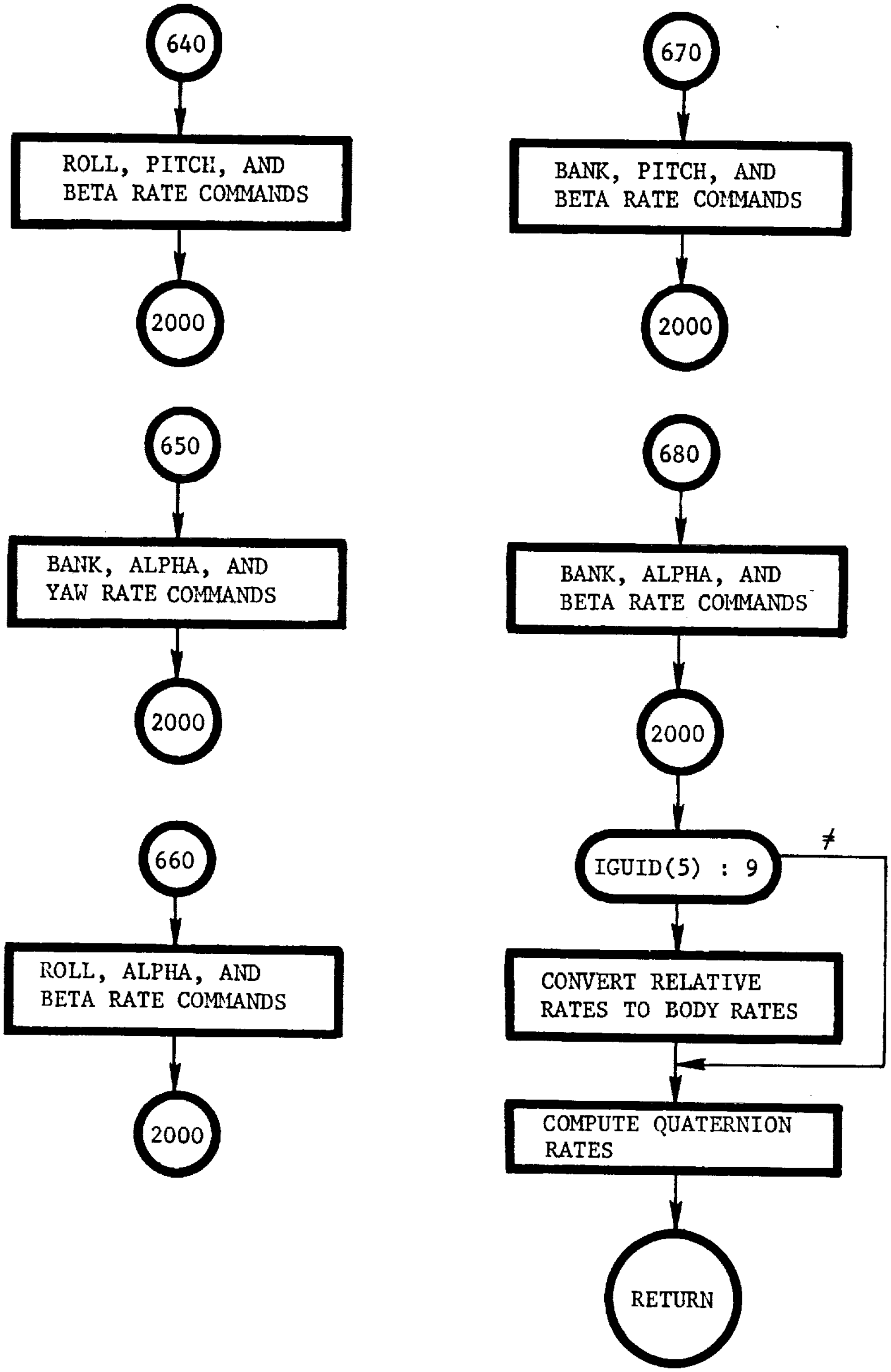

GUIDX
DETERMINE THE TABLE AND FUNCTION VALUES FOR CHANNEL N
COMPUTE THE COMMANDED ANGLE FOR CHANNEL N
UPDATE THE HISTORY VALUES
RETURN

GUID1: This routine calculates the steering angles based on the user-specified steering option.

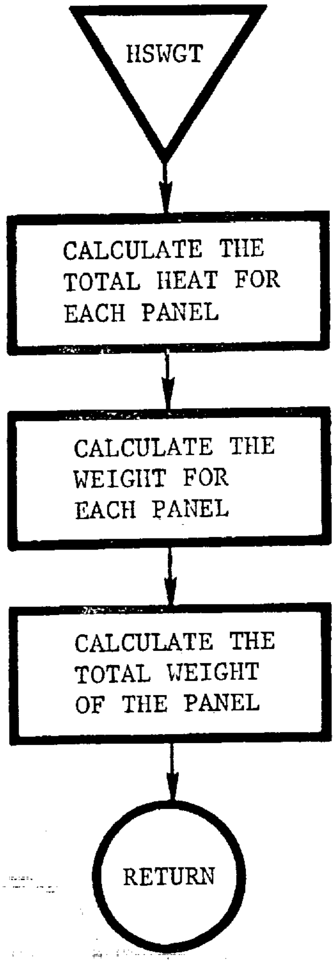

GUID1
IGUID(14) : 1
≠
=
ØLGM
IGUID(1) : 0
≠
=
COMPUTE AERODYNAMIC ANGLES BASED ON SELECTED STEERING OPTION
COMPUTE EULER ANGLES FOR EITHER RELATIVE OR INERTIAL ANGLES
IGUID(15) : 0
=
≠
ØLGØM
RETURN

GUID2: This routine calculates the Euler paramenter derivatives based on the inertial roll, pitch, and yaw, or the angle-of-attack, sideslip, and bank angle rate commands.

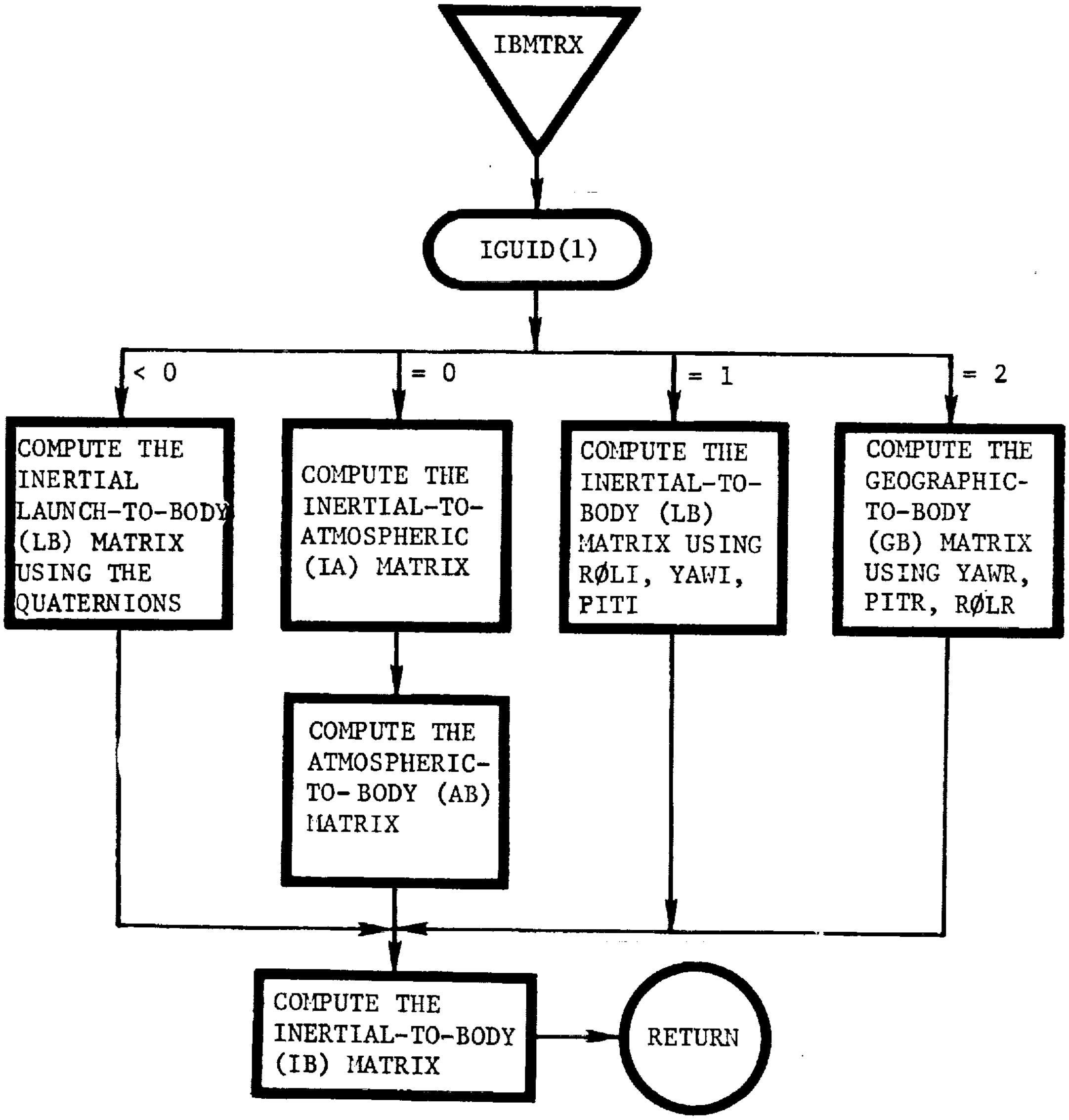

GUID2
IGUID(14) : 1
≠
=
ØLGM
ZERO OUT RATES
GAMLAM
IGUID(15) : 1
=
610
≠
COMPUTE THE B AND D VECTORS
GO TO (610, 620, 630, 640, 650, 660, 670, 680, 610, 630), IGUID(5)
610
ROLL, PITCH, AND YAW RATE COMMANDS
2000
620
BANK, PITCH, AND YAW RATE COMMANDS
2000
630
ROLL, ALPHA, AND YAW RATE COMMANDS
2000


640
ROLL, PITCH, AND BETA RATE COMMANDS
2000
650
BANK, ALPHA, AND YAW RATE COMMANDS
2000
660
ROLL, ALPHA, AND BETA RATE COMMANDS
2000
670
BANK, PITCH, AND BETA RATE COMMANDS
2000
680
BANK, ALPHA, AND BETA RATE COMMANDS
2000
IGUID(5) : 9
≠
CONVERT RELATIVE RATES TO BODY RATES
COMPUTE QUATERNION RATES
RETURN

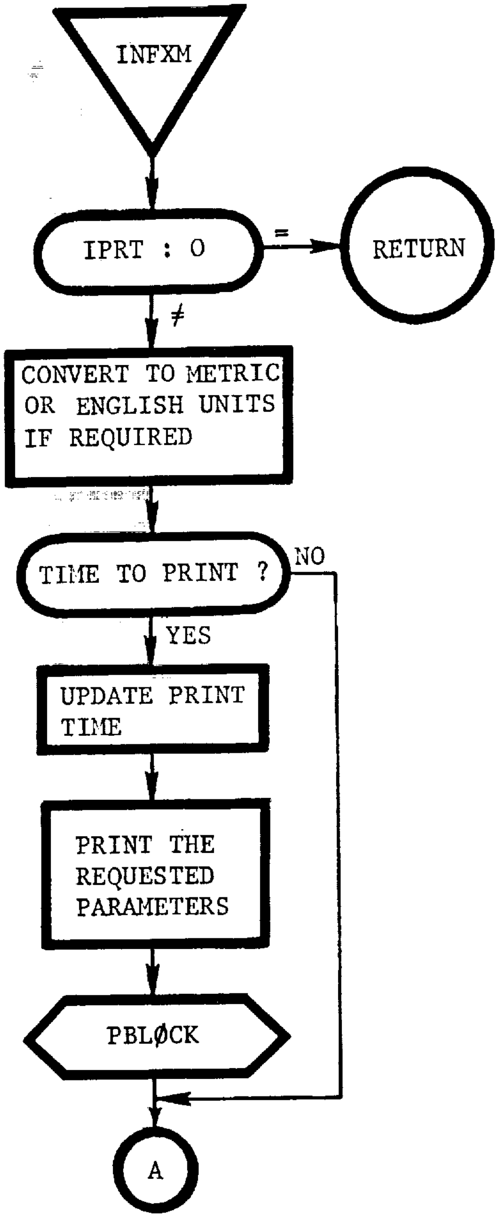
HSWGT: This routine calculates the total weight of the heat shield by summing the weights of its individual components.

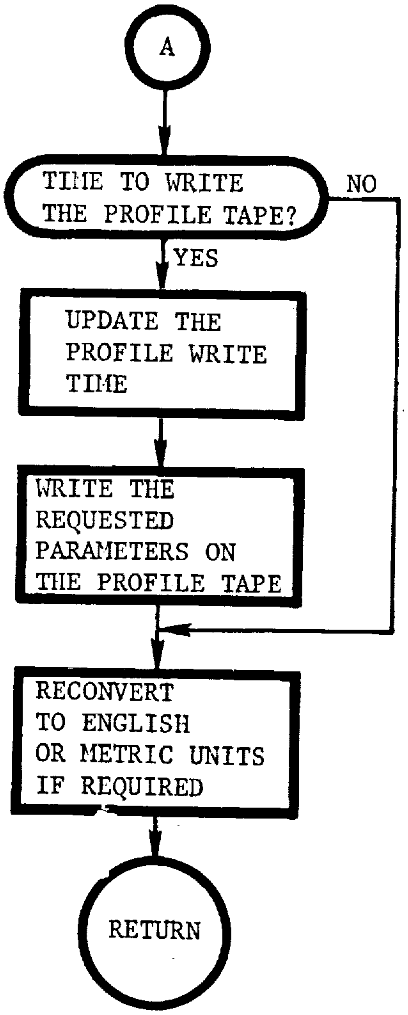

HSWGT
CALCULATE THE TOTAL HEAT FOR EACH PANEL
CALCULATE THE WEIGHT FOR EACH PANEL
CALCULATE THE TOTAL WEIGHT OF THE PANEL
RETURN

IBMTRX: This routine calculates the inertial-to-body (IB) matrix, based on the guidance (steering) option selected by user input.

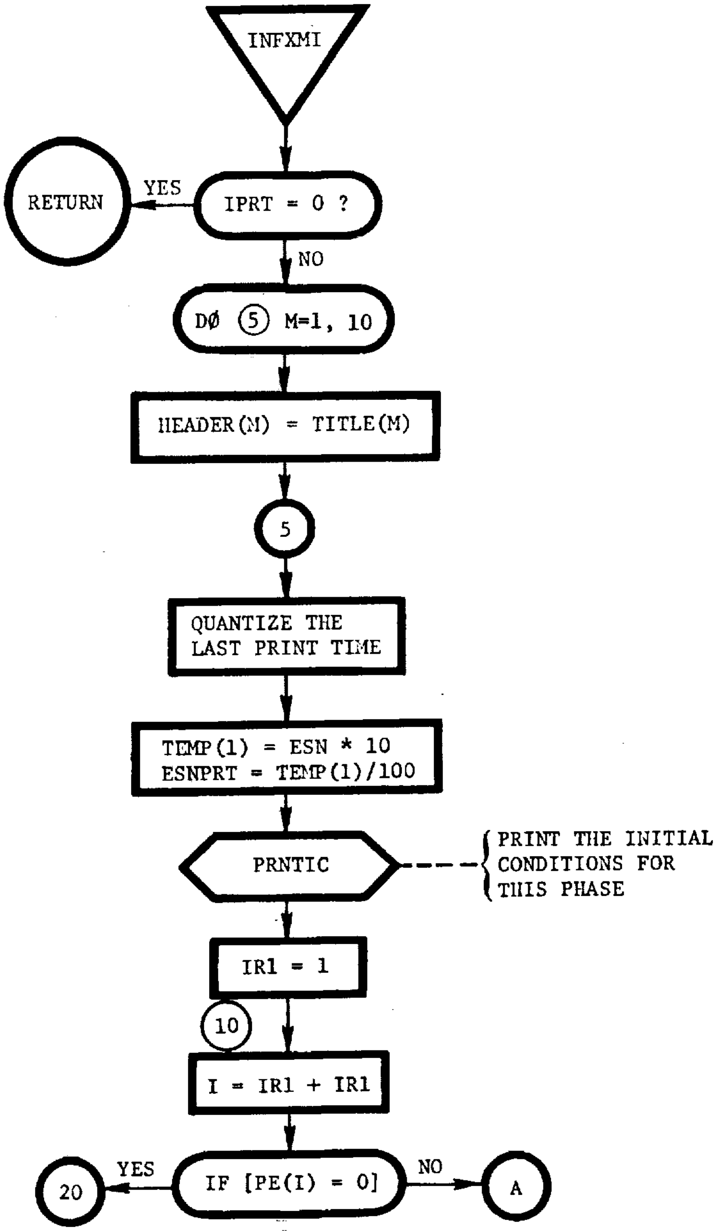

IBMTRX
IGUID(1)
< 0
= 0
= 1
= 2
COMPUTE THE INERTIAL LAUNCH-TO-BODY (LB) MATRIX USING THE QUATERNIONS
COMPUTE THE INERTIAL-TO-ATMOSPHERIC (IA) MATRIX
COMPUTE THE INERTIAL-TO-BODY (LB) MATRIX USING RØLI, YAWI, PITI
COMPUTE THE GEOGRAPHIC-TO-BODY (GB) MATRIX USING YAWR, PITR, RØLR
COMPUTE THE ATMOSPHERIC-TO-BODY (AB) MATRIX
COMPUTE THE INERTIAL-TO-BODY (IB) MATRIX
RETURN

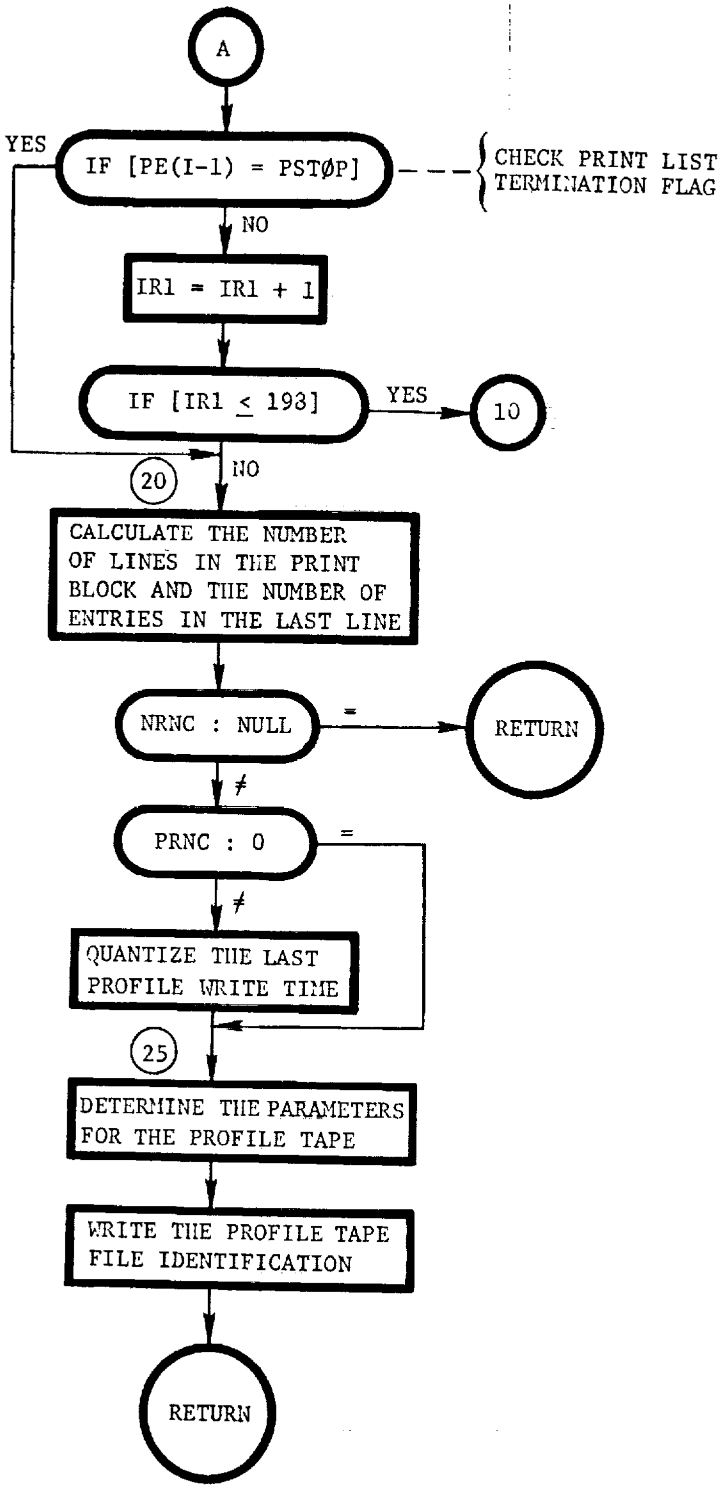
INFXM: This routine performs the output data processing functions. It also calls subroutine CØNIC, depending on the conic calculation option requested by NPC(1).


INFXM
IPRT : 0
=
RETURN
≠
CONVERT TO METRIC OR ENGLISH UNITS IF REQUIRED
TIME TO PRINT ?
NO
YES
UPDATE PRINT TIME
PRINT THE REQUESTED PARAMETERS
PBLØCK
A

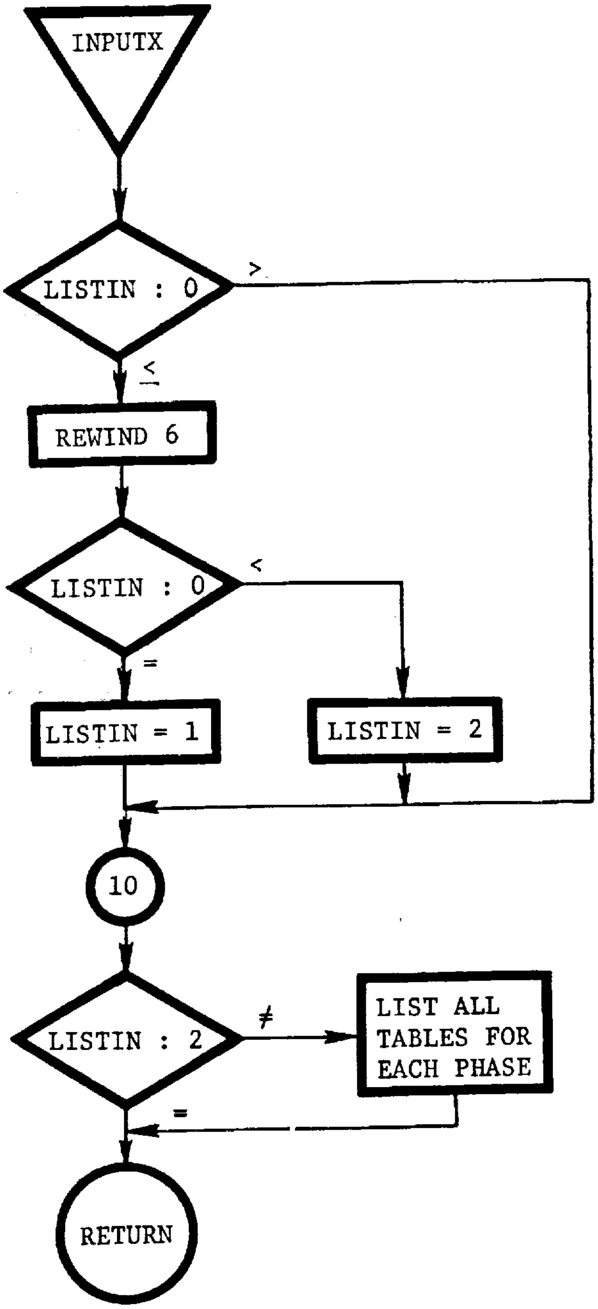

A
TIME TO WRITE THE PROFILE TAPE?
NO
YES
UPDATE THE PROFILE WRITE TIME
WRITE THE REQUESTED PARAMETERS ON THE PROFILE TAPE
RECONVERT TO ENGLISH OR METRIC UNITS IF REQUIRED
RETURN

INFXMI: This routine determines which variables are to be printed and which variables are to be written on the profile tape.




A
YES
IF [PE(I-1) = PSTØP]
CHECK PRINT LIST TERMINATION FLAG
NO
IR1 = IR1 + 1
IF [IR1 ≤ 198]
YES
10
20
NO
CALCULATE THE NUMBER OF LINES IN THE PRINT BLOCK AND THE NUMBER OF ENTRIES IN THE LAST LINE
NRNC : NULL
=
RETURN
≠
PRNC : 0
=
≠
QUANTIZE THE LAST PROFILE WRITE TIME
25
DETERMINE THE PARAMETERS FOR THE PROFILE TAPE
WRITE THE PROFILE TAPE FILE IDENTIFICATION
RETURN

INPUTX: This routine prints a summary of the input table data. The variable LISTIN is also checked to determine whether or not to rewind the output file, which eliminates the listing of the input data deck.



INTGRL (LIST, NUM, KEY): This routine initializes the list of variables to be integrated. The list of variables to be integrated is called DYNIL. It contains 148 cells that are stored in subroutine BLKDAT. There are three cells for each integrated variable, which corresponds to a total of 49 integration variables. The first cell indicates the total size of the array, the second cell contains the Hollerith name of the first variable to be integrated, the third cell contains the Hollerith name of its derivative, and the fourth cell contains a flag to indicate whether or not to integrate that variable, etc. If the flag is zero, the variable is not integrated; if the flag is equal to 1, the variable is integrated.

INTGRL is also used to turn the integration of variables on or off as desired. For example, if NPC(11) = 1, we wish to activate the inequality constraint integrations (i.e., the integration of FVAL1, FVAL2, FVAL3). In this case, subroutine MØTIAL calls INTGRL as follows to activate the integration:

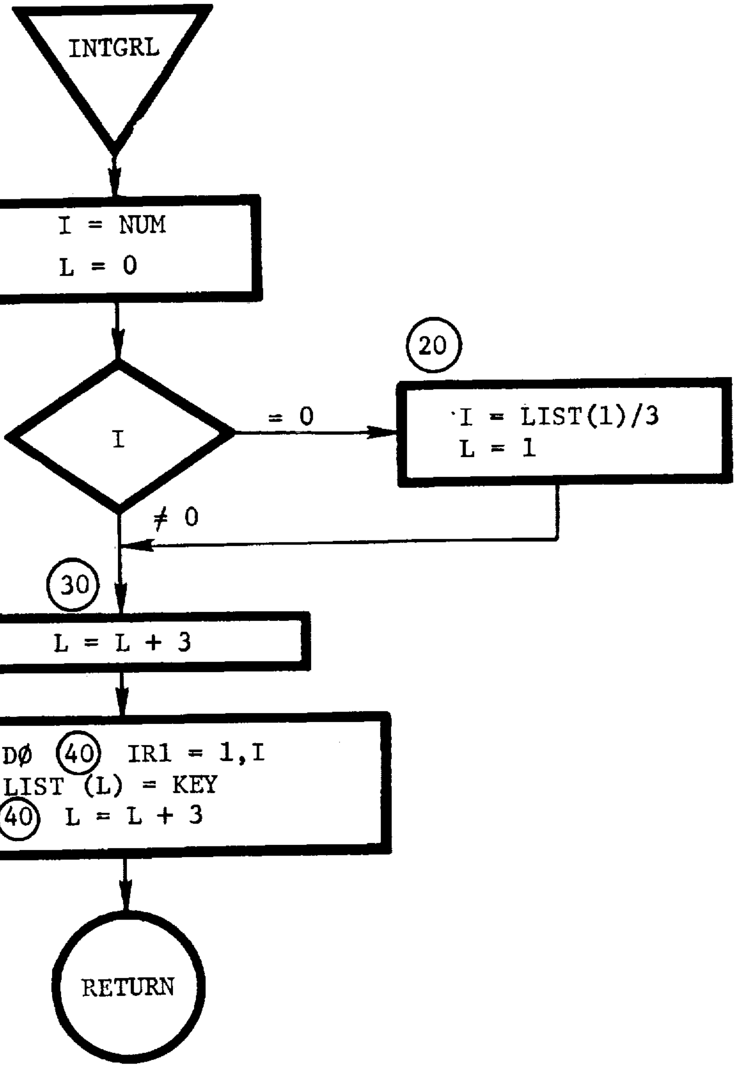
```
CALL INTGRL (DYNIL(38), N03, N01)
```

where:

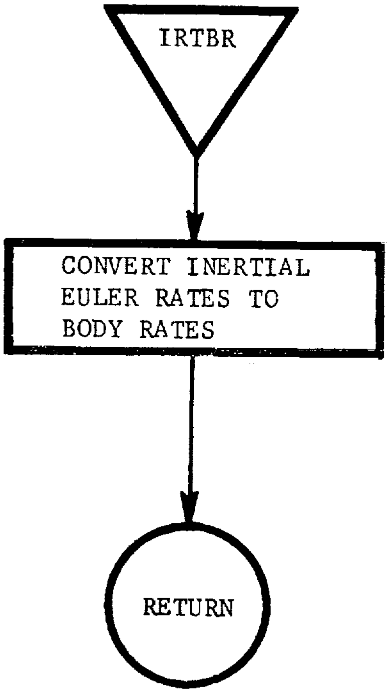
DYNIL(38) = the position of FVAL1 in the array DYNIL

N03 = fixed point 3, which means that the three integrals namely, FVAL1, FVAL2, and FVAL3, are to be turned on, since they are in sequence.

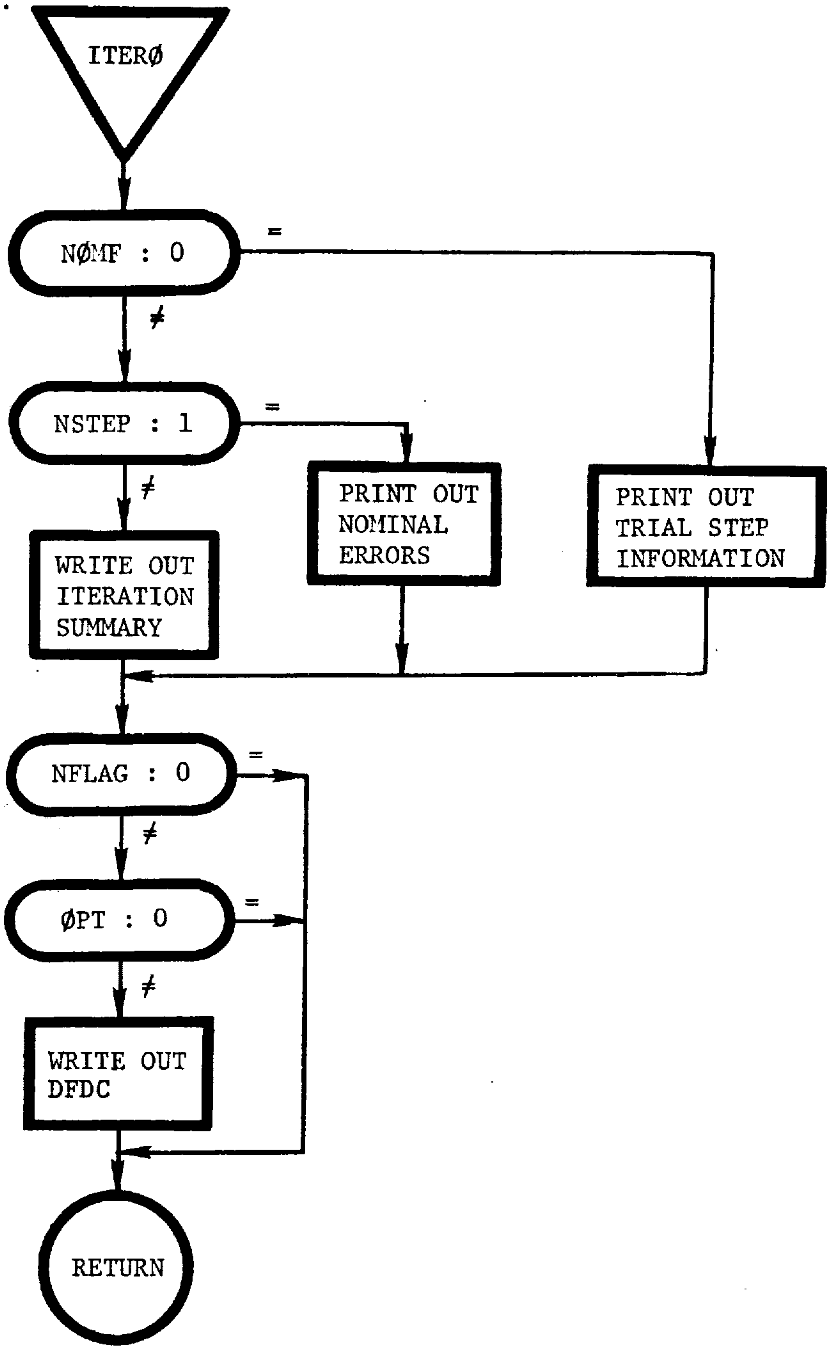
N01 = fixed point 1; this means turn on the integration of the variables. If the argument were N00 (fixed-point zero), the integration would be turned off.


INTGRL
I = NUM
L = 0
I
= 0
20
I = LIST(1)/3
L = 1
≠ 0
30
L = L + 3
DØ 40 IR1 = 1,I
LIST (L) = KEY
40 L = L + 3
RETURN

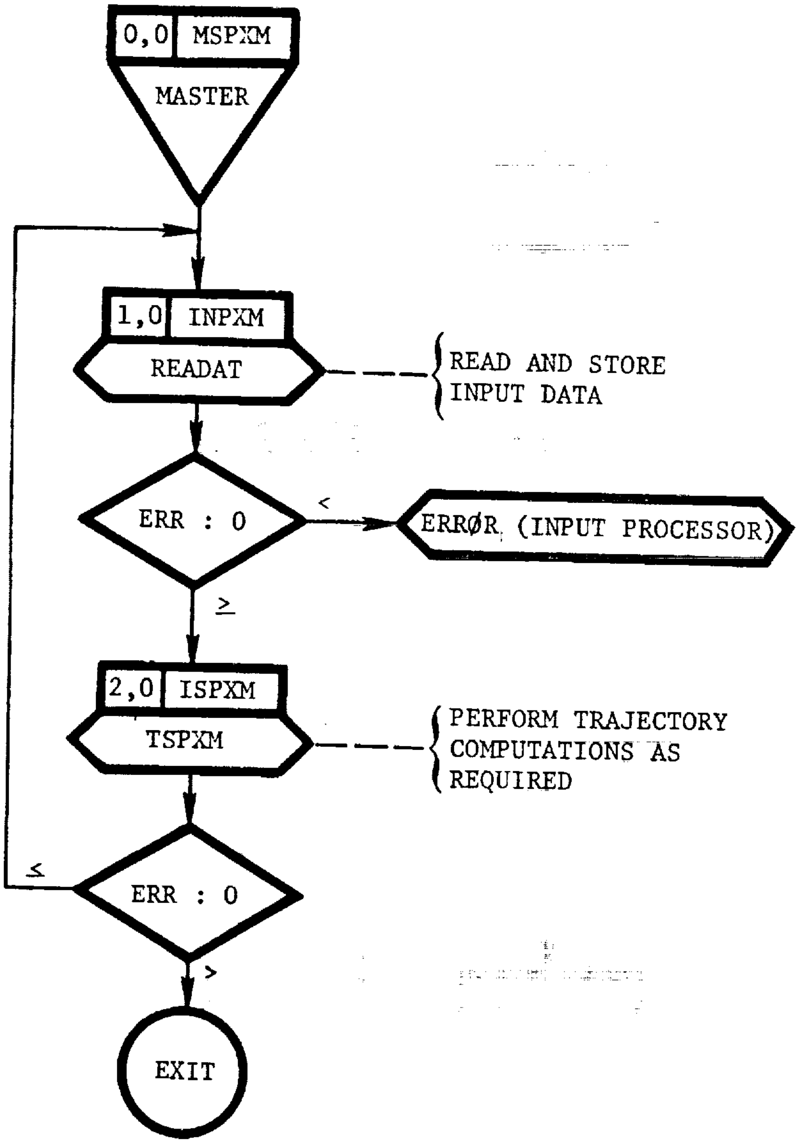
IRTBR: This routine calculates body rates from inertial Euler angle rates.



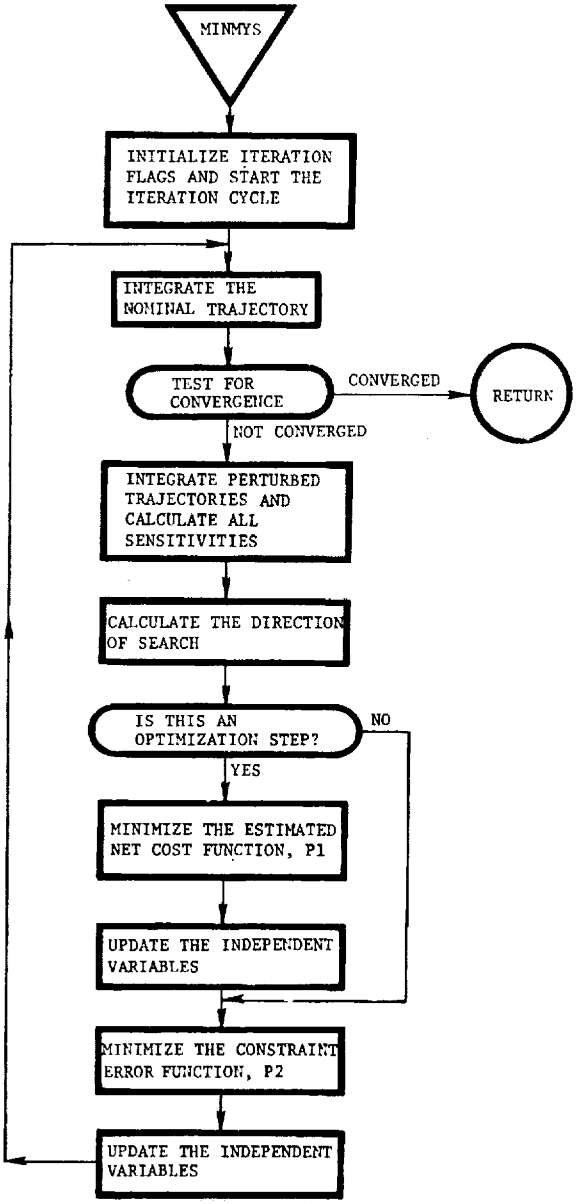
ITERØ: Main program of overlay (2,6). This routine prints out the iteration summary as required. The iteration summary contains all the information relating to the search/optimization algorithm.



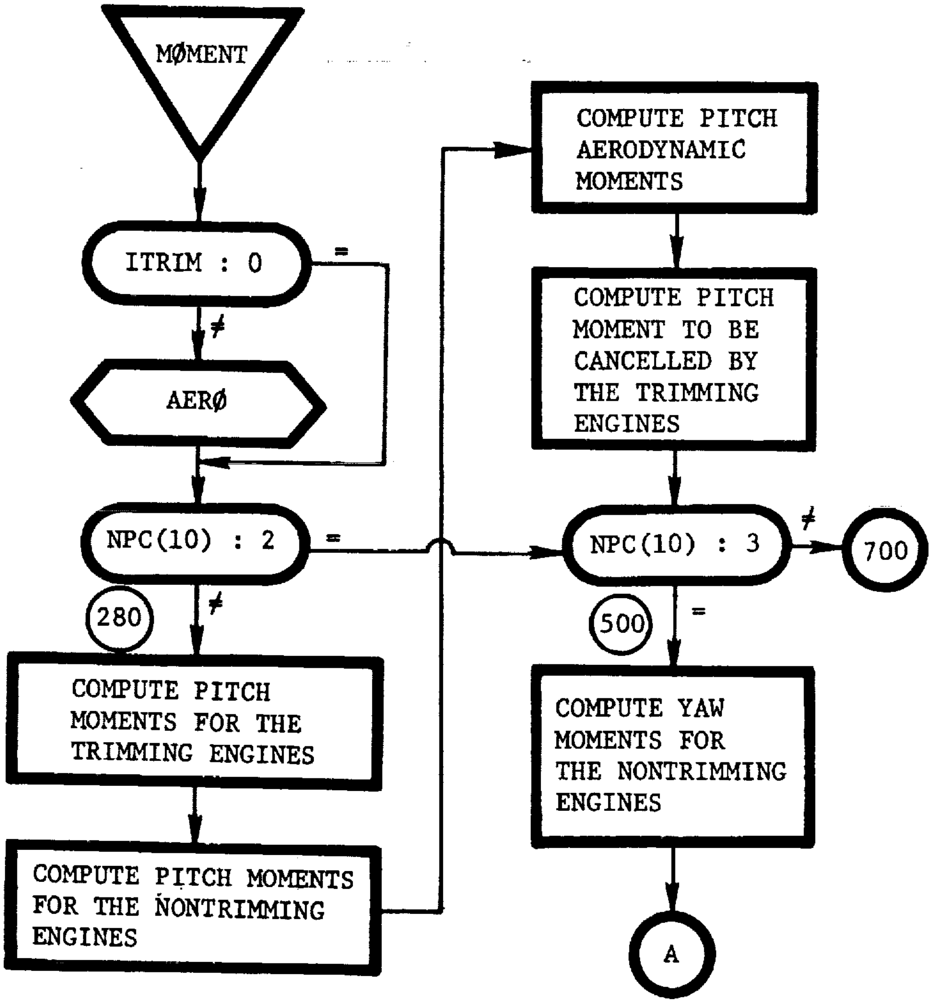
MASTER: This is the main program of overlay (0,0). It decides whether to read input data or execute the problem.



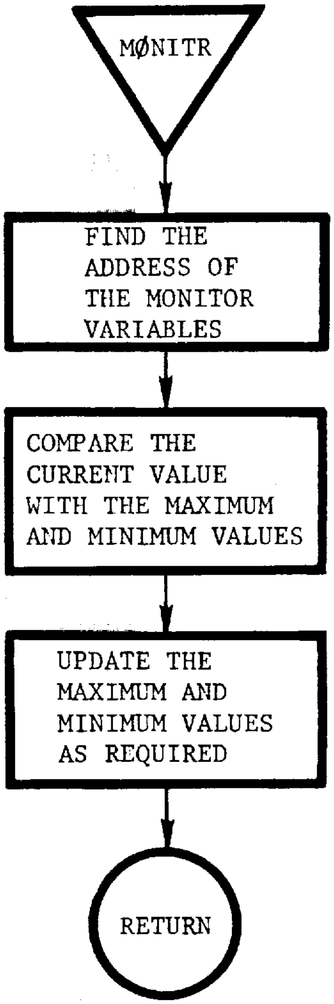
MINMYS: This routine contains the optimization executive logic for the various options that are available. It phases the various iteration paths by examining various properties of the objective function and the constraint manifolds.

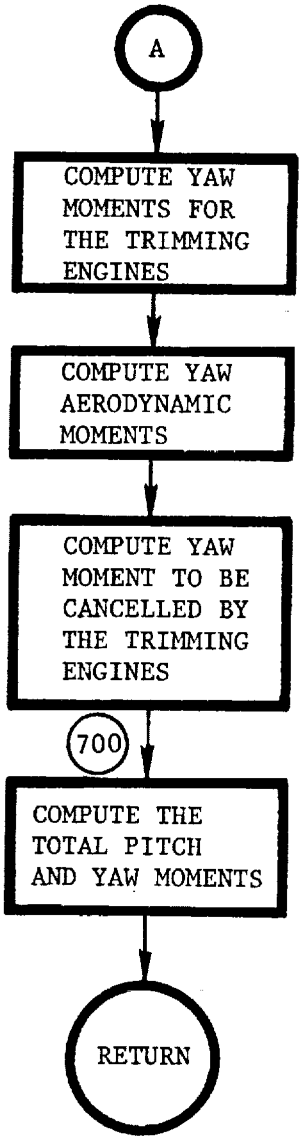

MINMYS
INITIALIZE ITERATION FLAGS AND START THE ITERATION CYCLE
INTEGRATE THE NOMINAL TRAJECTORY
TEST FOR CONVERGENCE
CONVERGED
RETURN
NOT CONVERGED
INTEGRATE PERTURBED TRAJECTORIES AND CALCULATE ALL SENSITIVITIES
CALCULATE THE DIRECTION OF SEARCH
IS THIS AN OPTIMIZATION STEP?
NO
YES
MINIMIZE THE ESTIMATED NET COST FUNCTION, P1
UPDATE THE INDEPENDENT VARIABLES
MINIMIZE THE CONSTRAINT ERROR FUNCTION, P2
UPDATE THE INDEPENDENT VARIABLES

MØMENT: This routine calculates the total thrust and aerodynamic moments for static trim.


MØMENT
ITRIM : 0
=
≠
AERØ
NPC(10) : 2
=
280
≠
COMPUTE PITCH MOMENTS FOR THE TRIMMING ENGINES
COMPUTE PITCH MOMENTS FOR THE NONTRIMMING ENGINES
COMPUTE PITCH AERODYNAMIC MOMENTS
COMPUTE PITCH MOMENT TO BE CANCELLED BY THE TRIMMING ENGINES
NPC(10) : 3
≠
700
500
=
COMPUTE YAW MOMENTS FOR THE NONTRIMMING ENGINES
A

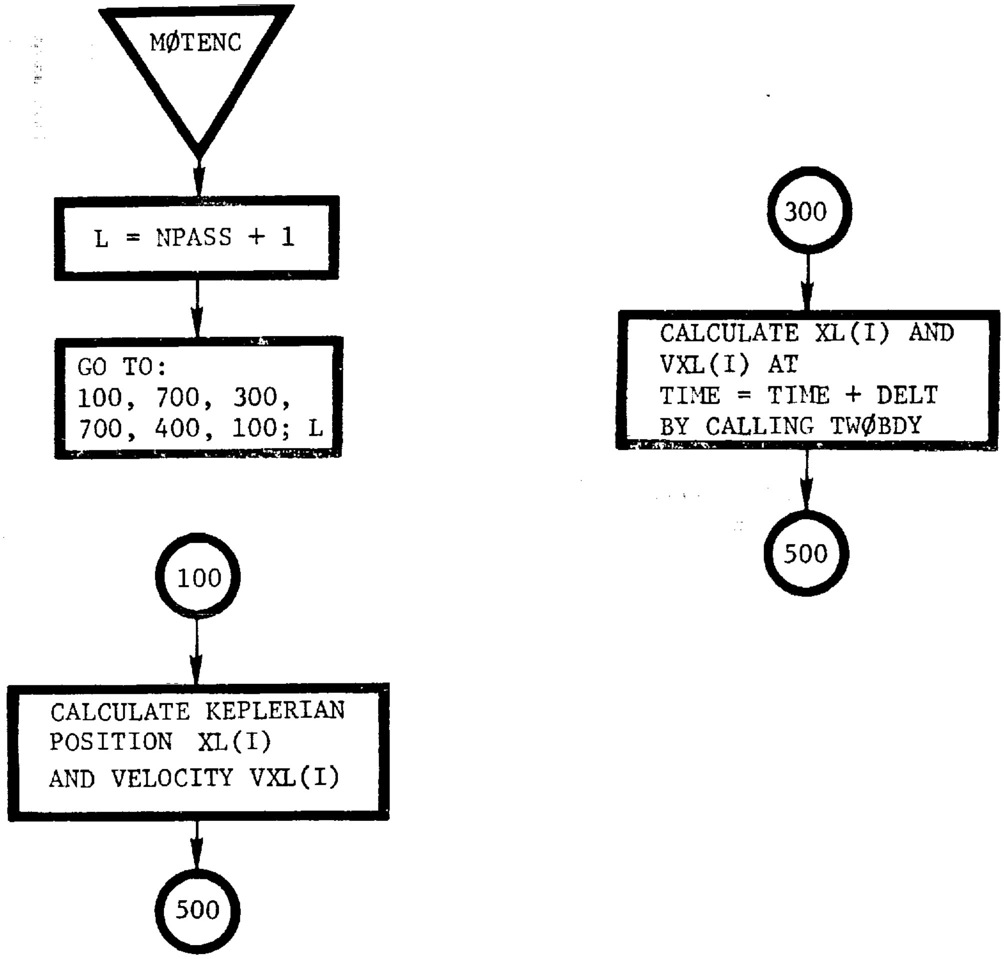
MØNITR: This routine determines the maximum and minimum values of the user-specified monitor variables.

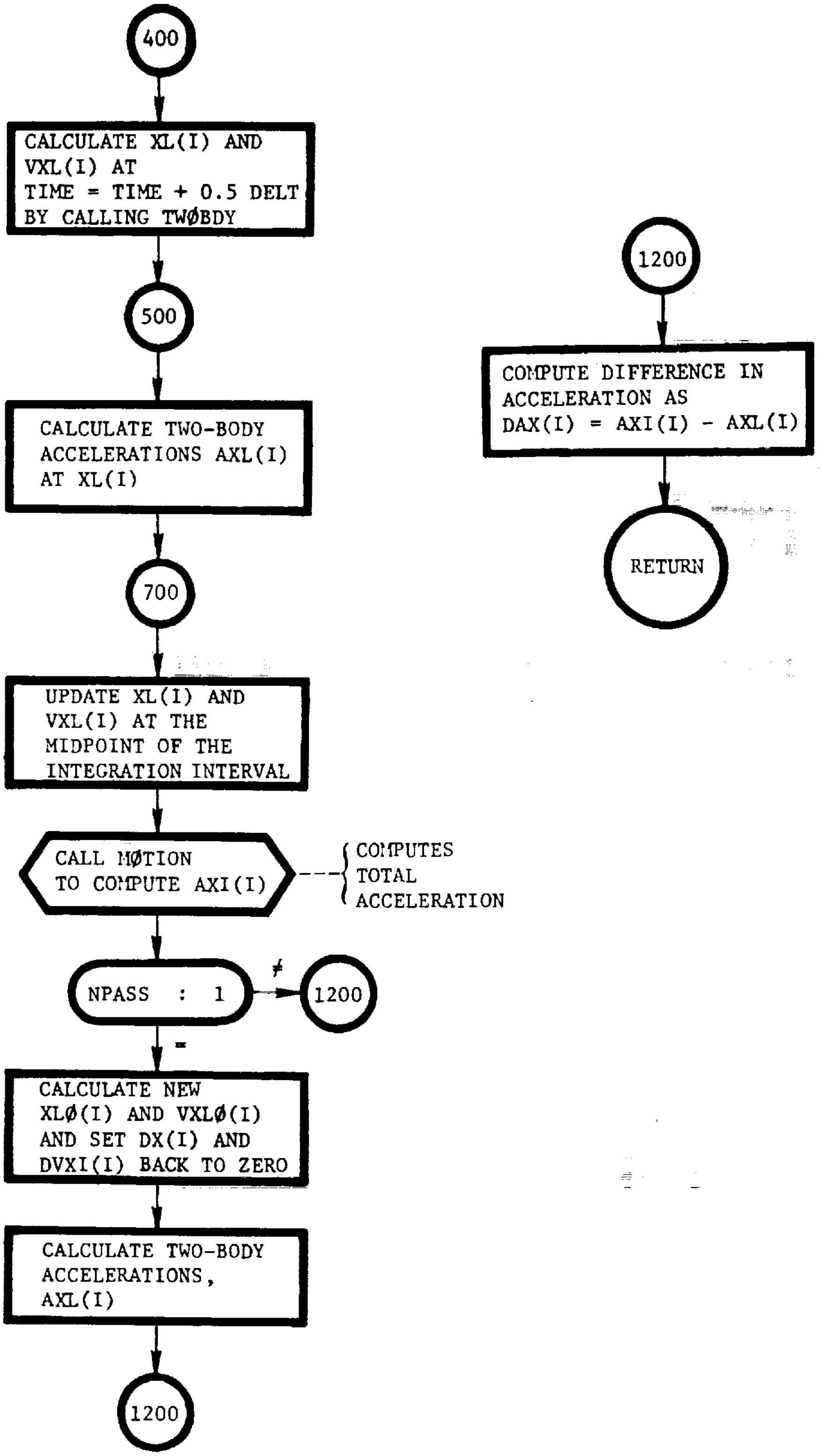

MØNITR
FIND THE ADDRESS OF THE MONITOR VARIABLES
COMPARE THE CURRENT VALUE WITH THE MAXIMUM AND MINIMUM VALUES
UPDATE THE MAXIMUM AND MINIMUM VALUES AS REQUIRED
RETURN



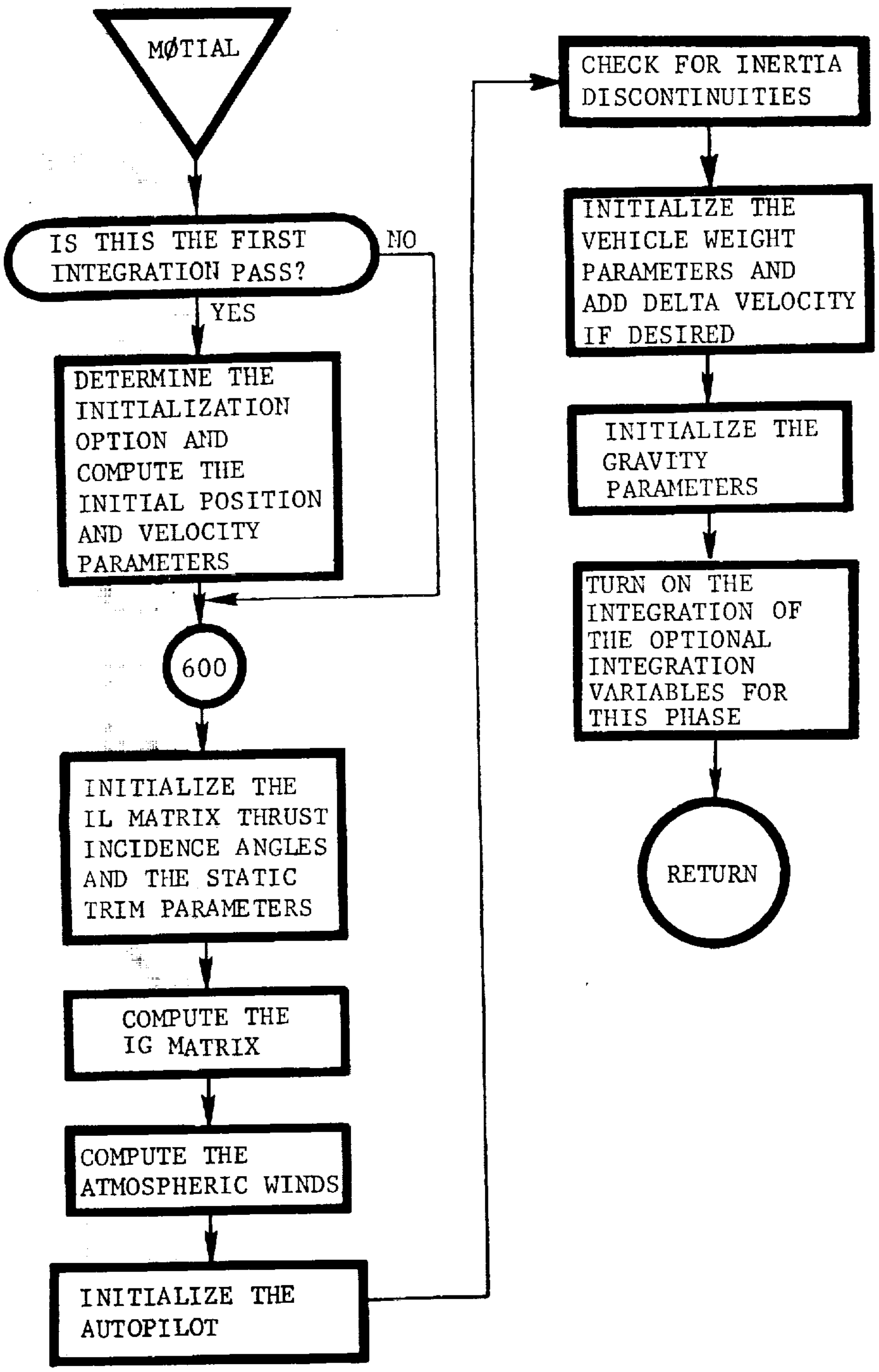
A
COMPUTE YAW MOMENTS FOR THE TRIMMING ENGINES
COMPUTE YAW AERODYNAMIC MOMENTS
COMPUTE YAW MOMENT TO BE CANCELLED BY THE TRIMMING ENGINES
700
COMPUTE THE TOTAL PITCH AND YAW MOMENTS
RETURN

MØTENC: This routine computes the difference between the total acceleration on the vehicle and the two-body acceleration.

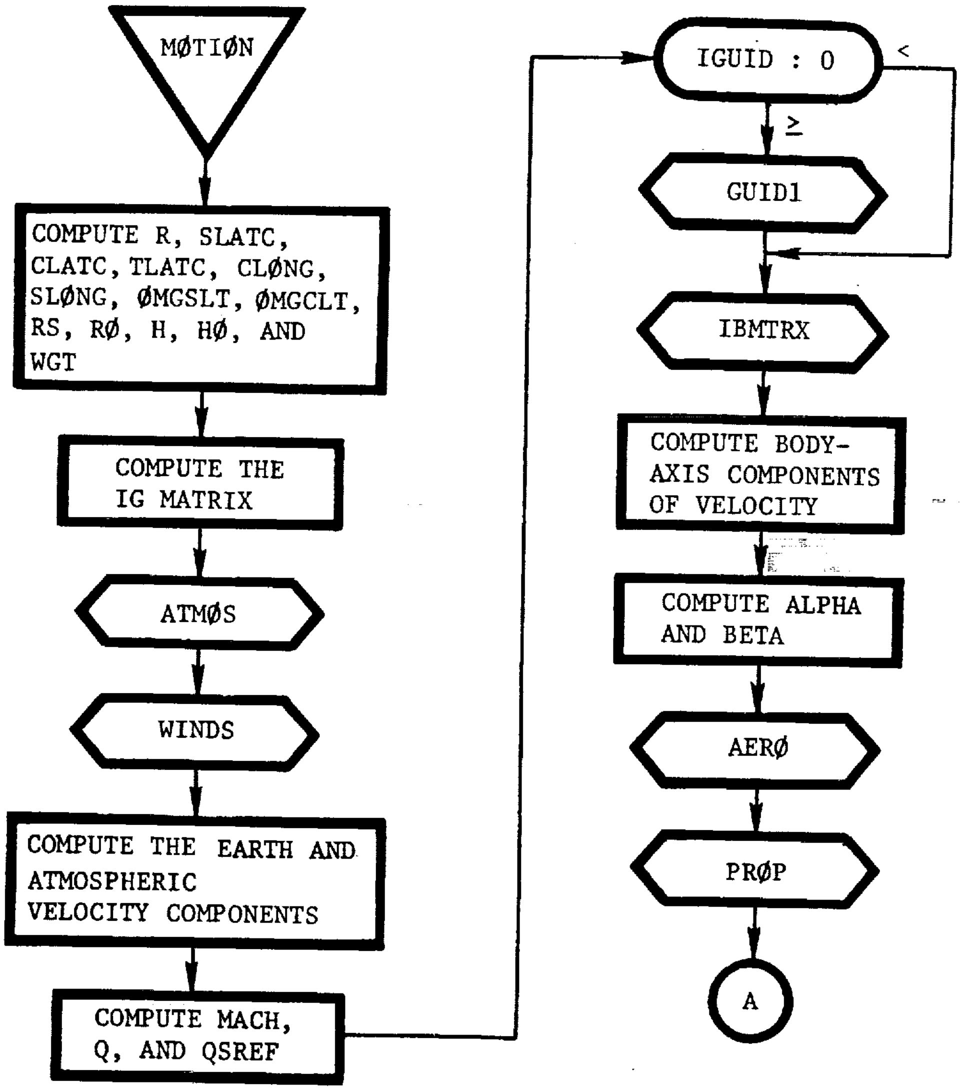

MØTENC
L = NPASS + 1
GO TO:
100, 700, 300,
700, 400, 100; L
100
CALCULATE KEPLERIAN POSITION XL(I) AND VELOCITY VXL(I)
500
300
CALCULATE XL(I) AND VXL(I) AT TIME = TIME + DELT BY CALLING TWØBDY
500

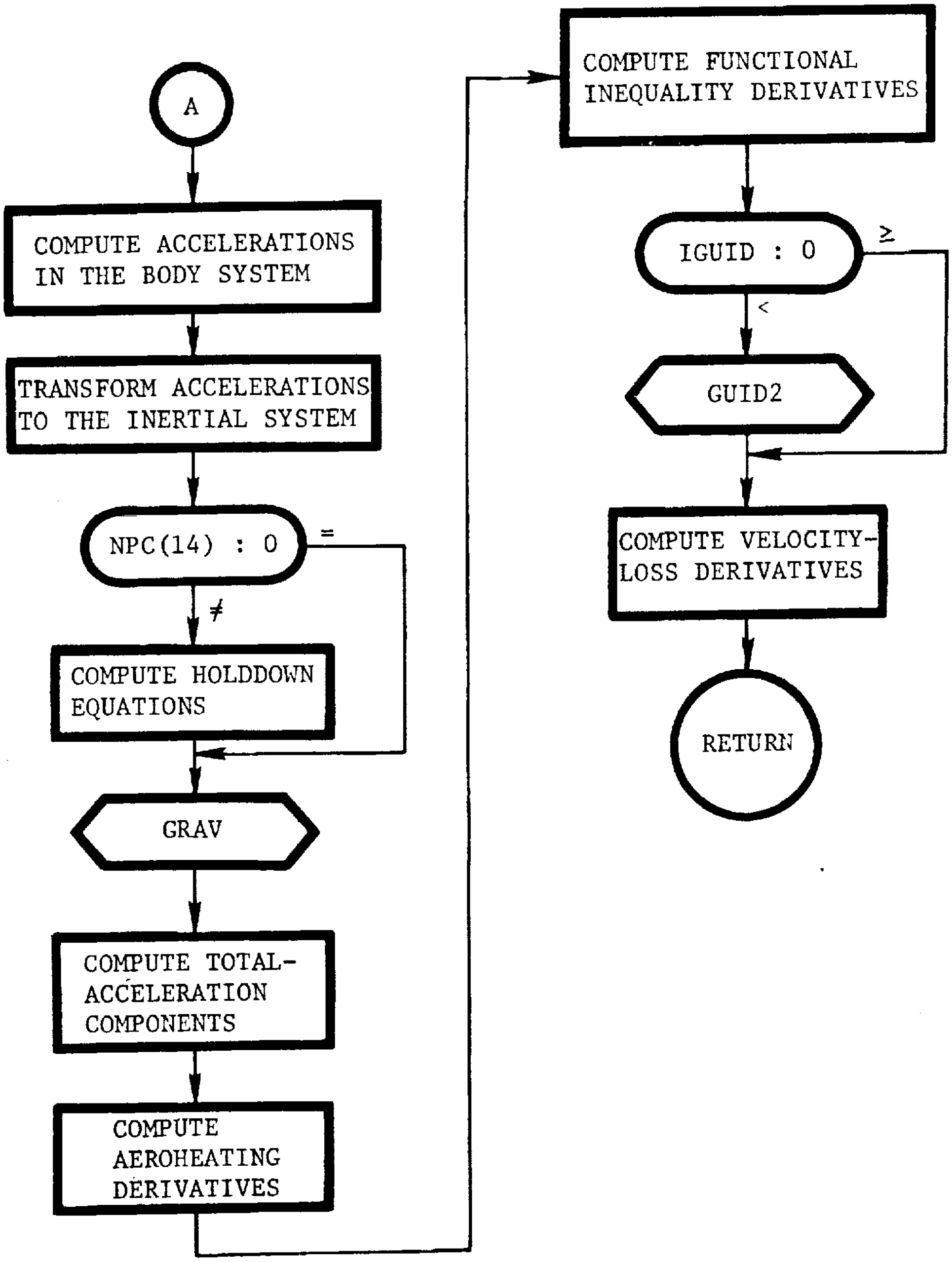

400
CALCULATE XL(I) AND VXL(I) AT TIME = TIME + 0.5 DELT BY CALLING TWØBDY
500
CALCULATE TWO-BODY ACCELERATIONS AXL(I) AT XL(I)
700
UPDATE XL(I) AND VXL(I) AT THE MIDPOINT OF THE INTEGRATION INTERVAL
CALL MØTION TO COMPUTE AXI(I)
COMPUTES TOTAL ACCELERATION
NPASS : 1
≠
1200
=
CALCULATE NEW XLØ(I) AND VXLØ(I) AND SET DX(I) AND DVXI(I) BACK TO ZERO
CALCULATE TWO-BODY ACCELERATIONS, AXL(I)
1200
1200
COMPUTE DIFFERENCE IN ACCELERATION AS DAX(I) = AXI(I) - AXL(I)
RETURN

MØTIAL: This routine initializes the equations of motion.

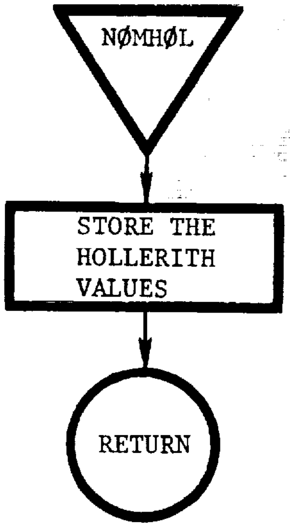

MØTIAL
IS THIS THE FIRST INTEGRATION PASS?
NO
YES
DETERMINE THE INITIALIZATION OPTION AND COMPUTE THE INITIAL POSITION AND VELOCITY PARAMETERS
600
INITIALIZE THE IL MATRIX THRUST INCIDENCE ANGLES AND THE STATIC TRIM PARAMETERS
COMPUTE THE IG MATRIX
COMPUTE THE ATMOSPHERIC WINDS
INITIALIZE THE AUTOPILOT
CHECK FOR INERTIA DISCONTINUITIES
INITIALIZE THE VEHICLE WEIGHT PARAMETERS AND ADD DELTA VELOCITY IF DESIRED
INITIALIZE THE GRAVITY PARAMETERS
TURN ON THE INTEGRATION OF THE OPTIONAL INTEGRATION VARIABLES FOR THIS PHASE
RETURN

MØTIØN: This routine calculates the equations of motion.

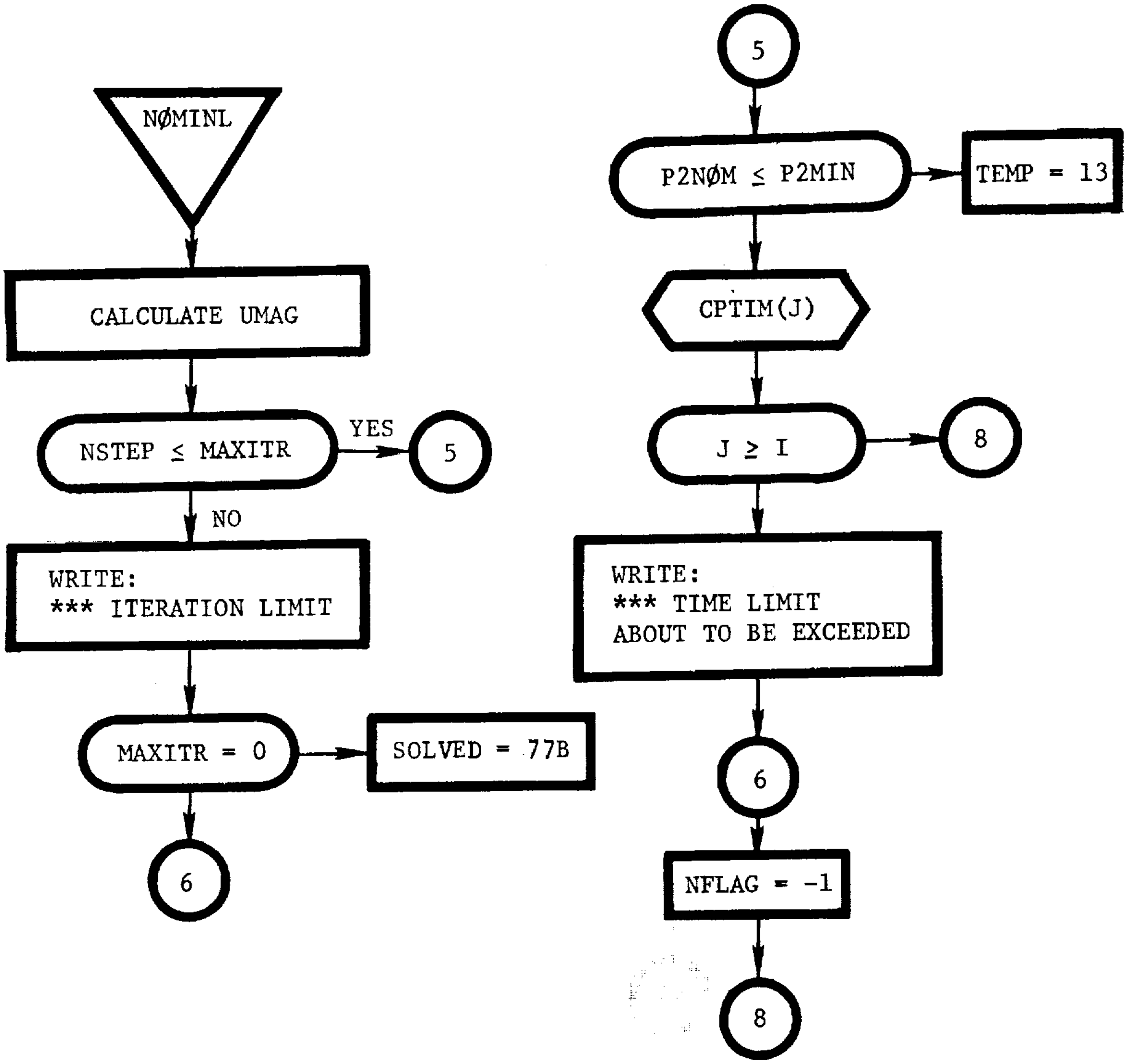

MØTIØN
COMPUTE R, SLATC, CLATC, TLATC, CLØNG, SLØNG, ØMGSLT, ØMGCLT, RS, RØ, H, HØ, AND WGT
COMPUTE THE IG MATRIX
ATMØS
WINDS
COMPUTE THE EARTH AND ATMOSPHERIC VELOCITY COMPONENTS
COMPUTE MACH, Q, AND QSREF
IGUID : 0
<
≥
GUID1
IBMTRX
COMPUTE BODY-AXIS COMPONENTS OF VELOCITY
COMPUTE ALPHA AND BETA
AERØ
PRØP
A

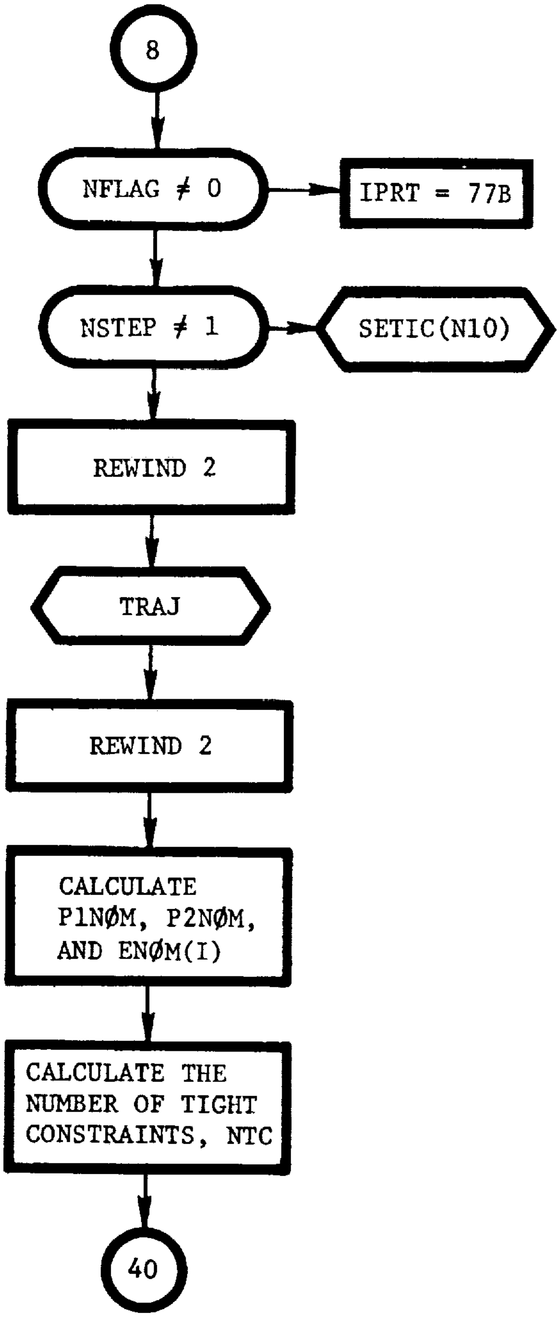

A
COMPUTE ACCELERATIONS IN THE BODY SYSTEM
TRANSFORM ACCELERATIONS TO THE INERTIAL SYSTEM
NPC(14) : 0
=
≠
COMPUTE HOLDDOWN EQUATIONS
GRAV
COMPUTE TOTAL-ACCELERATION COMPONENTS
COMPUTE AEROHEATING DERIVATIVES
COMPUTE FUNCTIONAL INEQUALITY DERIVATIVES
IGUID : 0
≥
<
GUID2
COMPUTE VELOCITY-LOSS DERIVATIVES
RETURN

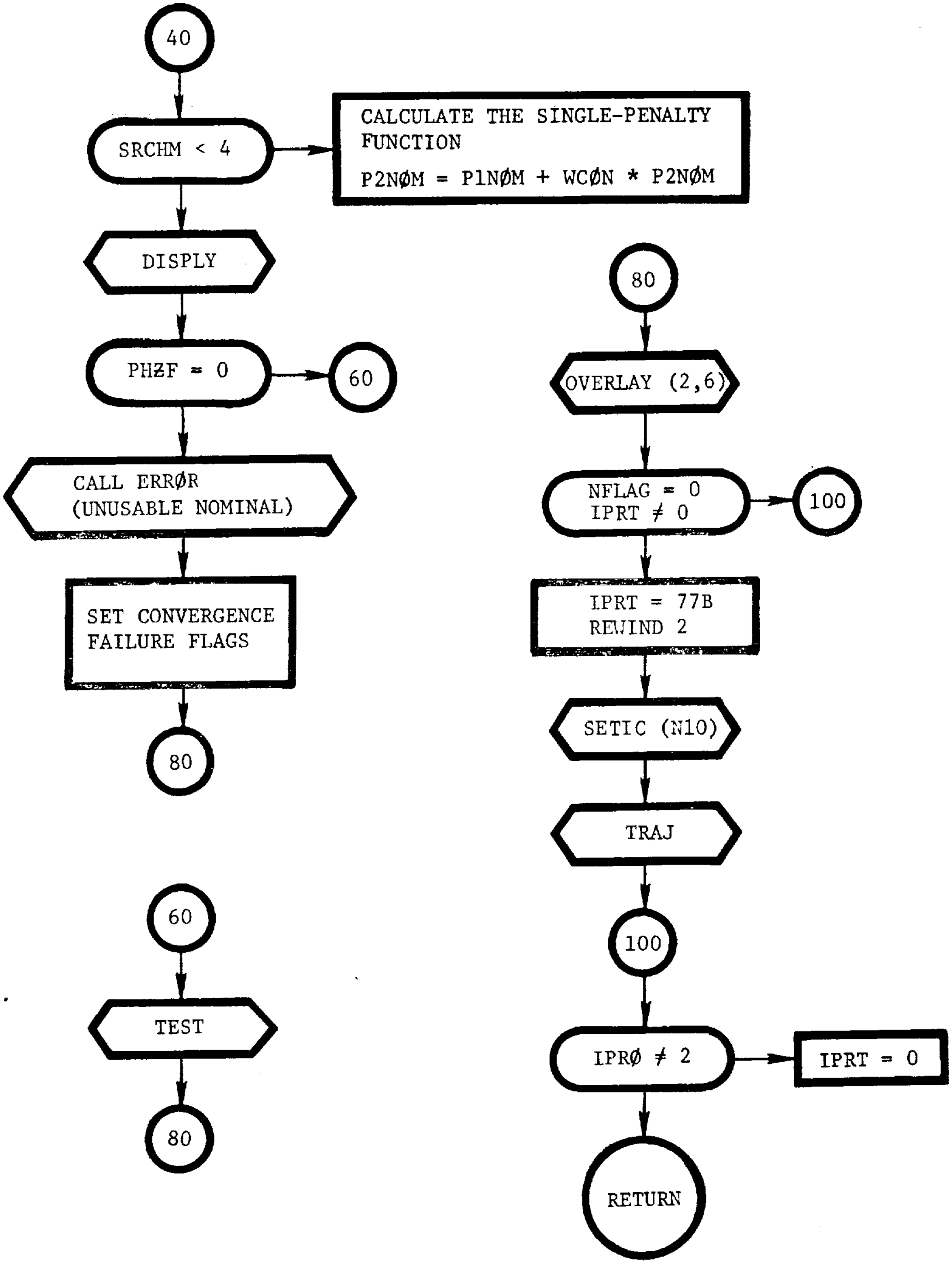
NØMHØL: This routine initializes the values of all HOLLERITH variables to the stored values.



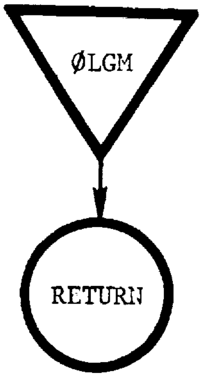
NØMINL: This subroutine runs the nominal trajectories (one per iteration), saving core at the beginning of each phase. The routine also calls TEST to determine if the iteration reference has converged or failed to converge.



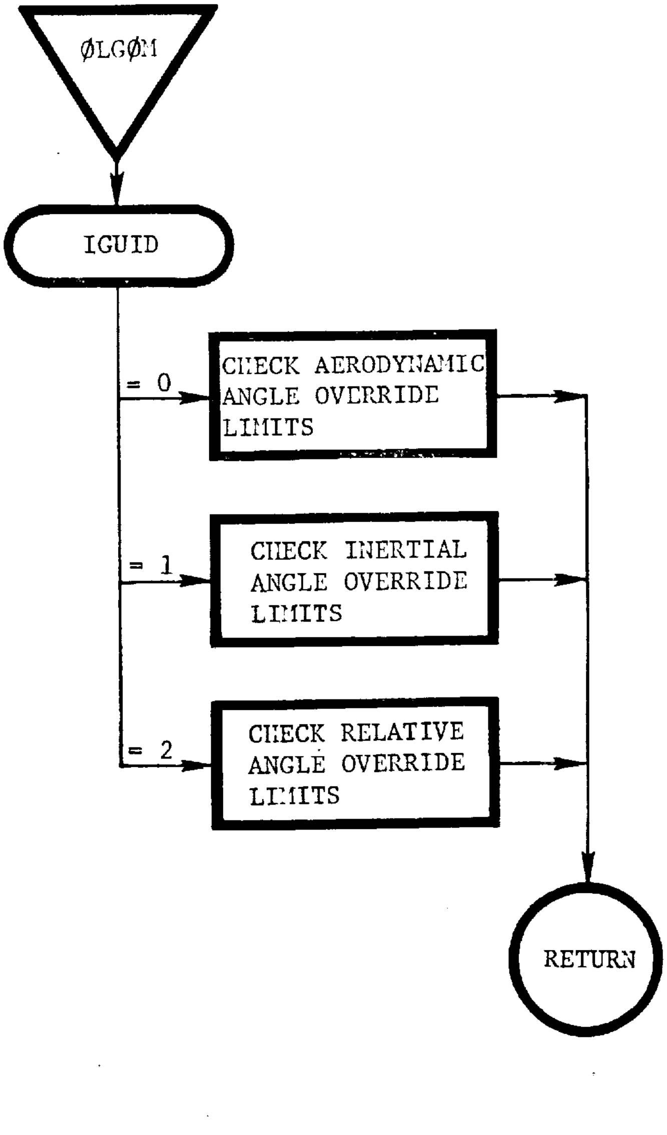

8
NFLAG ≠ 0
IPRT = 77B
NSTEP ≠ 1
SETIC(N10)
REWIND 2
TRAJ
REWIND 2
CALCULATE
P1NØM, P2NØM,
AND ENØM(I)
CALCULATE THE
NUMBER OF TIGHT
CONSTRAINTS, NTC
40


40
SRCHM < 4
CALCULATE THE SINGLE-PENALTY FUNCTION
P2NØM = P1NØM + WCØN * P2NØM
DISPLY
PHZF = 0
60
CALL ERRØR (UNUSABLE NOMINAL)
SET CONVERGENCE FAILURE FLAGS
80
60
TEST
80
80
OVERLAY (2,6)
NFLAG = 0
IPRT ≠ 0
100
IPRT = 77B
REWIND 2
SETIC (N10)
TRAJ
100
IPRØ ≠ 2
IPRT = 0
RETURN

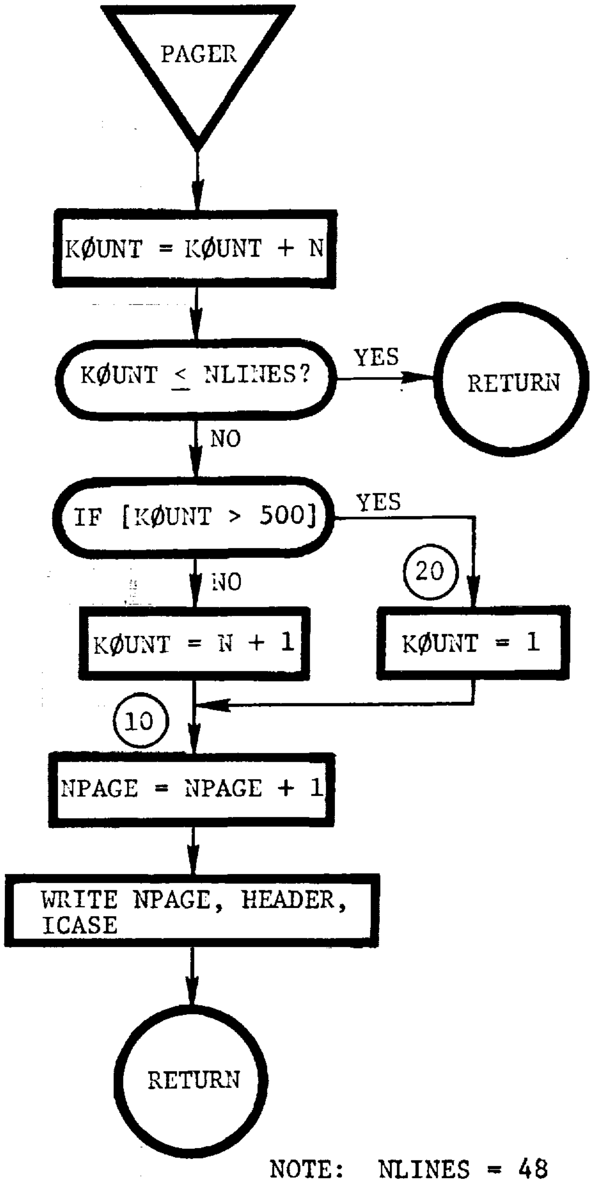
ØLGM: This is a blank routine that is to be used for special open-loop guidance (steering) models. The polynominal coefficients or angular values to be used by GUID1 can be calculated in this routine and then used by the user-selected option, based on the IGUID array.



ØLGØM: This routine allows the guidance (steering) option values determined by the IGUID array to be overridden if a specified parameter test has been violated by the commanded angle. For example, the commanded angle of attack can be overridden if the value of QALPHA exceeds an input limit. This allows the program to follow the limit until it is no longer violated by the commanded angle.


ØLGØM
IGUID
= 0
CHECK AERODYNAMIC ANGLE OVERRIDE LIMITS
= 1
CHECK INERTIAL ANGLE OVERRIDE LIMITS
= 2
CHECK RELATIVE ANGLE OVERRIDE LIMITS
RETURN

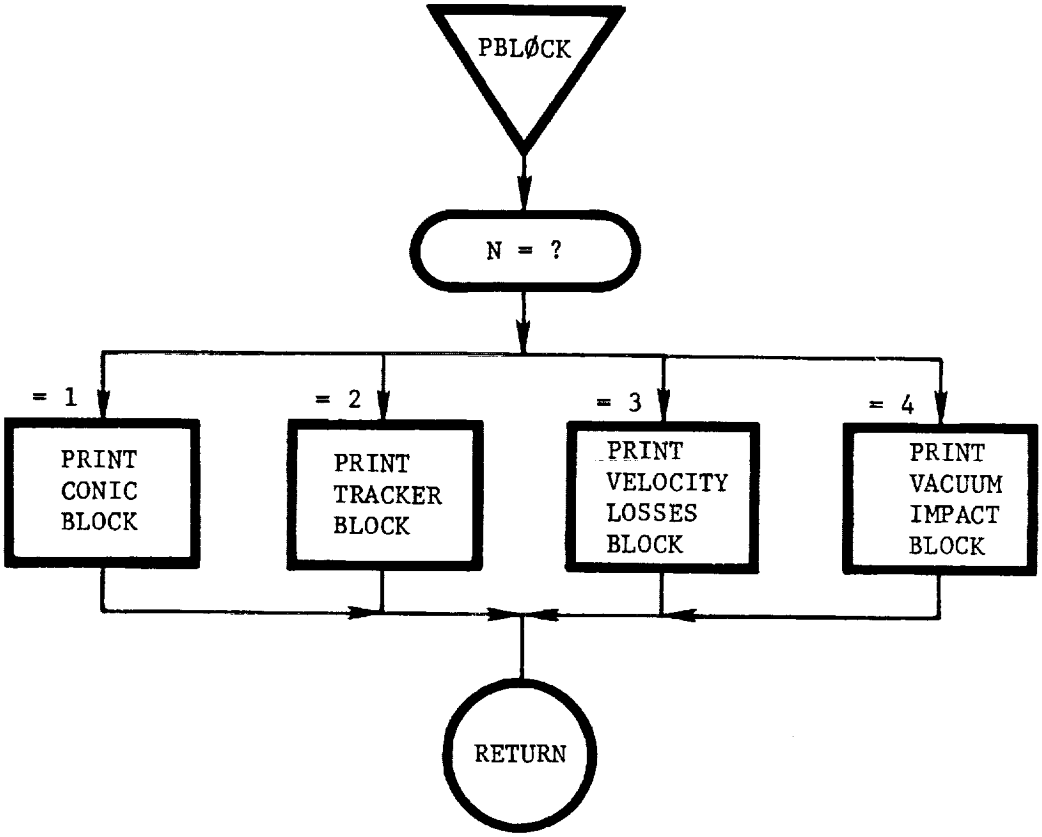
PAGER(N): This routine determines when a new page is required prior to printing. The argument is the number of lines to be printed.



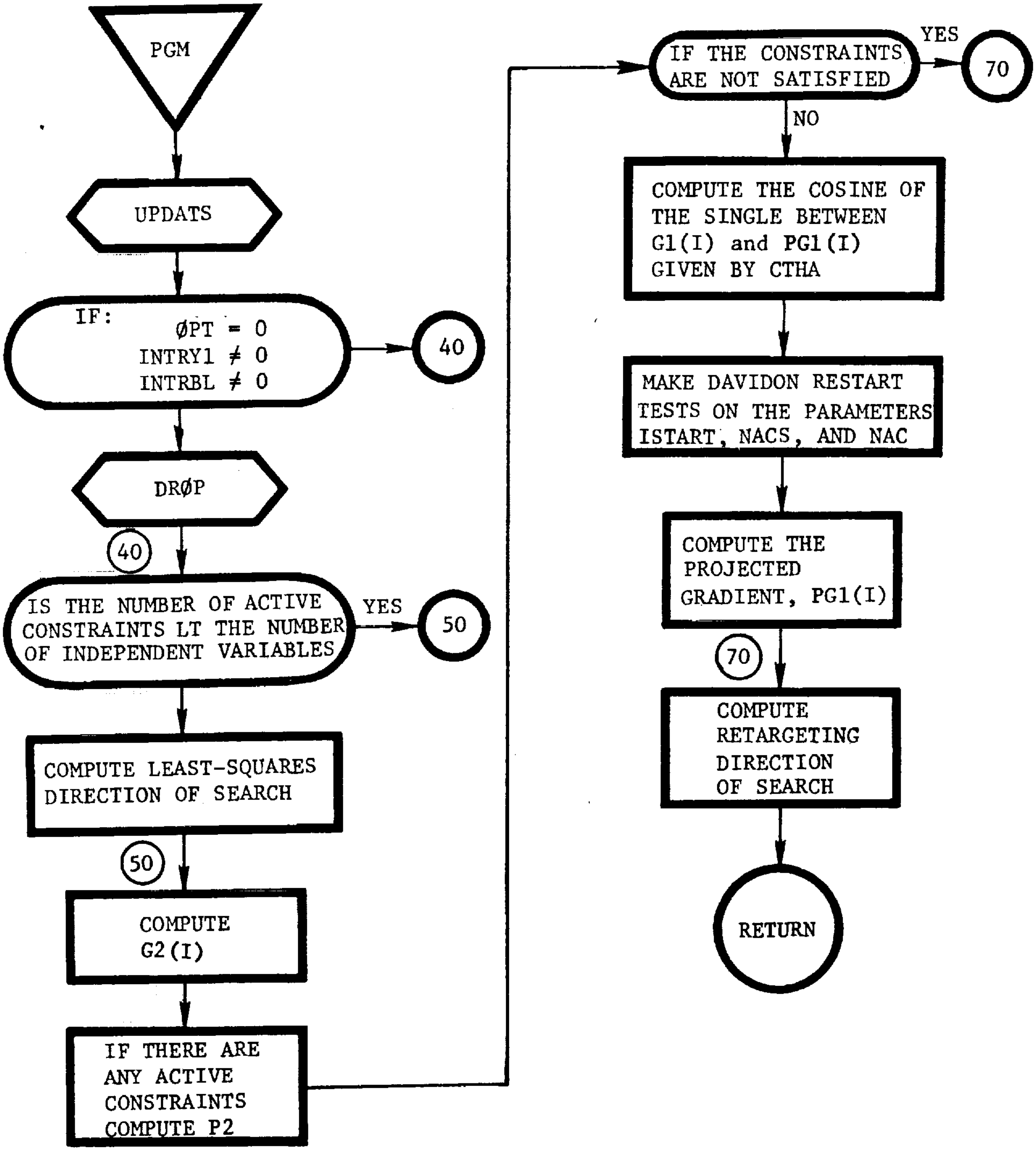
NOTE: NLINES = 48

PRECEDING PAGE BLANK NOT FILMED

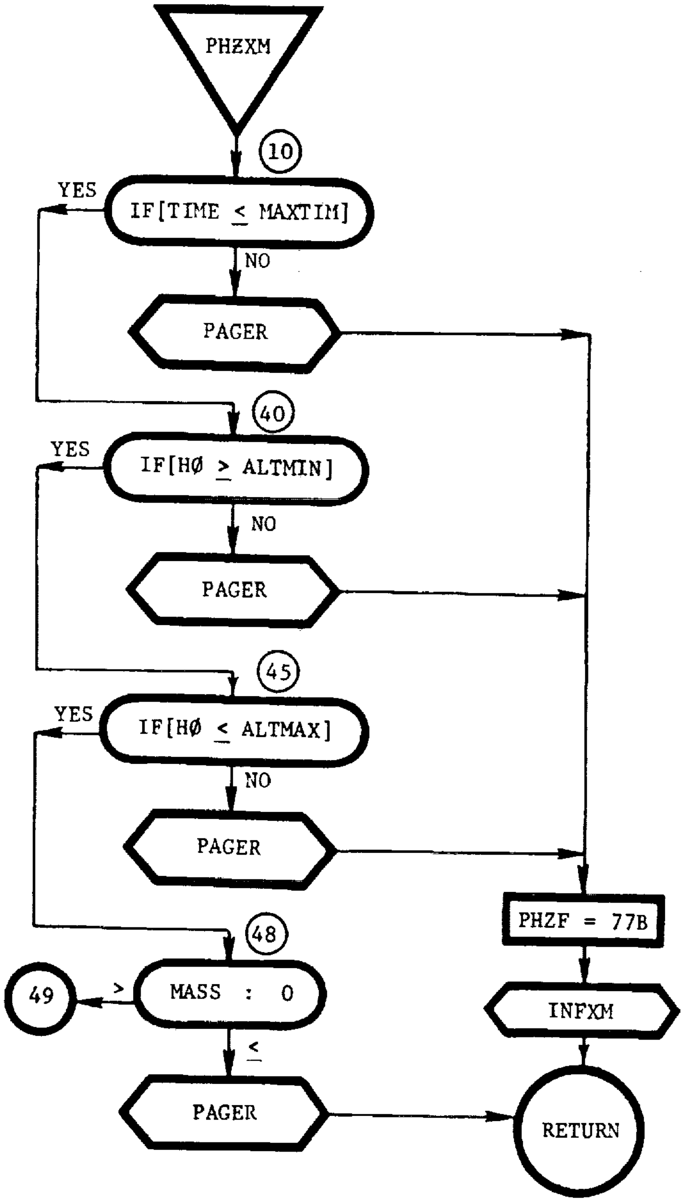
PBLØCK (N): This routine generates a summary print block for the option requested by the argument N.

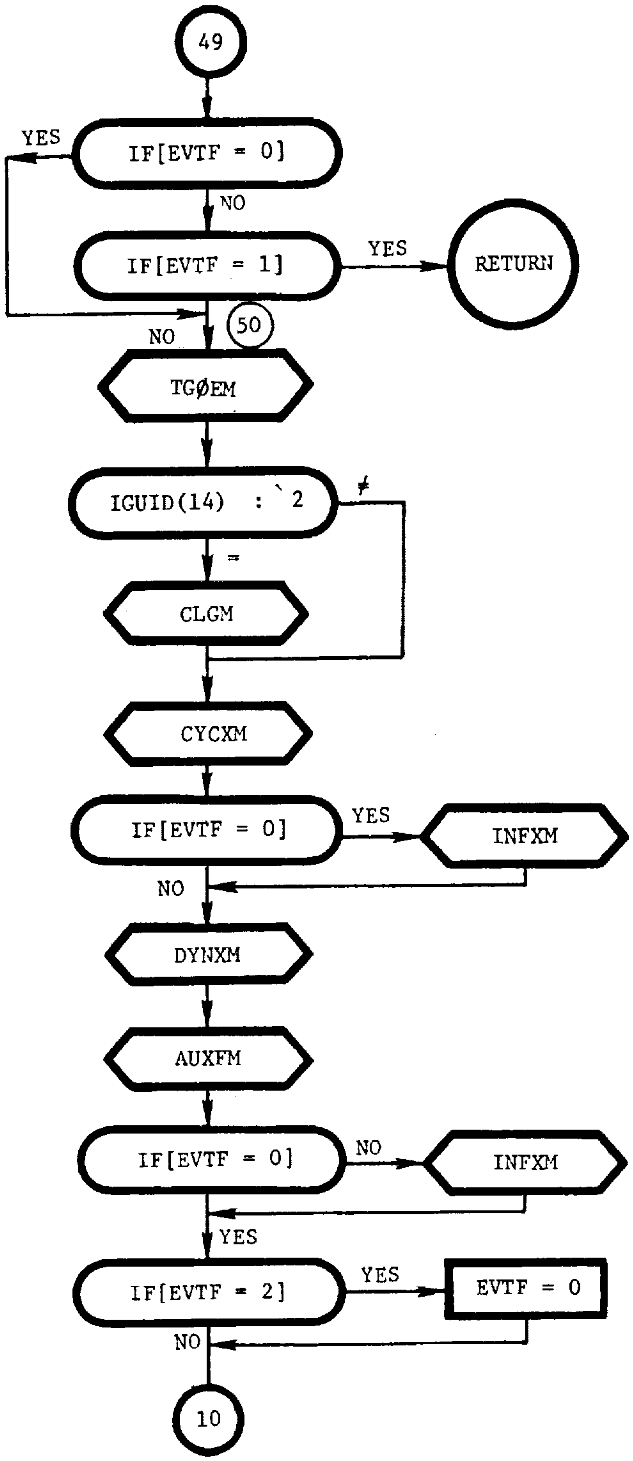


PGM: This routine determines the direction of search when the projected gradient method is used.

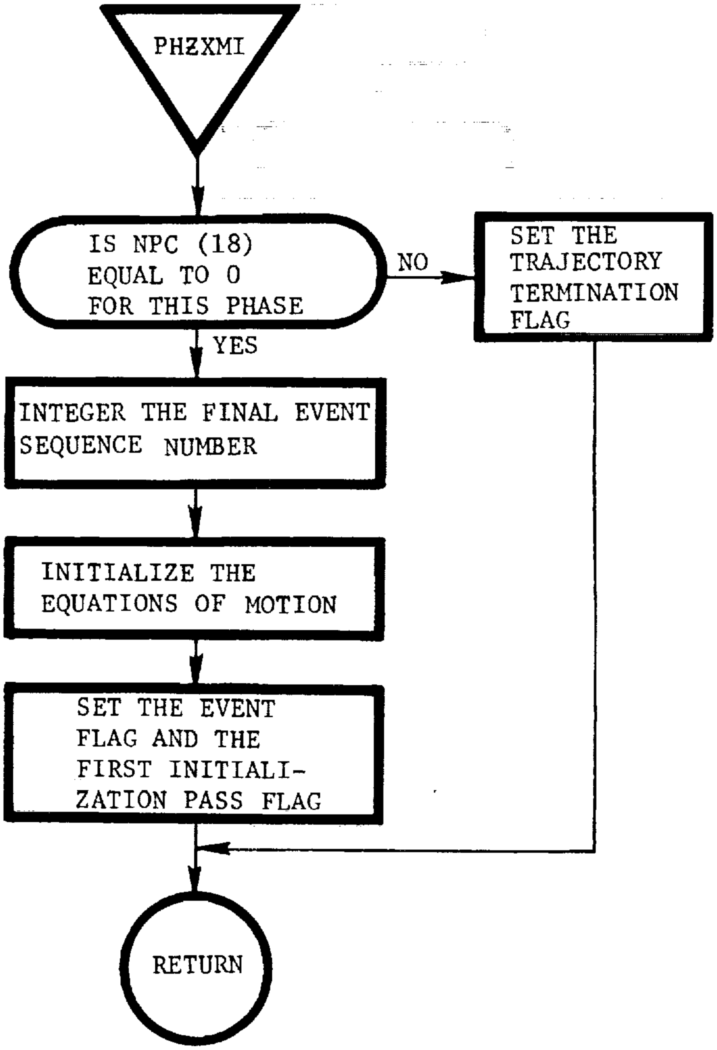

PGM
UPDATS
IF: ØPT = 0
INTRY1 ≠ 0
INTRBL ≠ 0
40
DRØP
40
IS THE NUMBER OF ACTIVE CONSTRAINTS LT THE NUMBER OF INDEPENDENT VARIABLES
YES
50
COMPUTE LEAST-SQUARES DIRECTION OF SEARCH
50
COMPUTE G2(I)
IF THERE ARE ANY ACTIVE CONSTRAINTS COMPUTE P2
IF THE CONSTRAINTS ARE NOT SATISFIED
YES
70
NO
COMPUTE THE COSINE OF THE SINGLE BETWEEN G1(I) and PG1(I) GIVEN BY CTHA
MAKE DAVIDON RESTART TESTS ON THE PARAMETERS ISTART, NACS, AND NAC
COMPUTE THE PROJECTED GRADIENT, PG1(I)
70
COMPUTE RETARGETING DIRECTION OF SEARCH
RETURN

PHZXM: This routine is the executive routine for overlay (2,3). It controls the integration of the equations of motion and determines whether the parameters MAXTIM, ALTMIN, or ALTMAX have been exceeded; if they have, the trajectory is terminated.

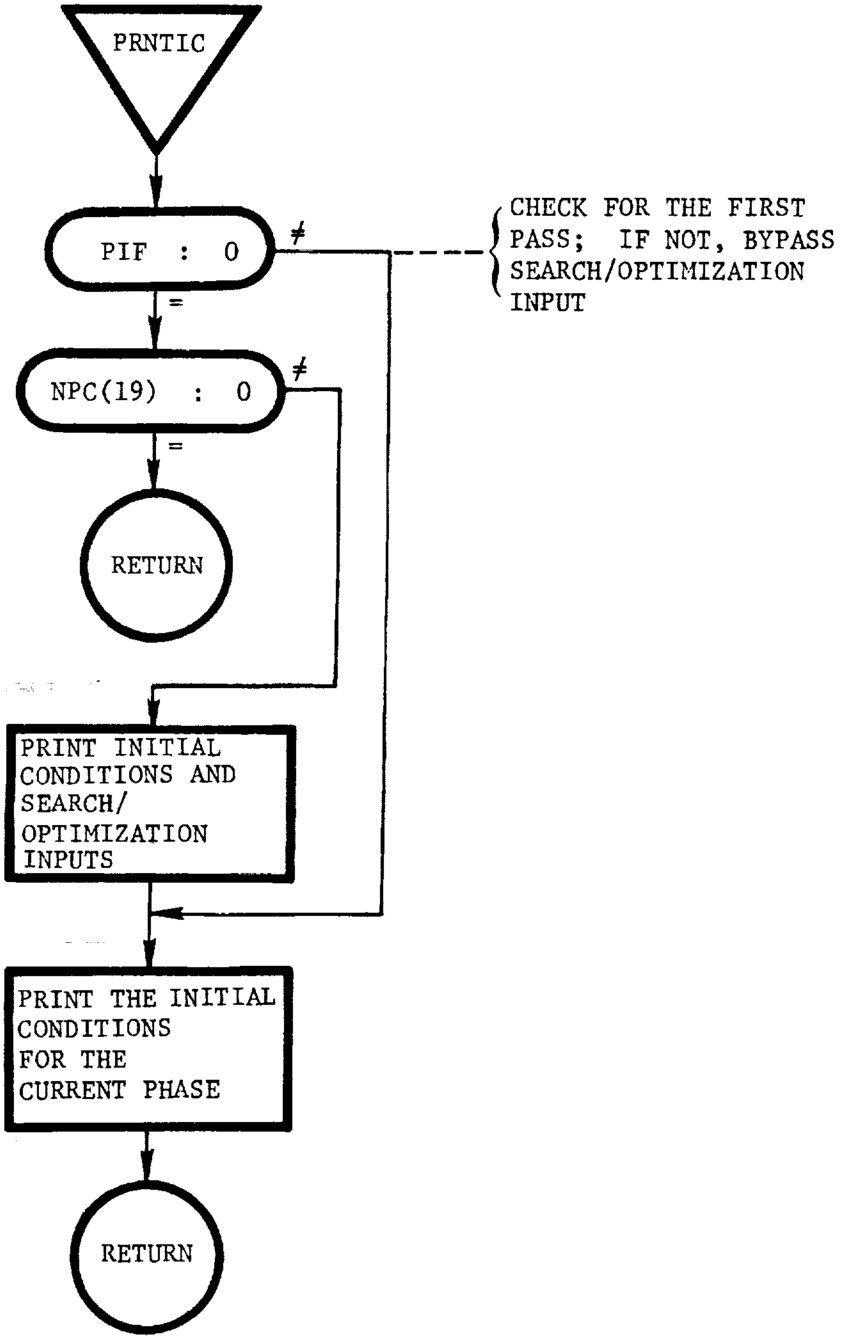



49
IF[EVTF = 0]
YES
NO
IF[EVTF = 1]
YES
RETURN
NO
50
TGØEM
IGUID(14) : 2
≠
=
CLGM
CYCXM
IF[EVTF = 0]
YES
INFXM
NO
DYNXM
AUXFM
IF[EVTF = 0]
NO
INFXM
YES
IF[EVTF = 2]
YES
EVTF = 0
NO
10

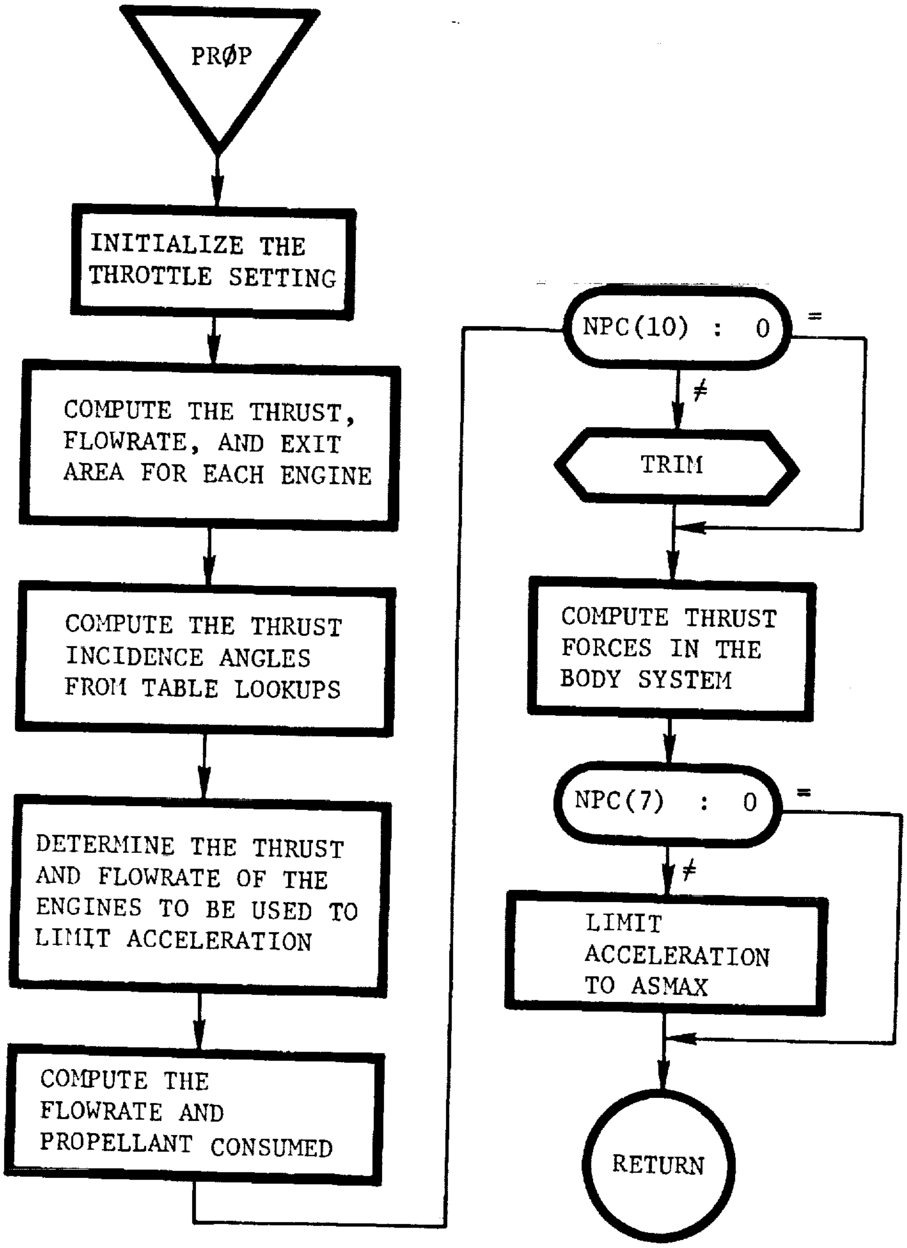
PHZXMI: This routine is the executive program of overlay (2,2). It performs the executive function for the phase initialization process.


PHZXMI
IS NPC (18) EQUAL TO 0 FOR THIS PHASE
NO
SET THE TRAJECTORY TERMINATION FLAG
YES
INTEGER THE FINAL EVENT SEQUENCE NUMBER
INITIALIZE THE EQUATIONS OF MOTION
SET THE EVENT FLAG AND THE FIRST INITIALIZATION PASS FLAG
RETURN

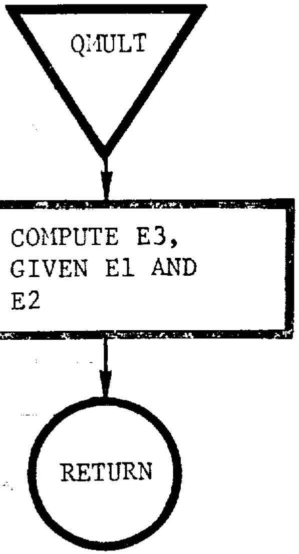
PRNTIC: This routine prints a summary of the initial conditions for the current phase.


PRNTIC
PIF : 0
≠
=
CHECK FOR THE FIRST PASS; IF NOT, BYPASS SEARCH/OPTIMIZATION INPUT
NPC(19) : 0
≠
=
RETURN
PRINT INITIAL CONDITIONS AND SEARCH/ OPTIMIZATION INPUTS
PRINT THE INITIAL CONDITIONS FOR THE CURRENT PHASE
RETURN

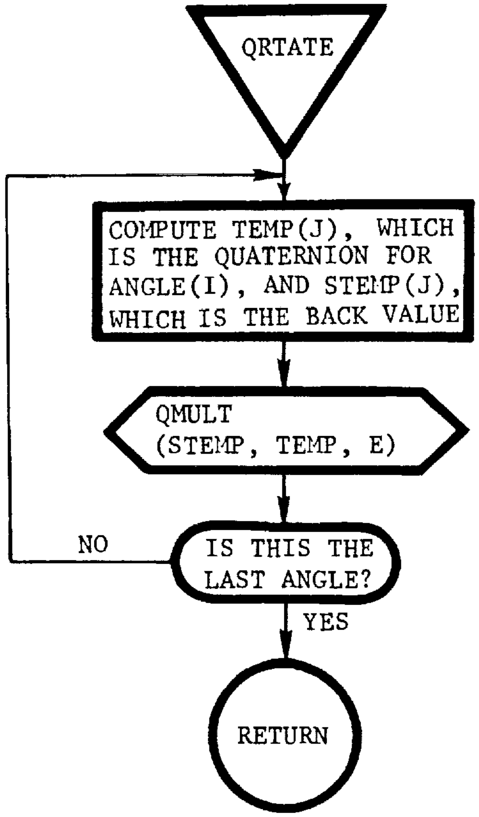
PRØP: This routine calculates the thrust forces in the body-coordinate system, based on the use of either rocket or jet engines. The routine also determines the value of the throttling parameter required to limit the acceleration when NPC(7) = 1.


PRØP
INITIALIZE THE THROTTLE SETTING
COMPUTE THE THRUST, FLOWRATE, AND EXIT AREA FOR EACH ENGINE
COMPUTE THE THRUST INCIDENCE ANGLES FROM TABLE LOOKUPS
DETERMINE THE THRUST AND FLOWRATE OF THE ENGINES TO BE USED TO LIMIT ACCELERATION
COMPUTE THE FLOWRATE AND PROPELLANT CONSUMED
NPC(10) : 0
=
≠
TRIM
COMPUTE THRUST FORCES IN THE BODY SYSTEM
NPC(7) : 0
=
≠
LIMIT ACCELERATION TO ASMAX
RETURN

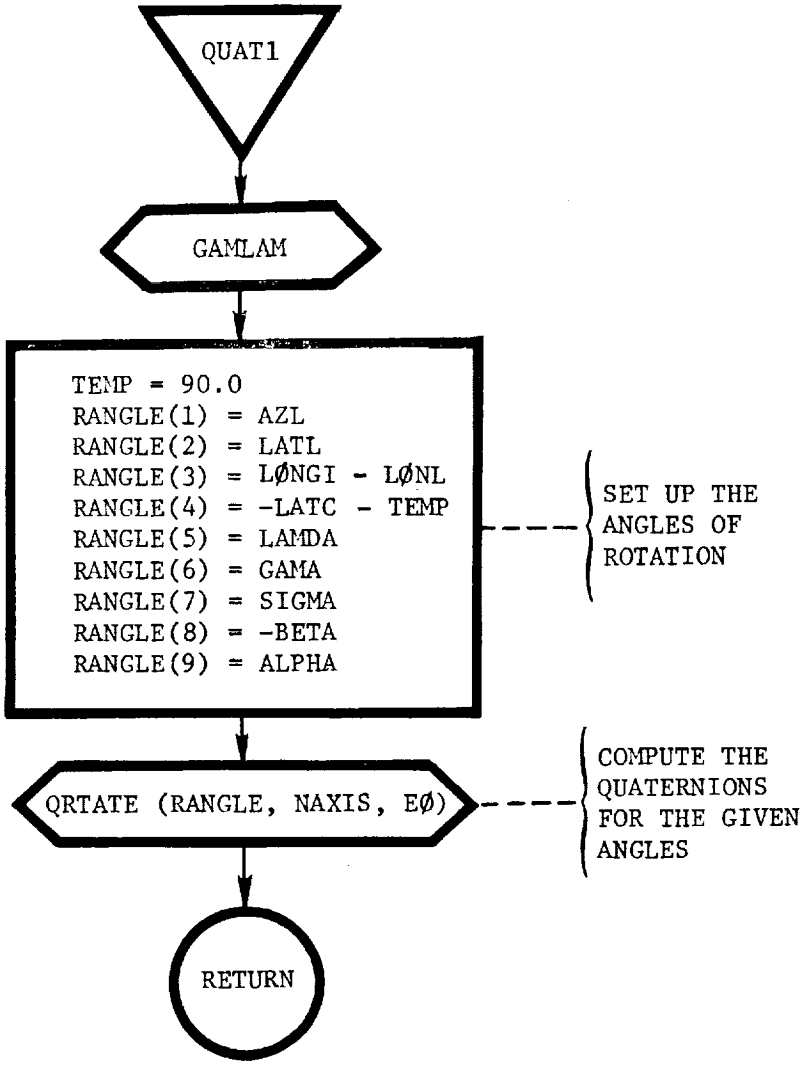
QMULT: This routine multiplies quaternion El and quaternion E2 to yield quaternion E3.



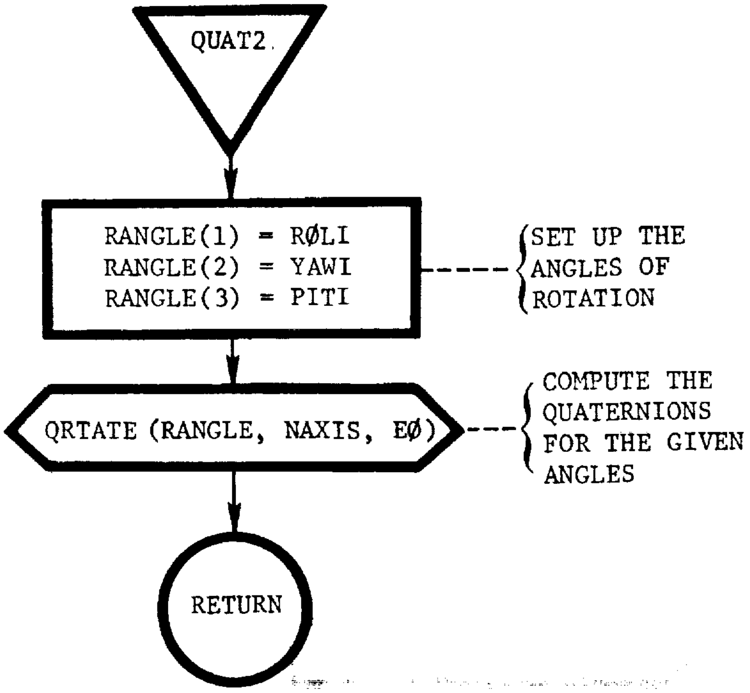
QRTATE: This routine computes the quaternion E that results from rotating through the specified angles, ANGLE, about the specified axes, NAXIS.


QRTATE
COMPUTE TEMP(J), WHICH IS THE QUATERNION FOR ANGLE(I), AND STEMP(J), WHICH IS THE BACK VALUE
QMULT (STEMP, TEMP, E)
IS THIS THE LAST ANGLE?
NO
YES
RETURN

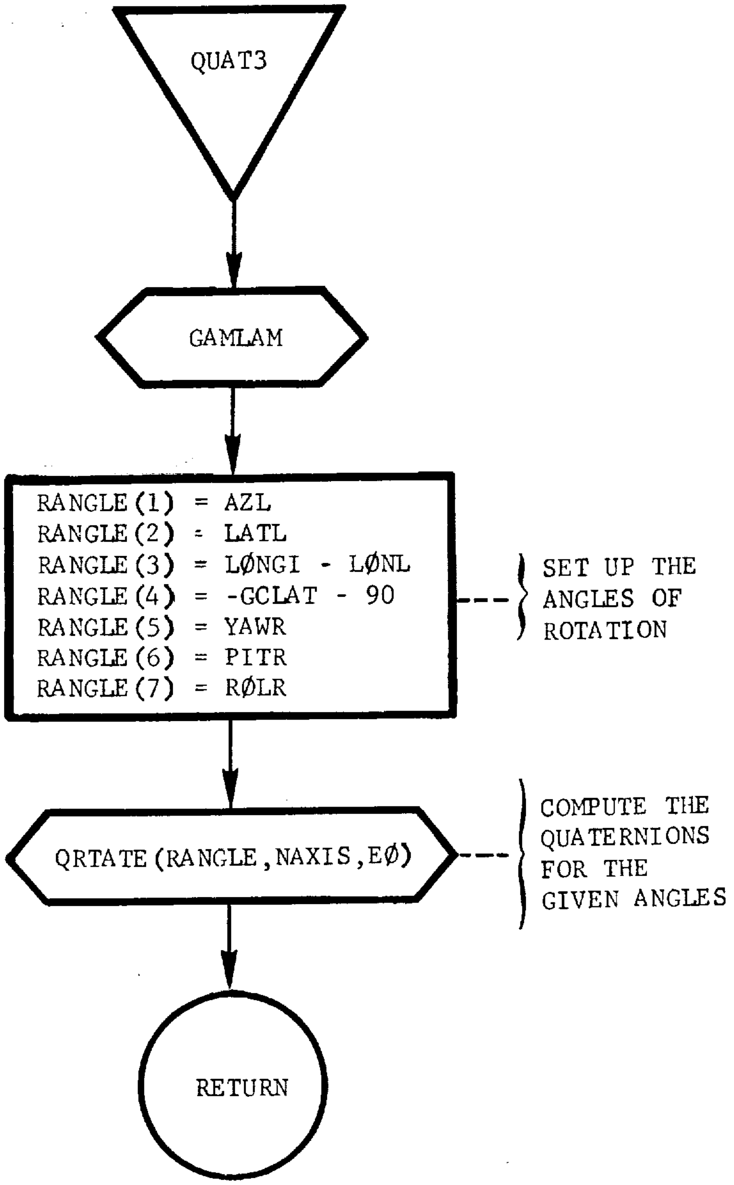
QUAT1: This routine computes the quaternions, given the angle of attack, sideslip, and bank angle.



QUAT2: This routine computes the quaternions, given the local attitude angles in yaw, pitch, and roll.


QUAT2
RANGLE(1) = RØLI
RANGLE(2) = YAWI
RANGLE(3) = PITI
SET UP THE ANGLES OF ROTATION
QRTATE (RANGLE, NAXIS, EØ)
COMPUTE THE QUATERNIONS FOR THE GIVEN ANGLES
RETURN

QUAT3: This routine computes the quaternions given the relative euler angles.



READAT: This routine performs the actual processing of the input data. Subroutines RSEARCH, RGENDAT, RTBLMLT, and RTAB are called as required to perform the actual reading of namelists "SEARCH", "GENDAT", "TBLMLT", and "TAB", respectively. The namelists are always read in a given sequence that can be terminated at any point by setting ENDPHS = 1. The calling sequence for reading the namelists is: RSEARCH, RGENDAT, RTBLMLT, and RTAB.

After reading each namelist, the data for that phase are packed into one of two data buffers, depending on the type of data being processed. The two data buffers are broken down as follows:

1) IGEN: 1500 decimal cells of storage for all input varibles except for event criteria and input tables for all phases. The table multipliers are also stored in this array.

2) IBKT: 1500 decimal cells of storage for the event criteria and tables for all phases.

The actual storage detail is shown in Table 2. The input variables are stored in sequence as they are read and the data are grouped according to phase number.

TABLE 2

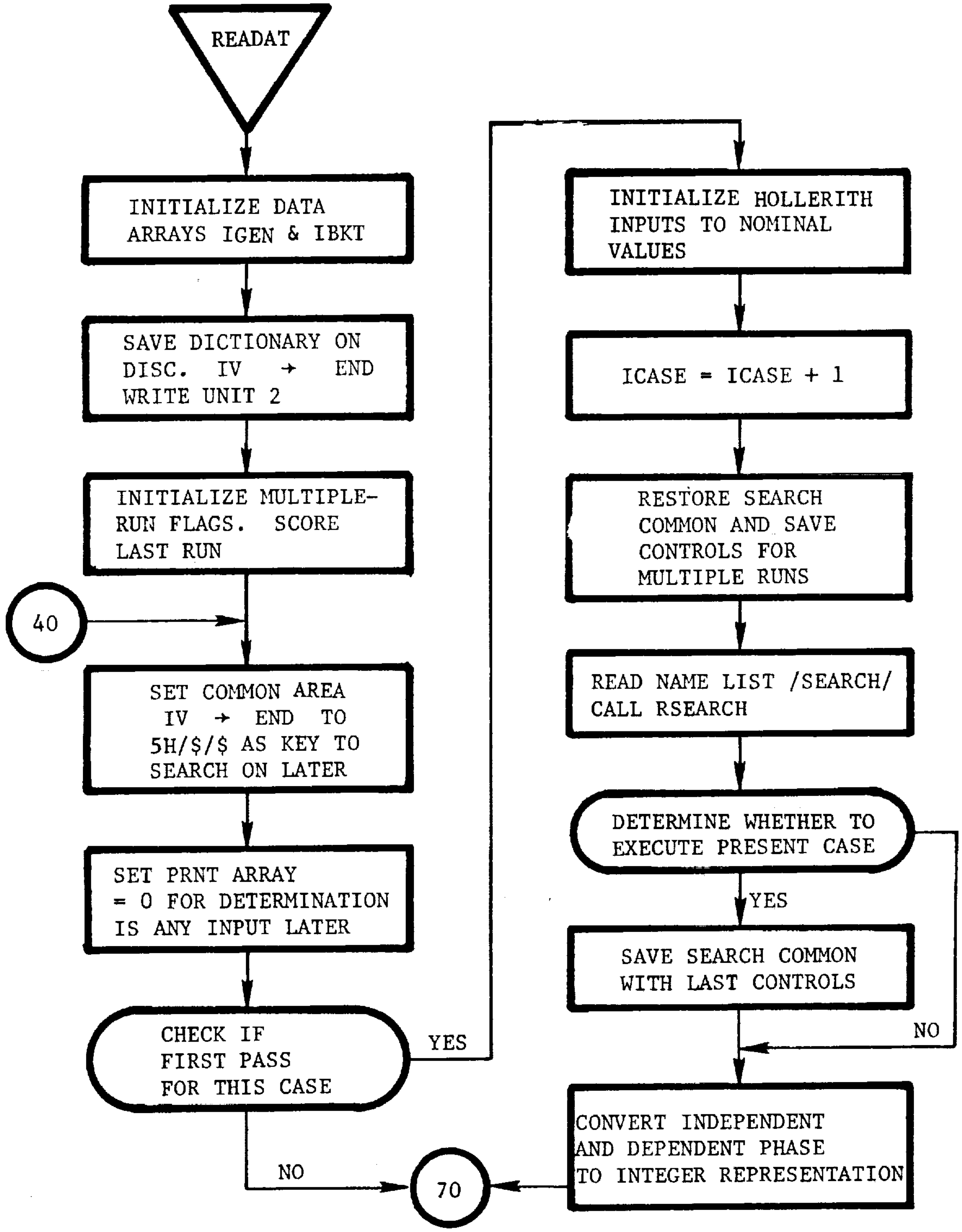
CONTENTS OF DATA BUFFERS

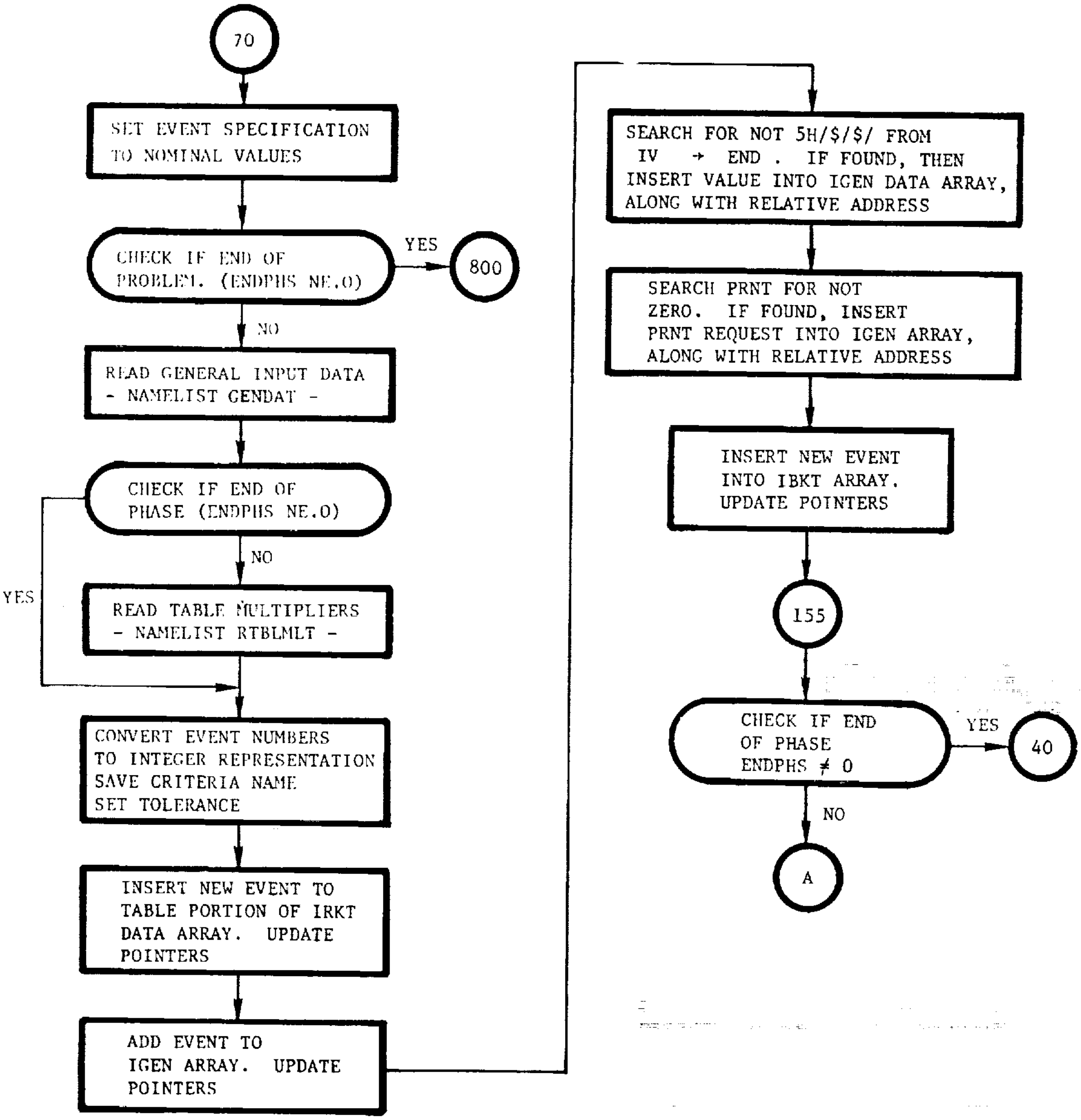
Location	BUFFER "IGEN" (1500 decimal cells)	Location	BUFFER "IBKT" (1500 decimal cells)
1	number of cells occupied in IGEN	1	number of cells occupied in IBKT
2	number of cells occupied by data for the first phase (NG1)	2	number of cells occupied by event sequence data (NB1)
3	event sequence number for the first phase	3	event sequence number for the first phase
4	address of the first variable following the event sequence number in IGEN	4	event type for the first phase
5	value of the first variable in IGEN	5	event criteria for the first phase
6	unused cell associated with the first variable stored in IGEN	6	criteria value for the first phase
7 thru NG1	variables stored by repeating the sequence shown in 4, 5, 6 for each variable	7	derivative name of the event criteria variable for the first phase
		8	tolerance
NG1+1	number of cells occupied by data for the second phase (NG2)	9	model number
NG1+2	repeat sequence 3 through 7-NG2 as before for the second event	10	unused cell associated with the event criteria for the first event
		11 thru NB1	repeat sequence 3-10 for remaining events

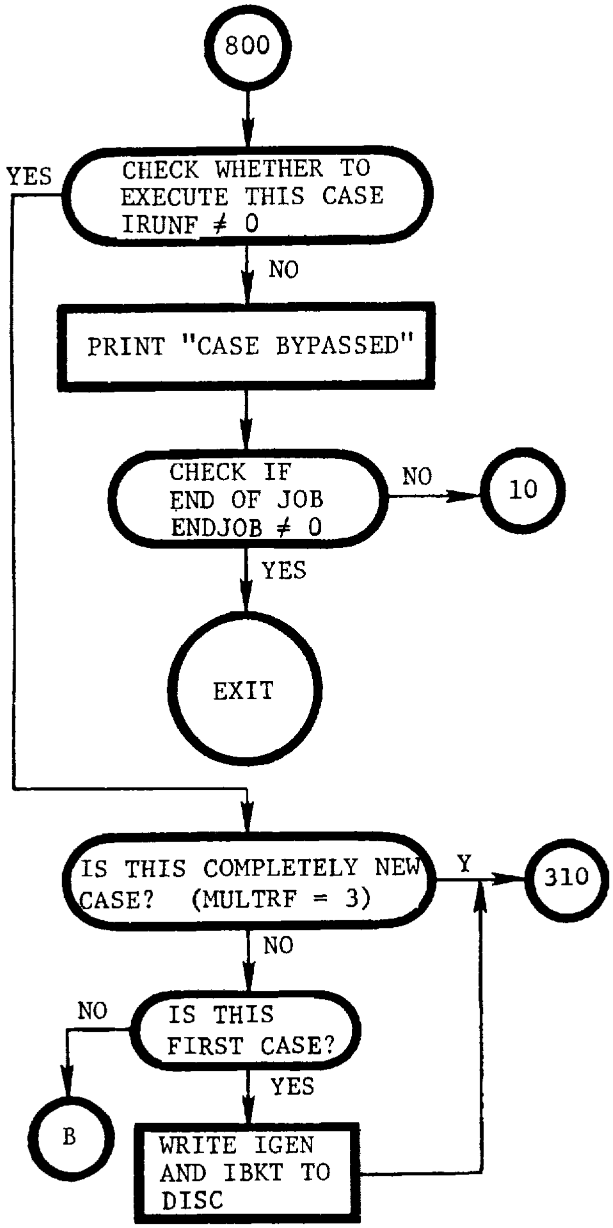
TABLE 2

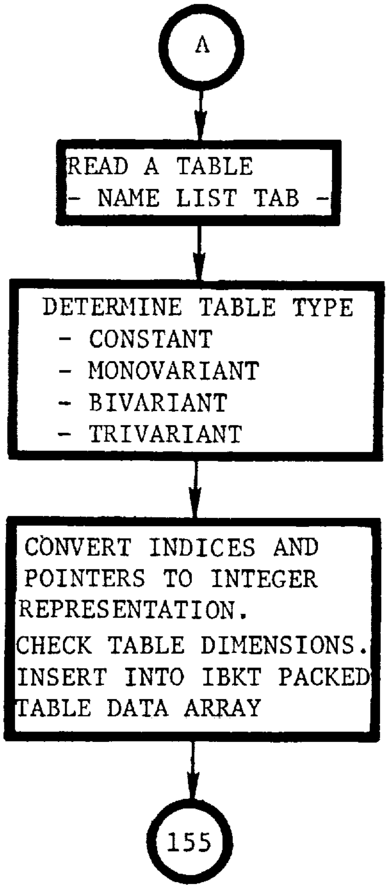
CONTENTS OF DATA BUFFERS - CONCLUDED

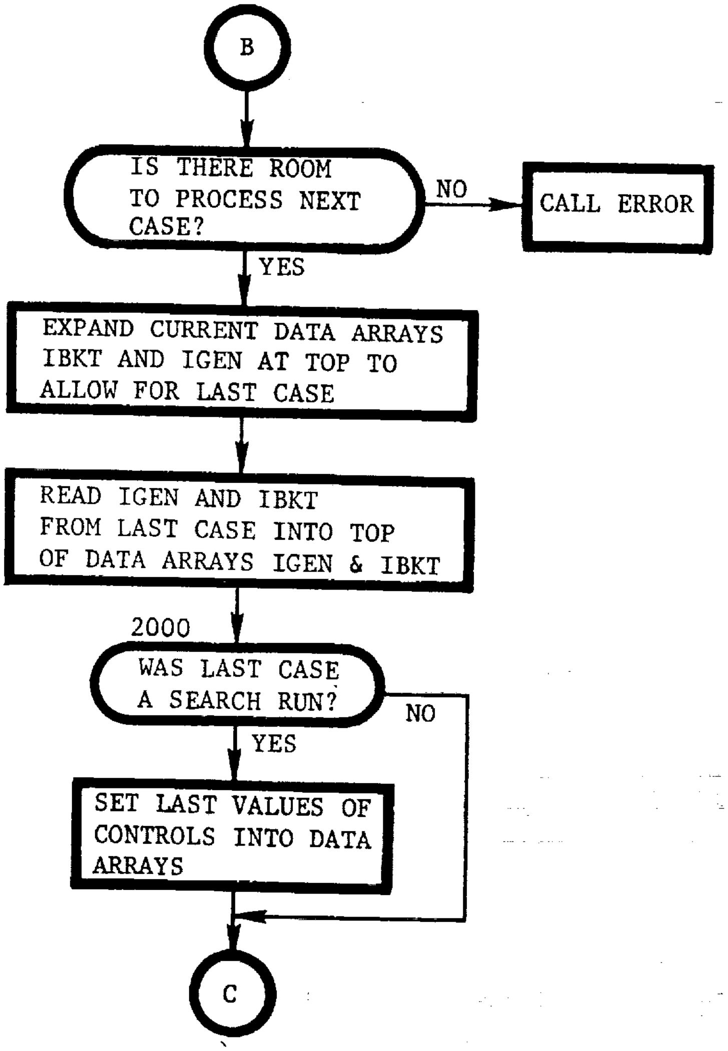
Location	BUFFER "IGEN"	Location	BUFFER "IBKT"
		NB1 + 1	number of cells occupied by all tables for the first phase (NB2)
		NB1 + 2	phase number associated with the first set of tables
		NB1 + 3	size of the first table to be input
		NB1 + 4	name of the first table (HOLLERITH)
		NB1 + 5	table pointers and values
			repeat above sequence for each table in the first phase (NB1 + 3 thru NB1 + 5)
			repeat above sequence for all phases (NB1 + 1 thru NB1 + 5)

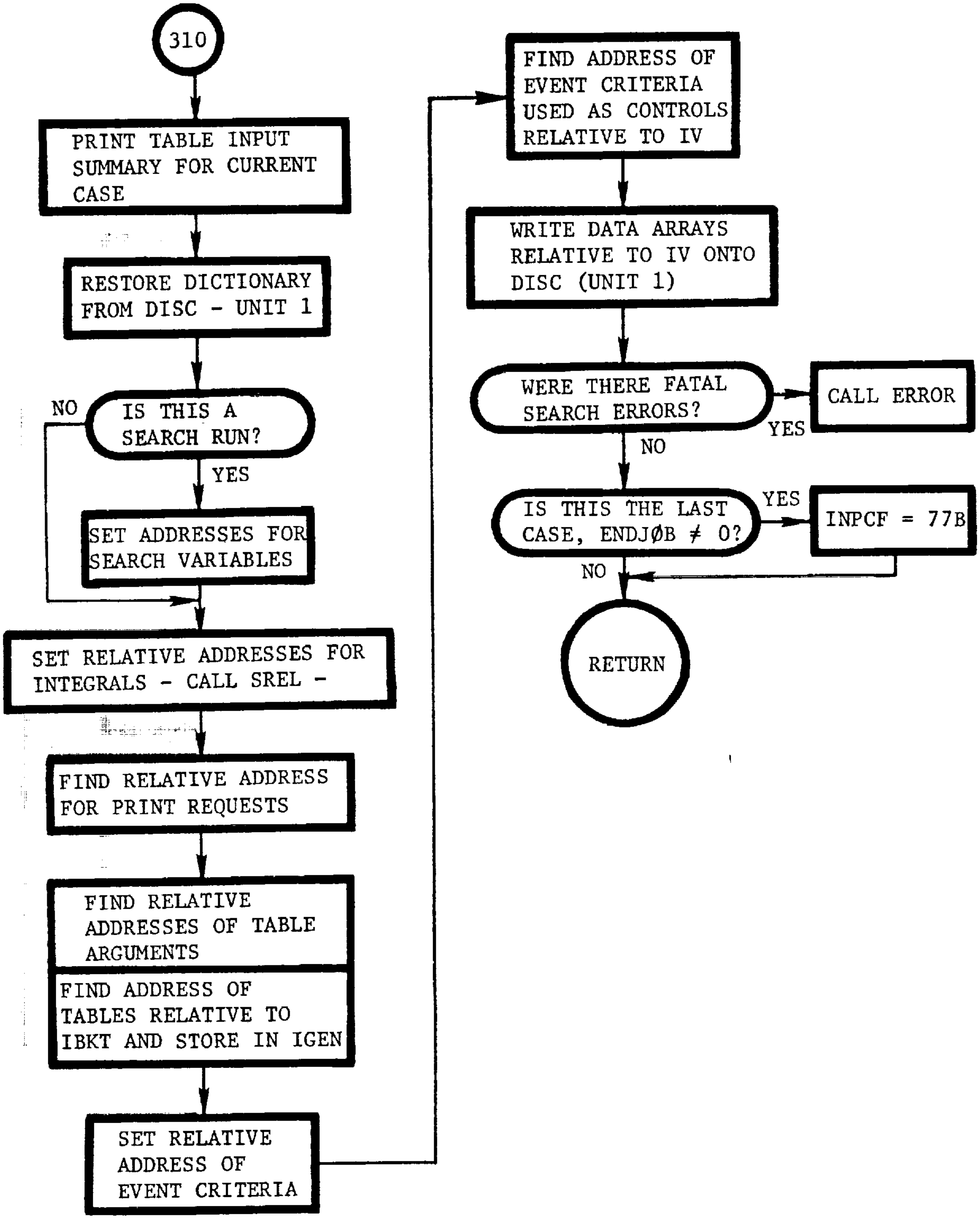

READAT
INITIALIZE DATA ARRAYS IGEN & IBKT
SAVE DICTIONARY ON DISC. IV → END WRITE UNIT 2
INITIALIZE MULTIPLE-RUN FLAGS. SCORE LAST RUN
40
SET COMMON AREA IV → END TO 5H/$/$ AS KEY TO SEARCH ON LATER
SET PRNT ARRAY = 0 FOR DETERMINATION IS ANY INPUT LATER
CHECK IF FIRST PASS FOR THIS CASE
YES
NO
70
INITIALIZE HOLLERITH INPUTS TO NOMINAL VALUES
ICASE = ICASE + 1
RESTORE SEARCH COMMON AND SAVE CONTROLS FOR MULTIPLE RUNS
READ NAME LIST /SEARCH/ CALL RSEARCH
DETERMINE WHETHER TO EXECUTE PRESENT CASE
YES
SAVE SEARCH COMMON WITH LAST CONTROLS
NO
CONVERT INDEPENDENT AND DEPENDENT PHASE TO INTEGER REPRESENTATION

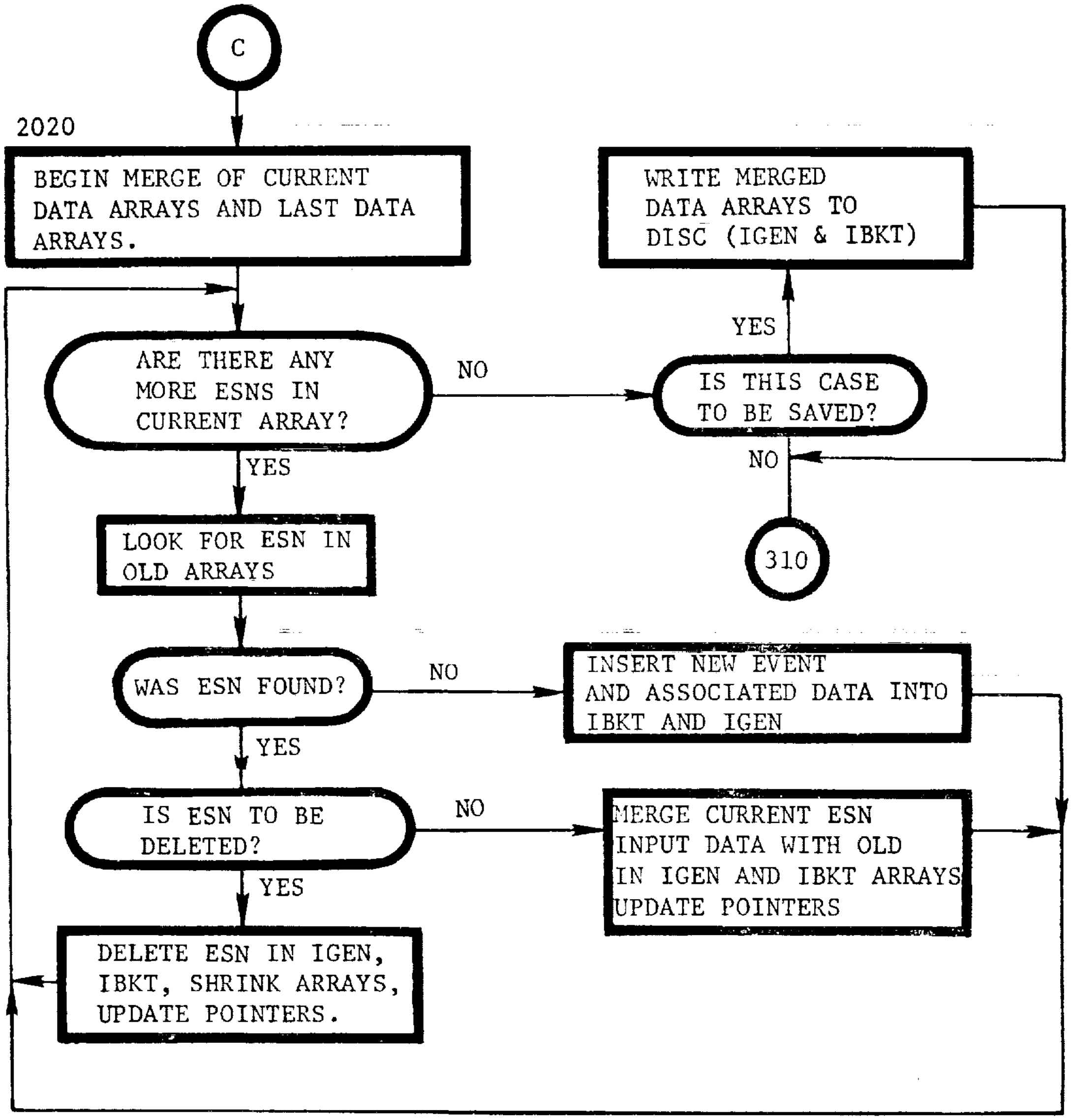

70
SET EVENT SPECIFICATION TO NOMINAL VALUES
CHECK IF END OF PROBLEM. (ENDPHS NE.0)
YES
800
NO
READ GENERAL INPUT DATA - NAMELIST GENDAT -
CHECK IF END OF PHASE (ENDPHS NE.0)
YES
NO
READ TABLE MULTIPLIERS - NAMELIST RTBLMLT -
CONVERT EVENT NUMBERS TO INTEGER REPRESENTATION SAVE CRITERIA NAME SET TOLERANCE
INSERT NEW EVENT TO TABLE PORTION OF IRKT DATA ARRAY. UPDATE POINTERS
ADD EVENT TO IGEN ARRAY. UPDATE POINTERS
SEARCH FOR NOT 5H/$/$/ FROM IV → END . IF FOUND, THEN INSERT VALUE INTO IGEN DATA ARRAY, ALONG WITH RELATIVE ADDRESS
SEARCH PRNT FOR NOT ZERO. IF FOUND, INSERT PRNT REQUEST INTO IGEN ARRAY, ALONG WITH RELATIVE ADDRESS
INSERT NEW EVENT INTO IBKT ARRAY. UPDATE POINTERS
155
CHECK IF END OF PHASE ENDPHS ≠ 0
YES
40
NO
A


800
YES
CHECK WHETHER TO EXECUTE THIS CASE IRUNF ≠ 0
NO
PRINT "CASE BYPASSED"
CHECK IF END OF JOB ENDJOB ≠ 0
NO
10
YES
EXIT
IS THIS COMPLETELY NEW CASE? (MULTRF = 3)
Y
310
NO
NO
IS THIS FIRST CASE?
YES
B
WRITE IGEN AND IBKT TO DISC

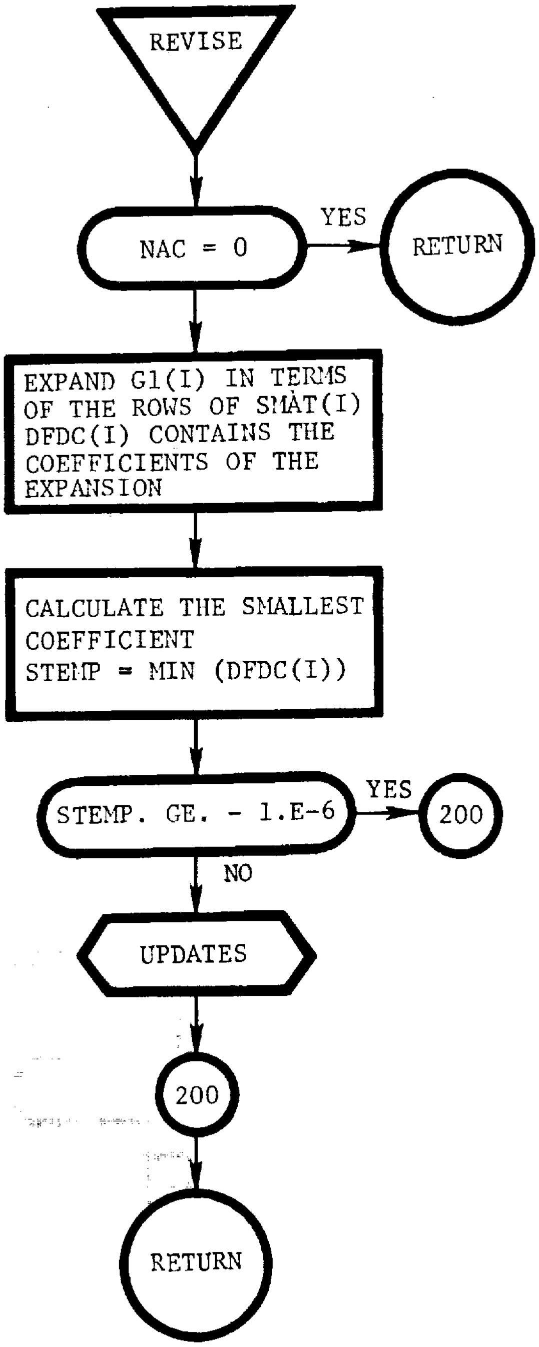

A
READ A TABLE - NAME LIST TAB -
DETERMINE TABLE TYPE
- CONSTANT
- MONOVARIANT
- BIVARIANT
- TRIVARIANT
CONVERT INDICES AND POINTERS TO INTEGER REPRESENTATION.
CHECK TABLE DIMENSIONS.
INSERT INTO IBKT PACKED TABLE DATA ARRAY
155


B
IS THERE ROOM TO PROCESS NEXT CASE?
NO
CALL ERROR
YES
EXPAND CURRENT DATA ARRAYS IBKT AND IGEN AT TOP TO ALLOW FOR LAST CASE
READ IGEN AND IBKT FROM LAST CASE INTO TOP OF DATA ARRAYS IGEN & IBKT
2000
WAS LAST CASE A SEARCH RUN?
NO
YES
SET LAST VALUES OF CONTROLS INTO DATA ARRAYS
C

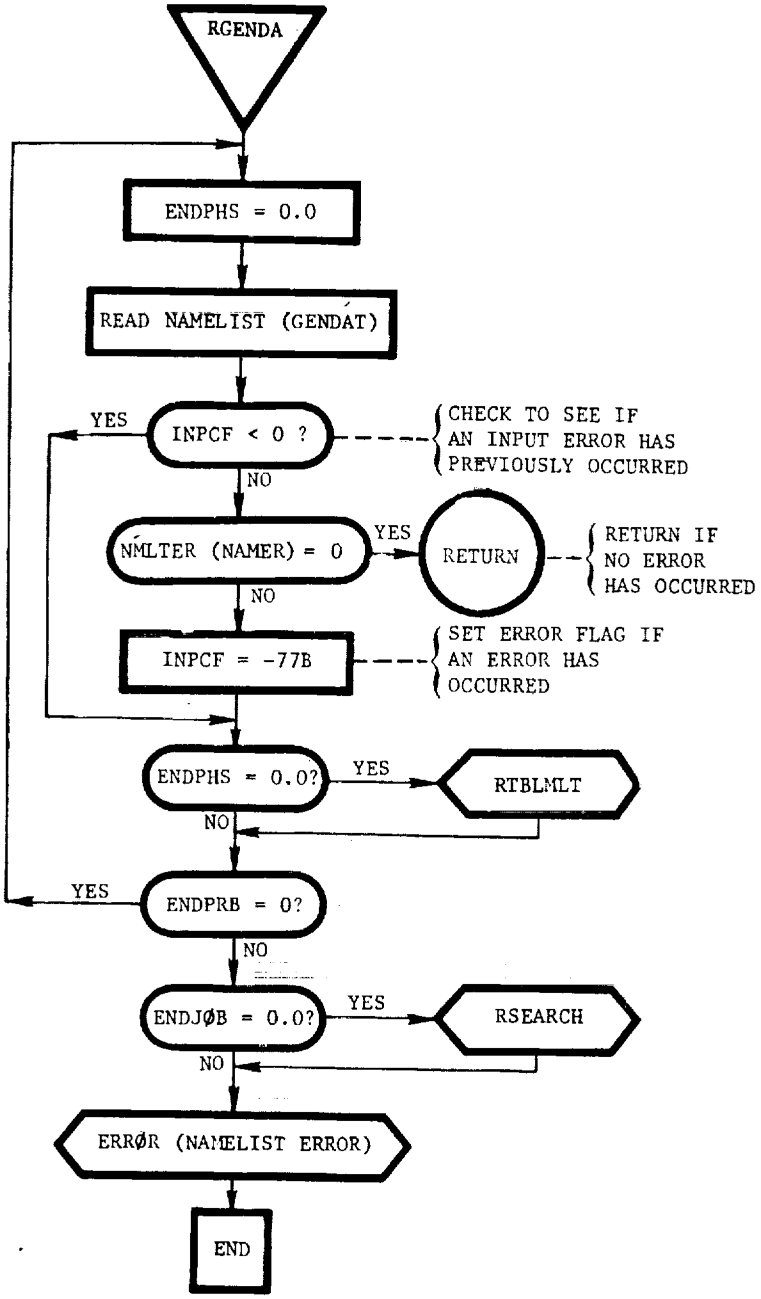

310
PRINT TABLE INPUT SUMMARY FOR CURRENT CASE
RESTORE DICTIONARY FROM DISC - UNIT 1
IS THIS A SEARCH RUN?
NO
YES
SET ADDRESSES FOR SEARCH VARIABLES
SET RELATIVE ADDRESSES FOR INTEGRALS - CALL SREL -
FIND RELATIVE ADDRESS FOR PRINT REQUESTS
FIND RELATIVE ADDRESSES OF TABLE ARGUMENTS
FIND ADDRESS OF TABLES RELATIVE TO IBKT AND STORE IN IGEN
SET RELATIVE ADDRESS OF EVENT CRITERIA
FIND ADDRESS OF EVENT CRITERIA USED AS CONTROLS RELATIVE TO IV
WRITE DATA ARRAYS RELATIVE TO IV ONTO DISC (UNIT 1)
WERE THERE FATAL SEARCH ERRORS?
YES
CALL ERROR
NO
IS THIS THE LAST CASE, ENDJØB ≠ 0?
YES
INPCF = 77B
NO
RETURN


C
2020
BEGIN MERGE OF CURRENT DATA ARRAYS AND LAST DATA ARRAYS.
ARE THERE ANY MORE ESNS IN CURRENT ARRAY?
NO
YES
LOOK FOR ESN IN OLD ARRAYS
WAS ESN FOUND?
NO
YES
IS ESN TO BE DELETED?
NO
YES
DELETE ESN IN IGEN, IBKT, SHRINK ARRAYS, UPDATE POINTERS.
WRITE MERGED DATA ARRAYS TO DISC (IGEN & IBKT)
YES
IS THIS CASE TO BE SAVED?
NO
310
INSERT NEW EVENT AND ASSOCIATED DATA INTO IBKT AND IGEN
MERGE CURRENT ESN INPUT DATA WITH OLD IN IGEN AND IBKT ARRAYS UPDATE POINTERS

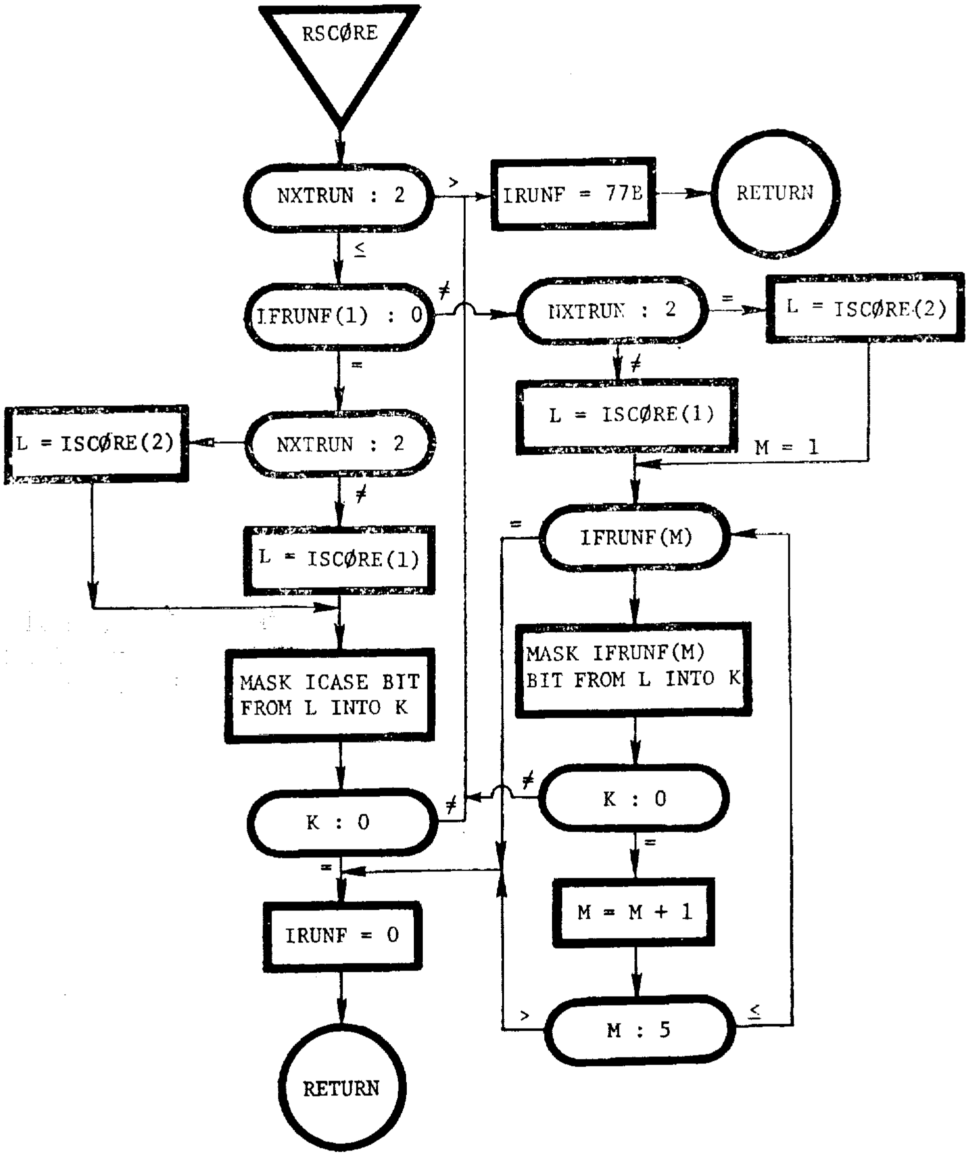
REVISE: This routine determines the indices of active constraints that are candidates to be dropped. This is done by expanding G1(I) in terms of the elements of the sensitivity matrix. The constraint with the most negative coefficient DFDC(I) is a candidate to be dropped.


REVISE
NAC = 0
YES
RETURN
EXPAND G1(I) IN TERMS OF THE ROWS OF SMAT(I) DFDC(I) CONTAINS THE COEFFICIENTS OF THE EXPANSION
CALCULATE THE SMALLEST COEFFICIENT STEMP = MIN (DFDC(I))
STEMP. GE. - 1.E-6
YES
200
NO
UPDATES
200
RETURN

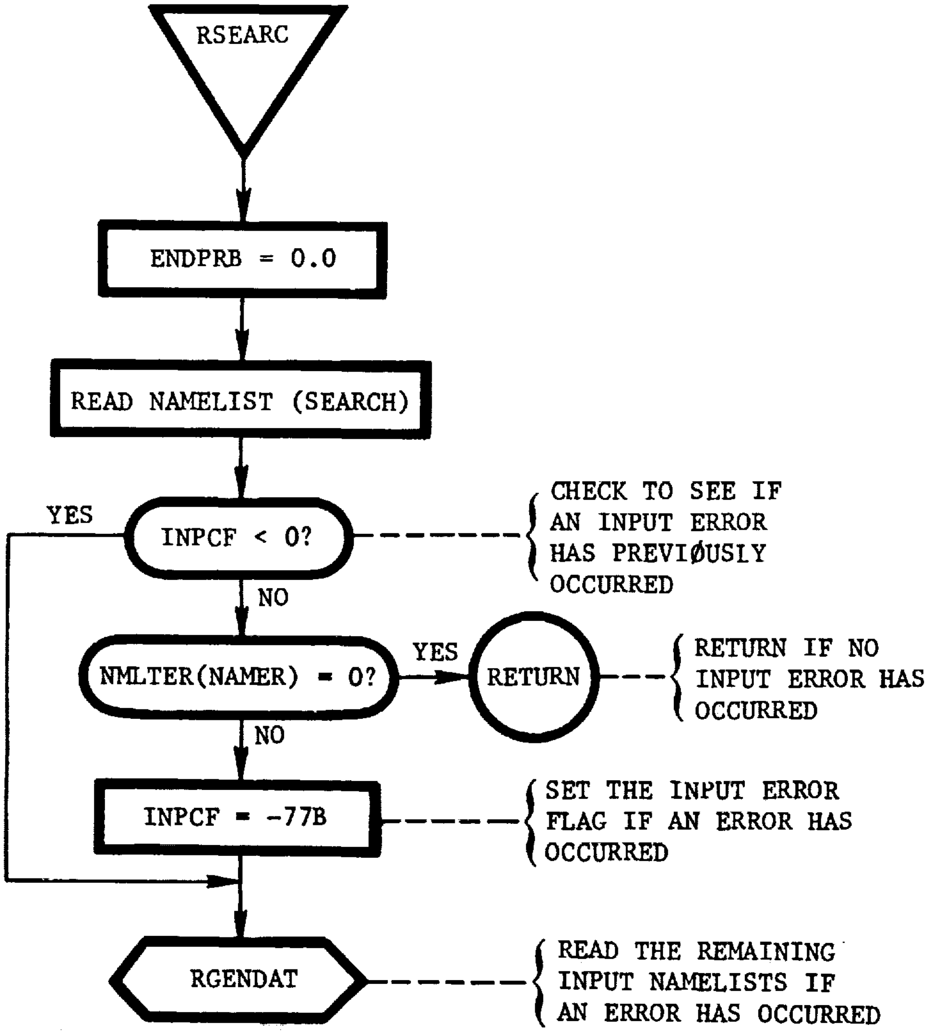
RGENDA: This routine reads namelist "GENDAT" and checks for any namelist errors.



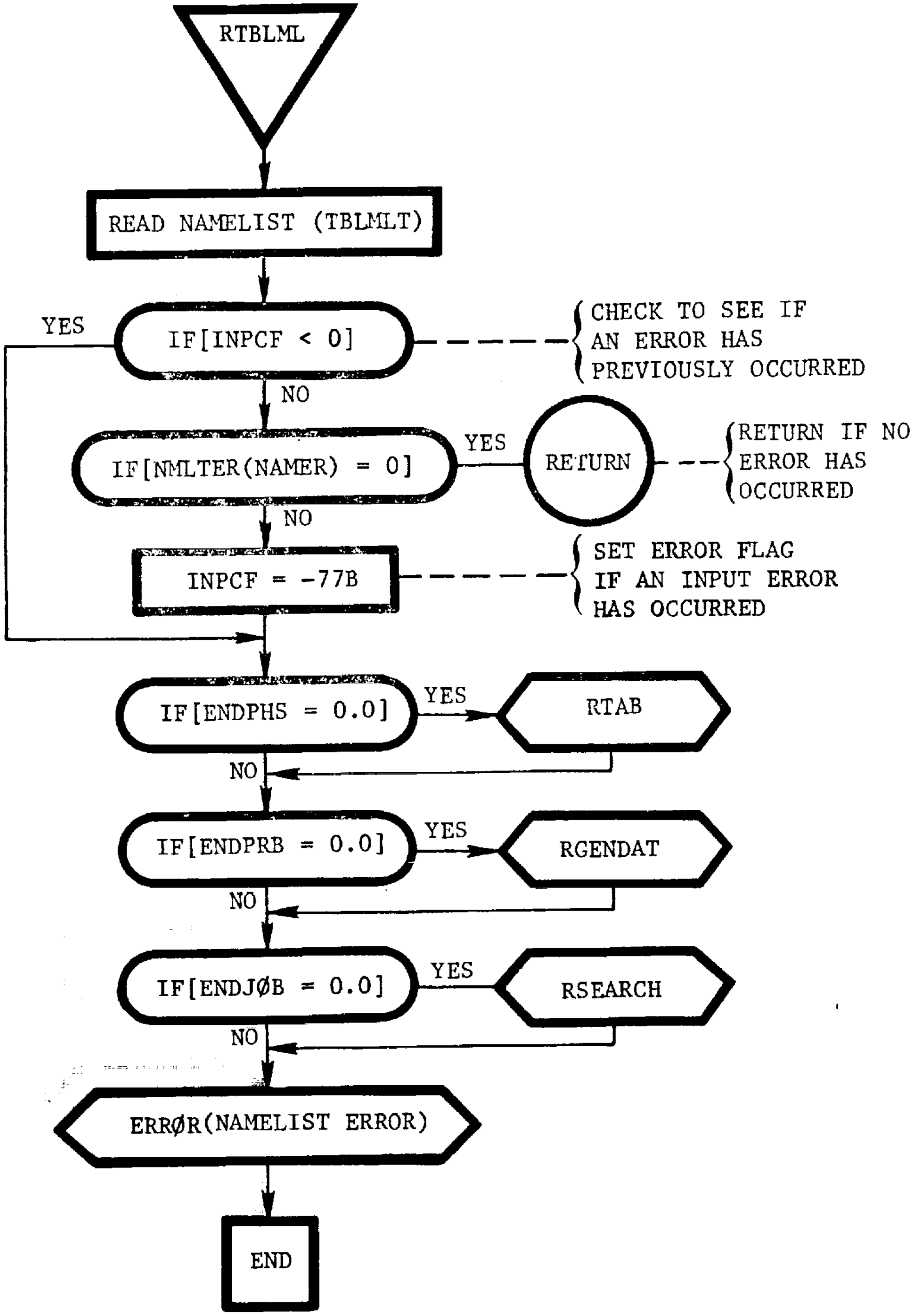
RSCØRE: This routine checks NXTRUN, IFRUNF(i), and ISCØRE to determine if criteria for executing the next case are met. A history of success or failure of previous cases are packed into ISCØRE by bit position. IRUNF is returned nonzero if the next case is to be executed.



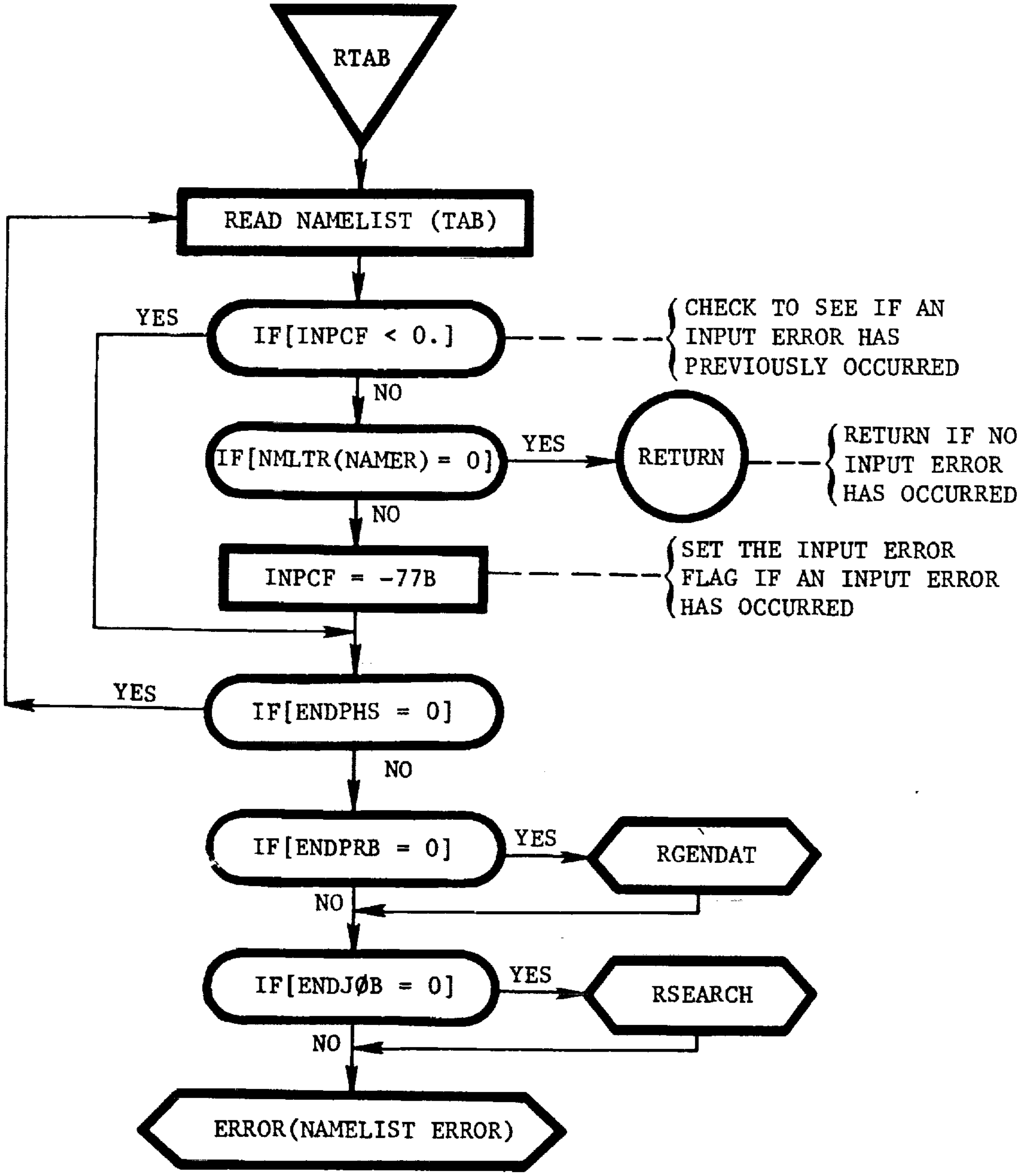
RSEARC: This routine reads namelist "SEARCH" and checks for any namelist errors.


RSEARC
ENDPRB = 0.0
READ NAMELIST (SEARCH)
INPCF < 0?
YES
NO
CHECK TO SEE IF AN INPUT ERROR HAS PREVIØUSLY OCCURRED
NMLTER(NAMER) = 0?
YES
RETURN
RETURN IF NO INPUT ERROR HAS OCCURRED
NO
INPCF = -77B
SET THE INPUT ERROR FLAG IF AN ERROR HAS OCCURRED
RGENDAT
READ THE REMAINING INPUT NAMELISTS IF AN ERROR HAS OCCURRED

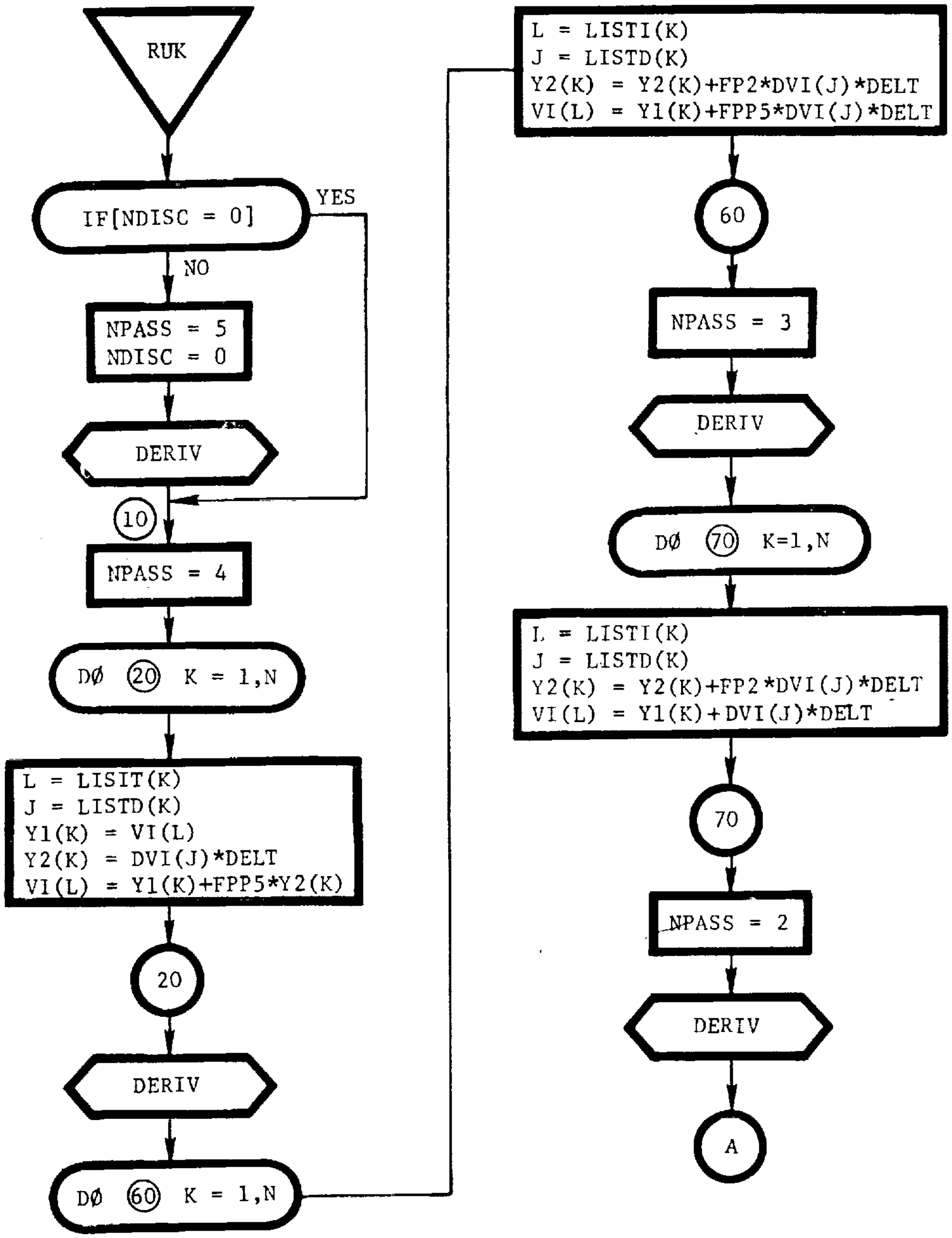
RTBLML: This routine reads namelist "TBLMLT" and checks for any namelist errors.

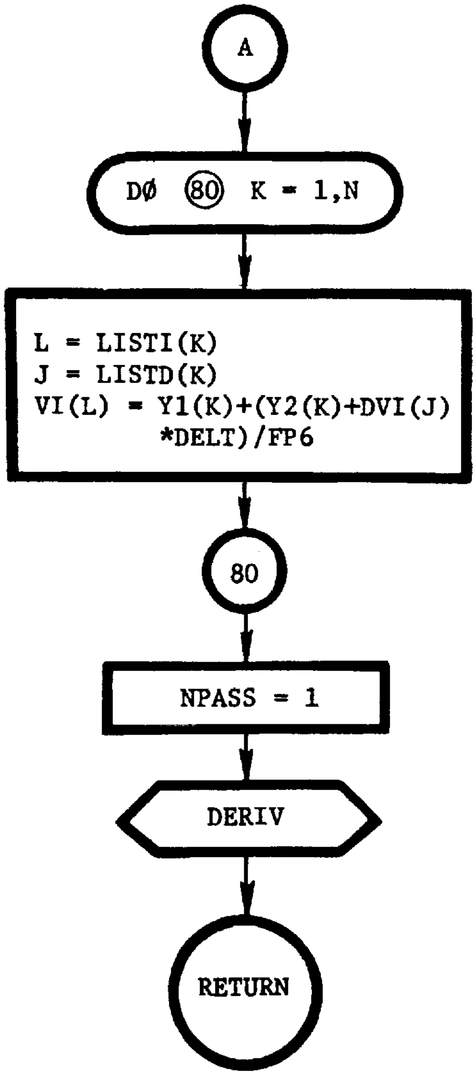


RTAB: This routine reads namelist "TAB" and checks for any namelist errors.

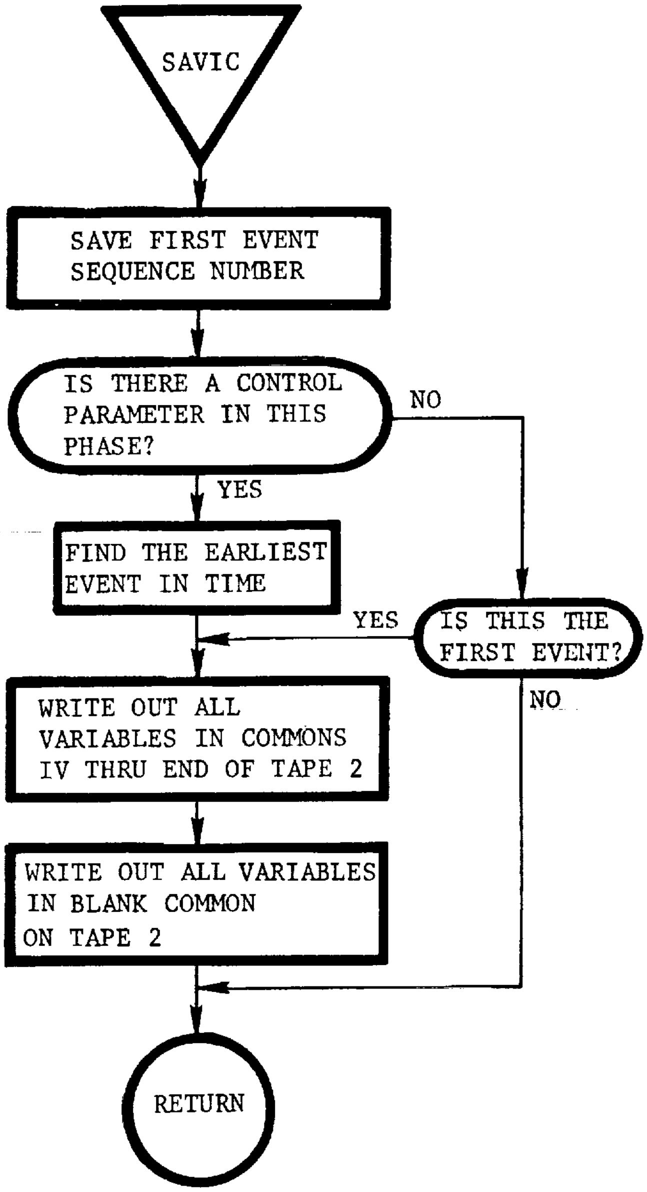


RUK: This routine contains the Runge-Kutta integration algorithm.

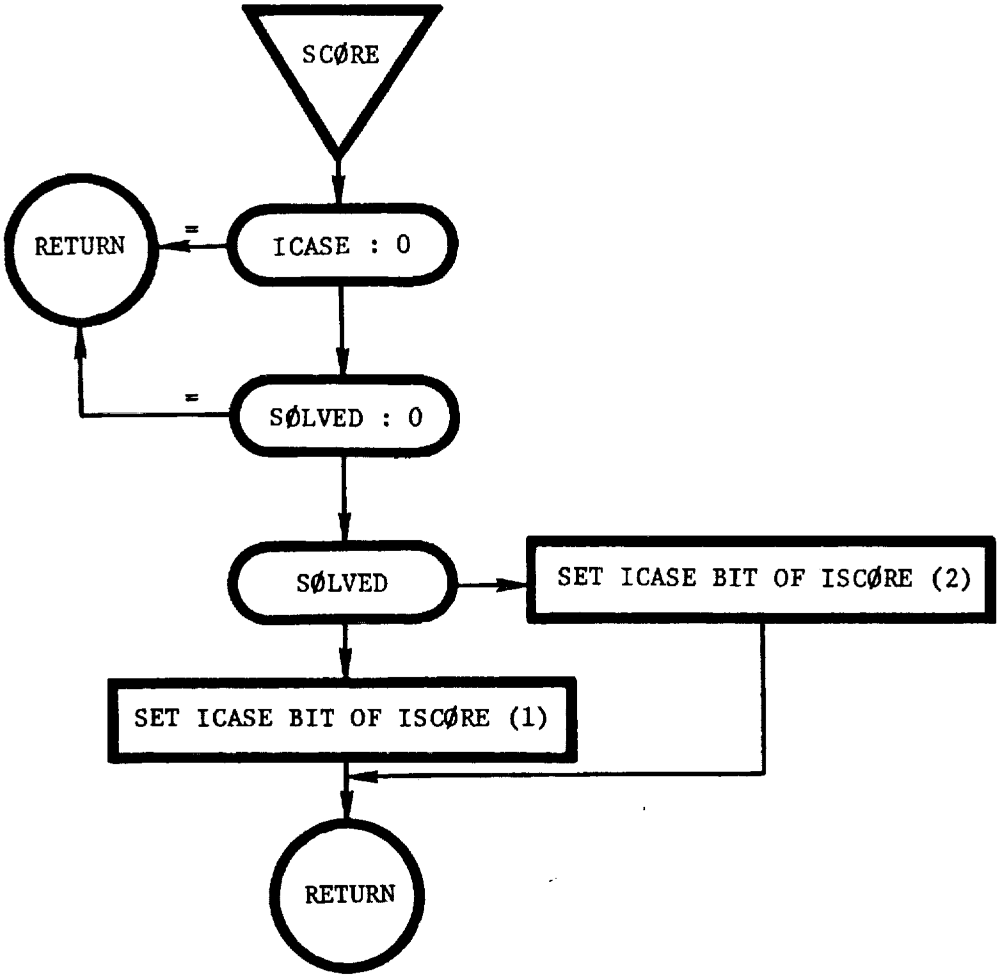



A
DØ 80 K = 1,N
L = LISTI(K)
J = LISTD(K)
VI(L) = Y1(K)+(Y2(K)+DVI(J)
*DELT)/FP6
80
NPASS = 1
DERIV
RETURN

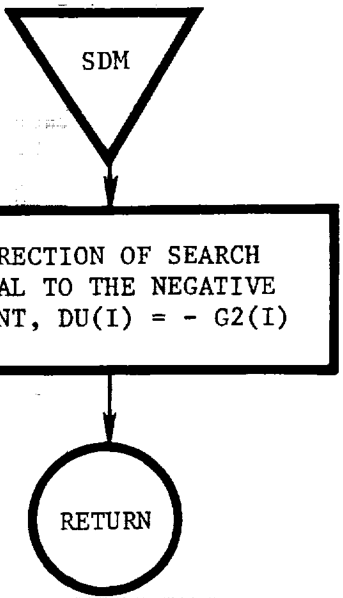
SAVIC: This routine buffers out commons IV and IBKT, which contain the state conditions, to file 2 at the beginning of each phase that contains an independent control variable. This information is used in running perturbed trajectories in the discrete-parameter mode.


SAVIC
SAVE FIRST EVENT SEQUENCE NUMBER
IS THERE A CONTROL PARAMETER IN THIS PHASE?
NO
YES
FIND THE EARLIEST EVENT IN TIME
YES
IS THIS THE FIRST EVENT?
NO
WRITE OUT ALL VARIABLES IN COMMONS IV THRU END OF TAPE 2
WRITE OUT ALL VARIABLES IN BLANK COMMON ON TAPE 2
RETURN

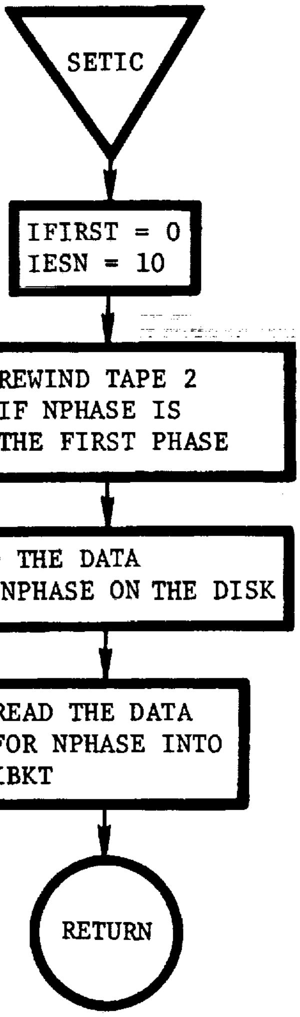
SCØRE: This routine sets the ICASE bit of ISCØRE(1) or ISCØRE(2) if SØLVED is + or -, respectively. No bit is set if SØLVED = 0.



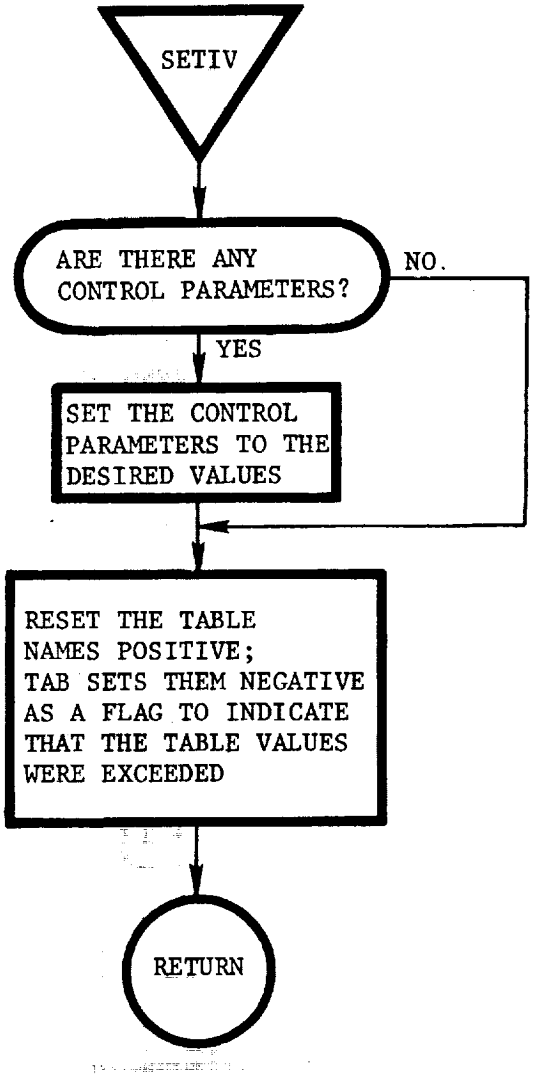
SDM: This routine computes the direction of search for minimizing P2 via the classic steepest-descent method.


SDM
SET DIRECTION OF SEARCH IS EQUAL TO THE NEGATIVE GRADIENT, DU(I) = - G2(I)
RETURN

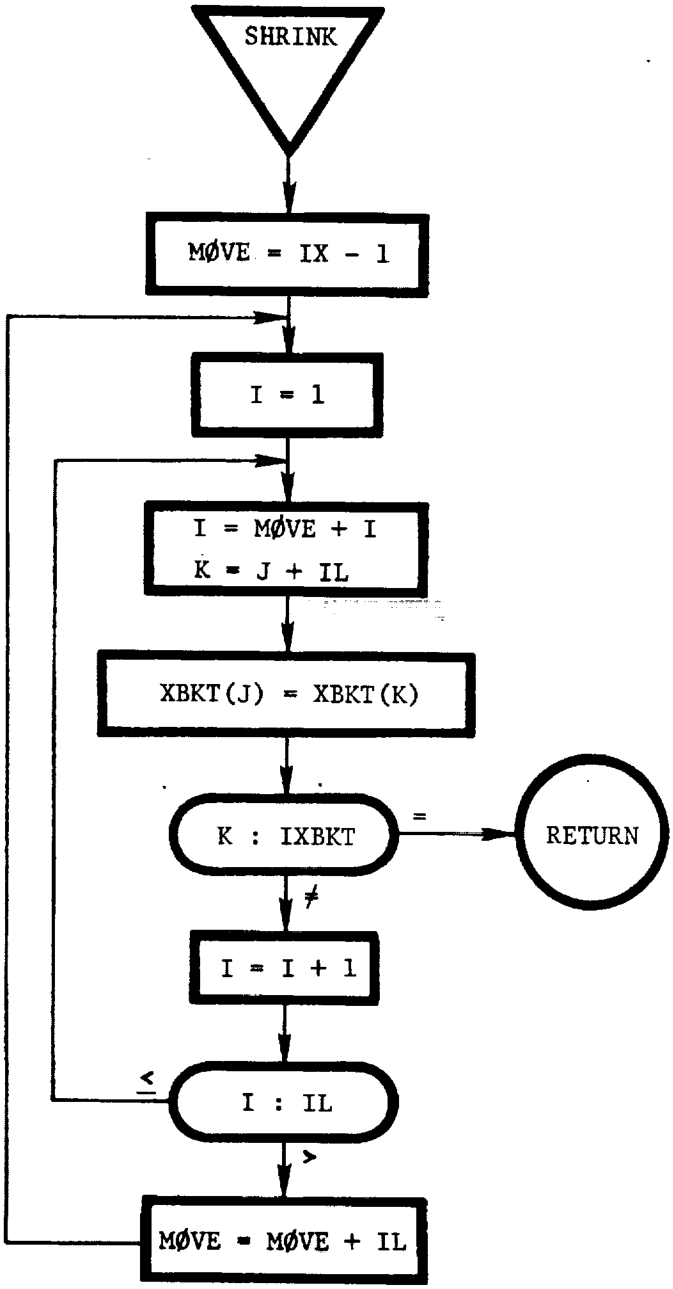
SETIC (NPHASE): This routine resets the initial conditions for the specified phase (NPHASE) equal to their nominal values.



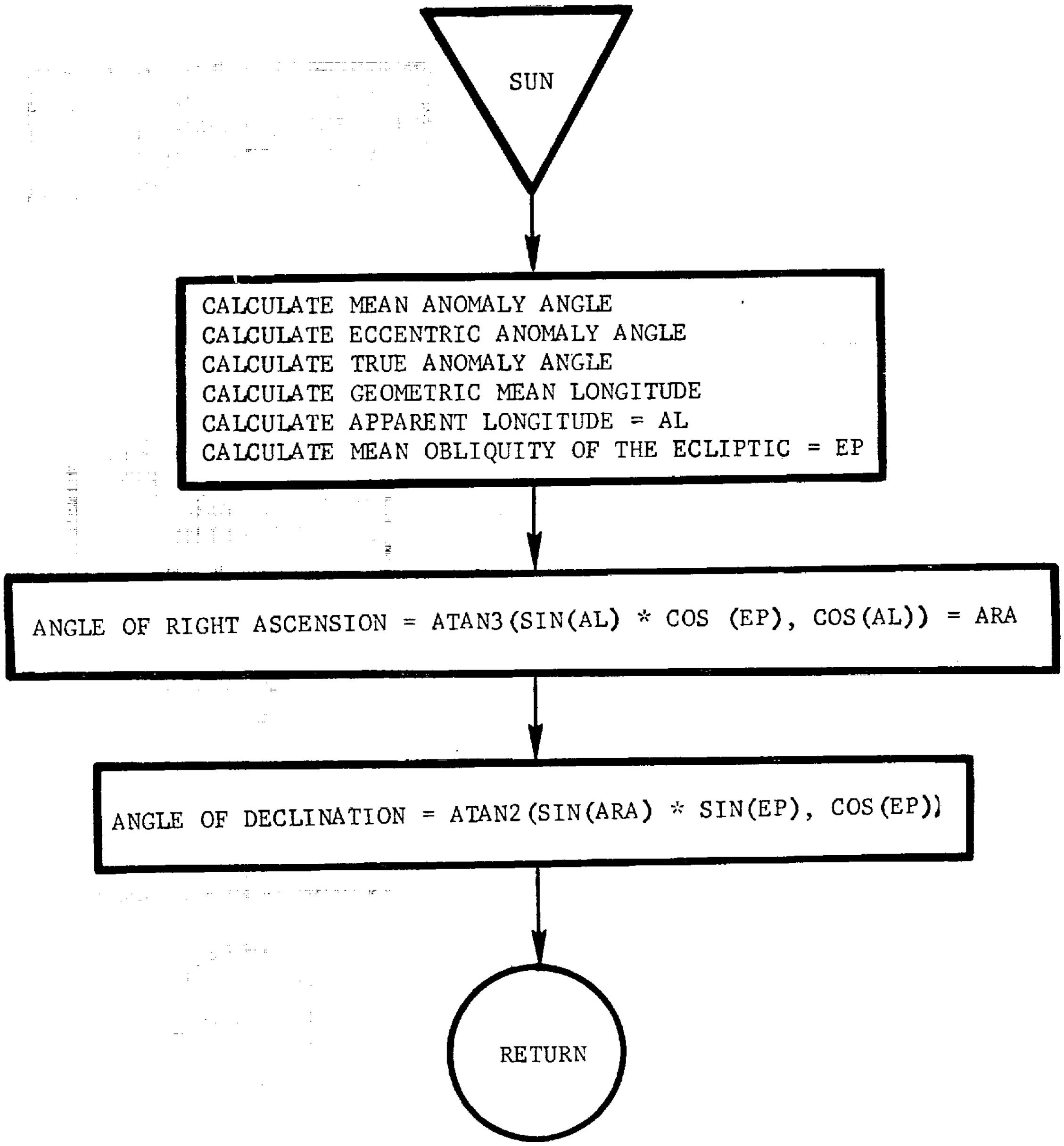
SETIV: This routine sets the control parameters to the desired values, based on the calculated control corrections.


SETIV
ARE THERE ANY CONTROL PARAMETERS?
NO.
YES
SET THE CONTROL PARAMETERS TO THE DESIRED VALUES
RESET THE TABLE NAMES POSITIVE; TAB SETS THEM NEGATIVE AS A FLAG TO INDICATE THAT THE TABLE VALUES WERE EXCEEDED
RETURN

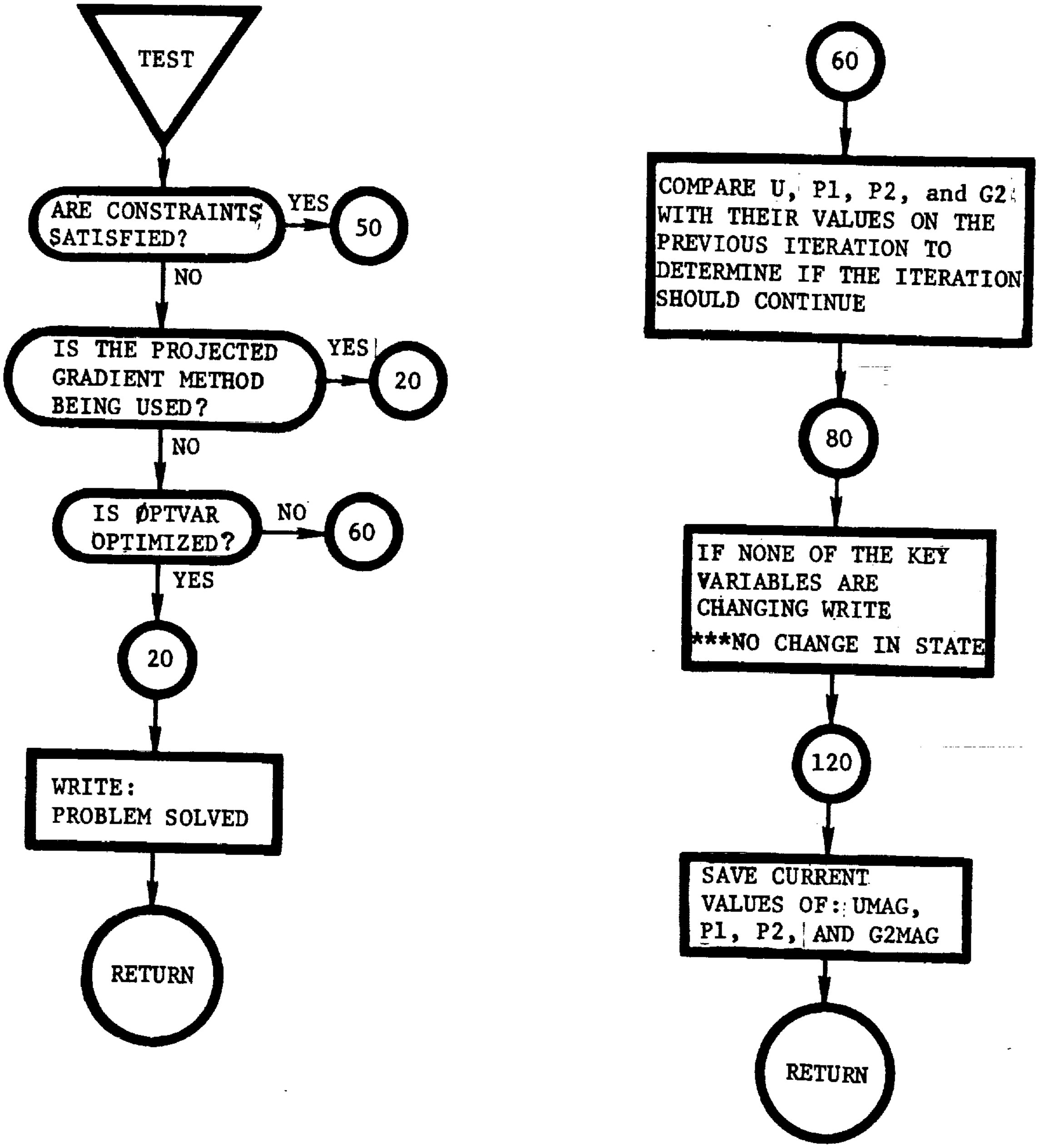
SHRINK (XBKT, IX, IL, IXBKT): This routine shrinks on array XBKT at position IX by IL words, where IXBKT is the total size of the array.



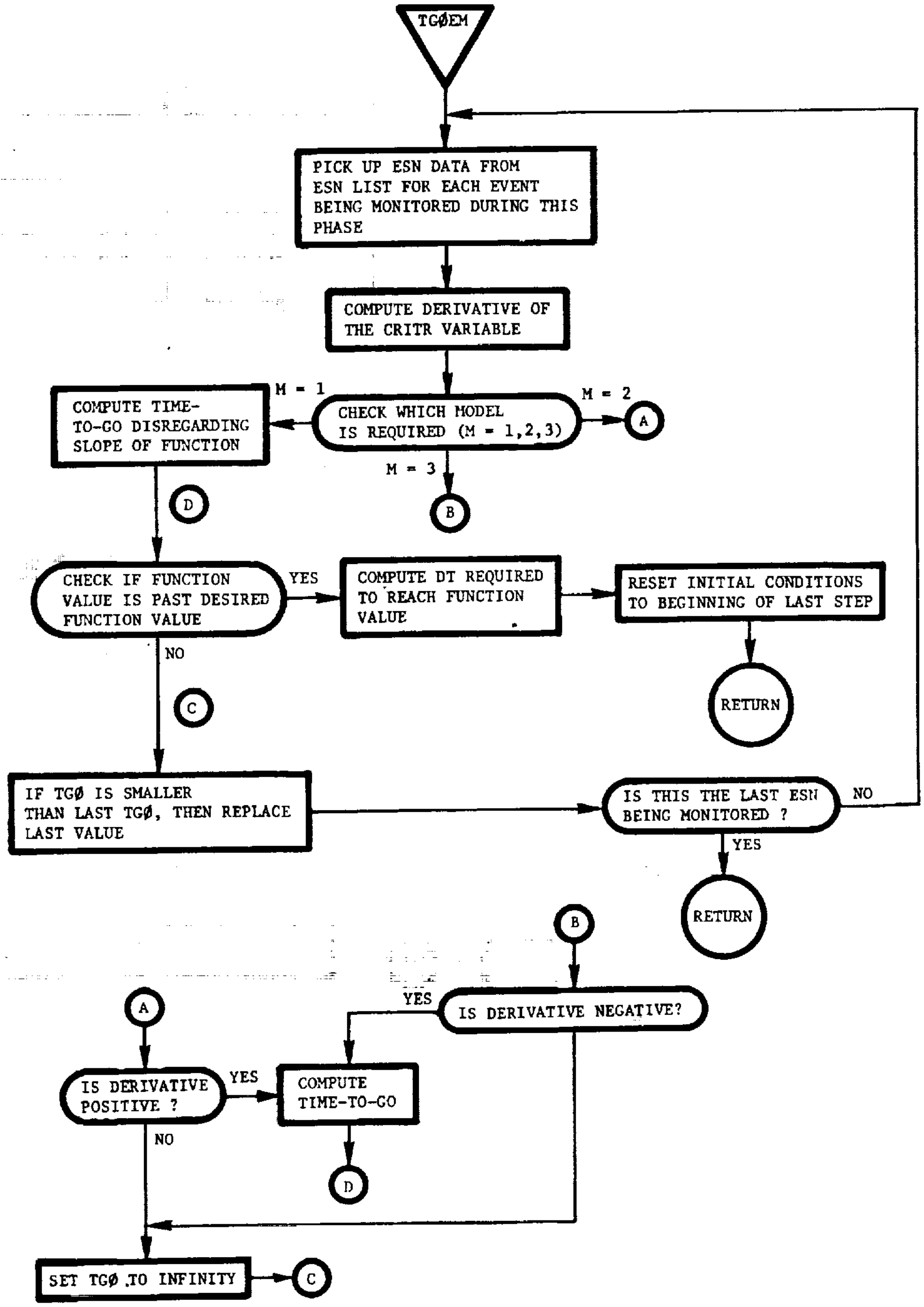
SUN: This routine calculates the right ascension and declination of the sun for any Julian date (from 1900 January).


SUN
CALCULATE MEAN ANOMALY ANGLE
CALCULATE ECCENTRIC ANOMALY ANGLE
CALCULATE TRUE ANOMALY ANGLE
CALCULATE GEOMETRIC MEAN LONGITUDE
CALCULATE APPARENT LONGITUDE = AL
CALCULATE MEAN OBLIQUITY OF THE ECLIPTIC = EP
ANGLE OF RIGHT ASCENSION = ATAN3(SIN(AL) * COS (EP), COS(AL)) = ARA
ANGLE OF DECLINATION = ATAN2(SIN(ARA) * SIN(EP), COS(EP))
RETURN

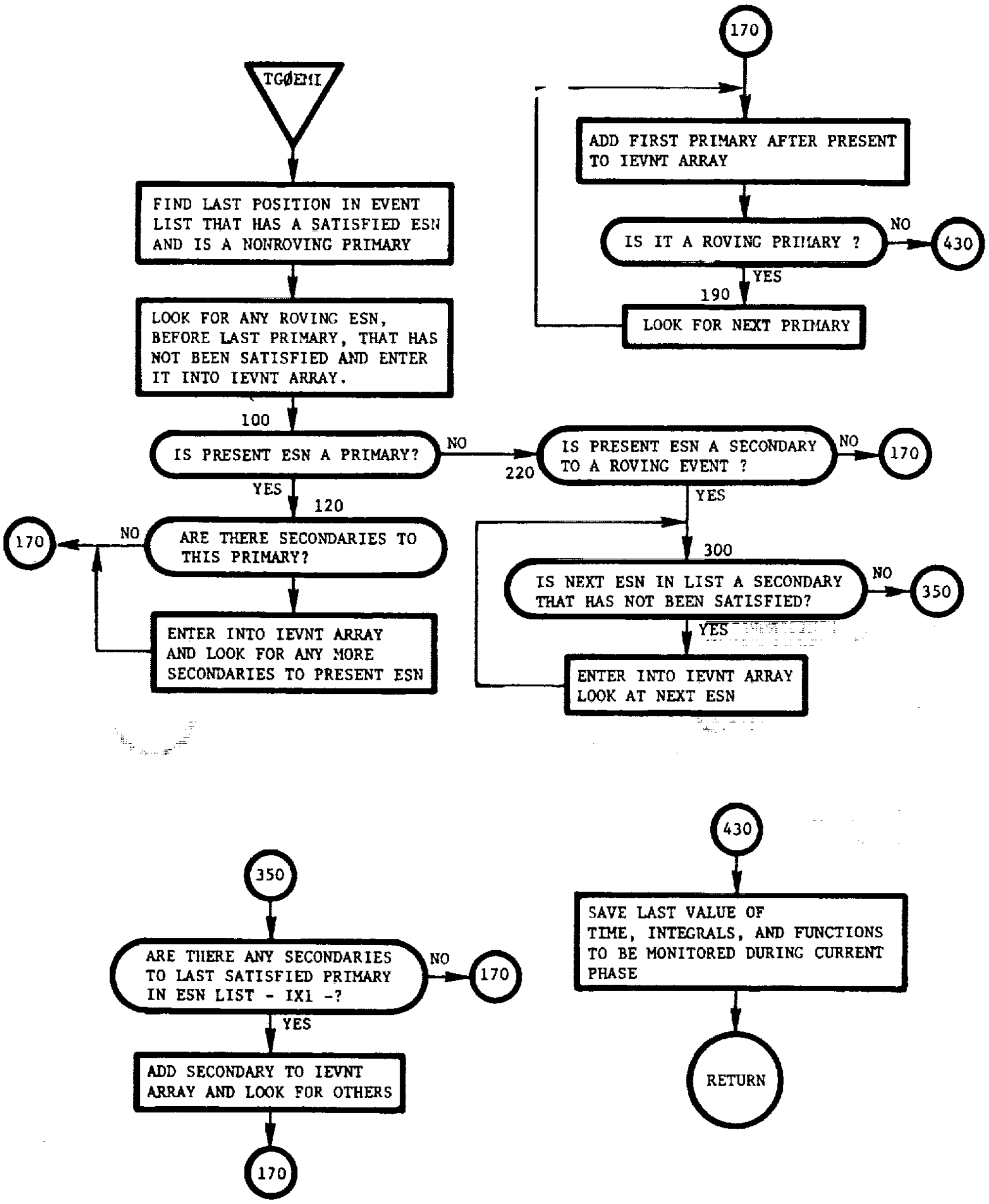
TEST: This routine tests for convergence.


TEST
ARE CONSTRAINTS SATISFIED?
YES
50
NO
IS THE PROJECTED GRADIENT METHOD BEING USED?
YES
20
NO
IS ØPTVAR OPTIMIZED?
NO
60
YES
20
WRITE: PROBLEM SOLVED
RETURN
60
COMPARE U, P1, P2, and G2 WITH THEIR VALUES ON THE PREVIOUS ITERATION TO DETERMINE IF THE ITERATION SHOULD CONTINUE
80
IF NONE OF THE KEY VARIABLES ARE CHANGING WRITE ***NO CHANGE IN STATE
120
SAVE CURRENT VALUES OF: UMAG, P1, P2, AND G2MAG
RETURN

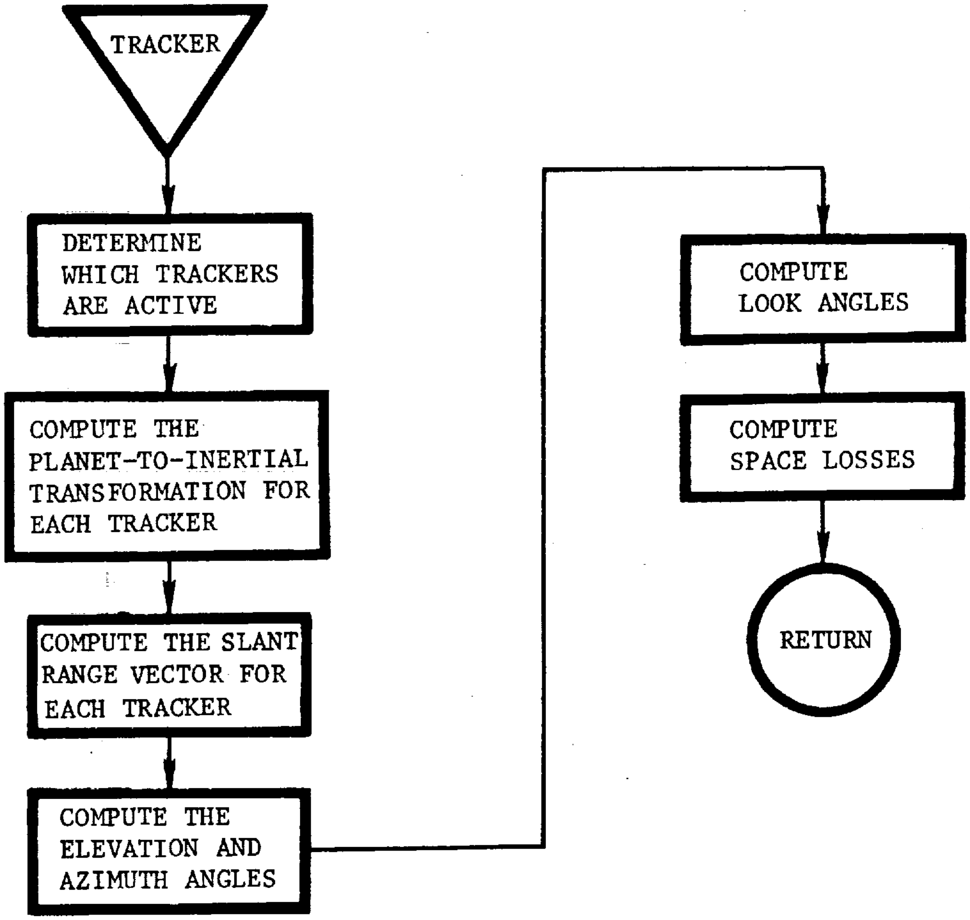
TGØEM: This routine domputes the time-to-go with the next event for each criterion being monitored during the current phase. The smallest value is then selected and returned to CYCXM.


TGØEM
PICK UP ESN DATA FROM ESN LIST FOR EACH EVENT BEING MONITORED DURING THIS PHASE
COMPUTE DERIVATIVE OF THE CRITR VARIABLE
CHECK WHICH MODEL IS REQUIRED (M = 1,2,3)
M = 1
M = 2
M = 3
A
B
COMPUTE TIME-TO-GO DISREGARDING SLOPE OF FUNCTION
D
CHECK IF FUNCTION VALUE IS PAST DESIRED FUNCTION VALUE
YES
NO
COMPUTE DT REQUIRED TO REACH FUNCTION VALUE
RESET INITIAL CONDITIONS TO BEGINNING OF LAST STEP
RETURN
C
IF TGØ IS SMALLER THAN LAST TGØ, THEN REPLACE LAST VALUE
IS THIS THE LAST ESN BEING MONITORED ?
NO
YES
RETURN
B
IS DERIVATIVE NEGATIVE?
YES
A
IS DERIVATIVE POSITIVE ?
YES
NO
COMPUTE TIME-TO-GO
D
SET TGØ TO INFINITY
C

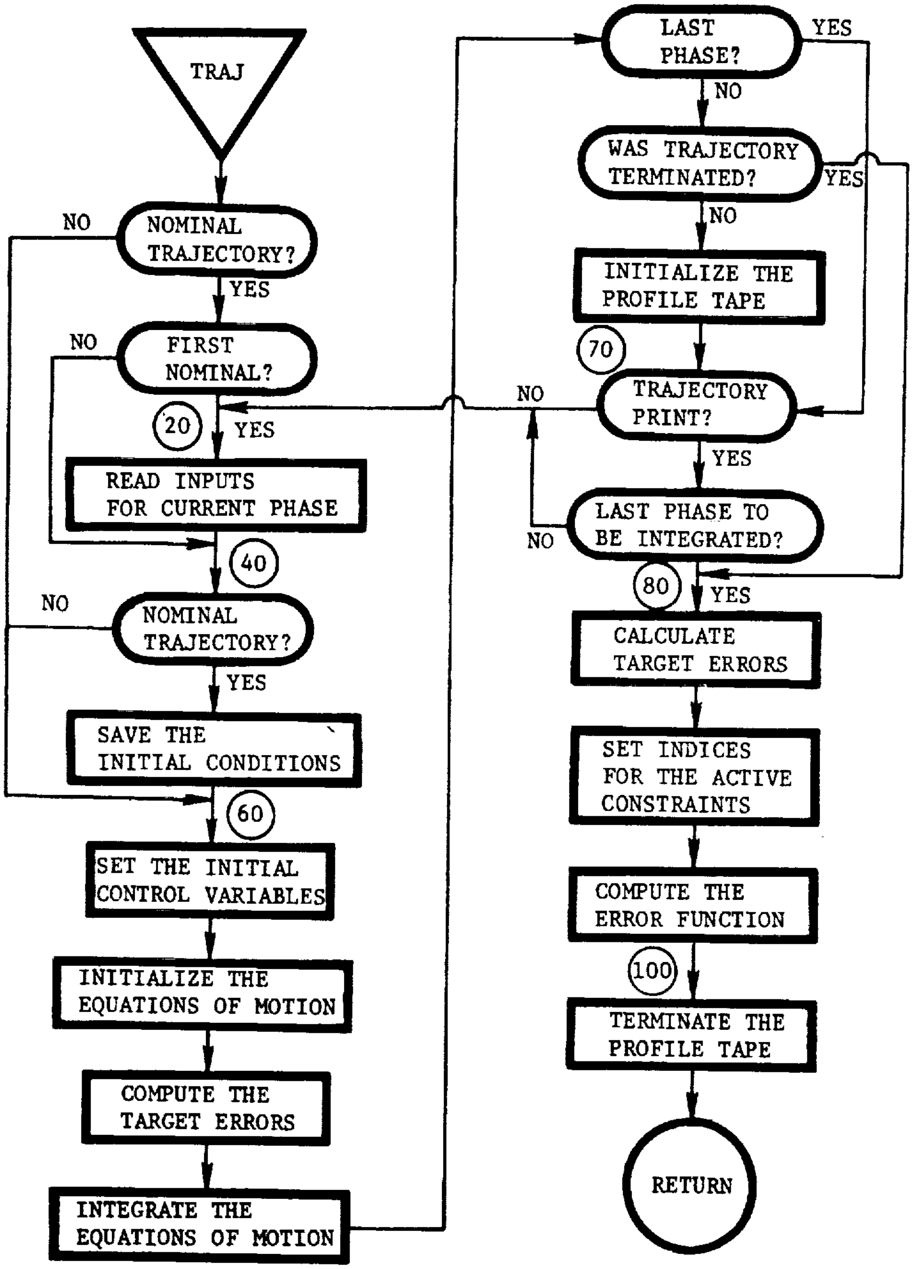
TGØEMI: This routine loads array IEVNT with the addresses of the events to be monitored during the current phase and specifies the order in which they are to be monitored.


TGØEMI
FIND LAST POSITION IN EVENT LIST THAT HAS A SATISFIED ESN AND IS A NONROVING PRIMARY
LOOK FOR ANY ROVING ESN, BEFORE LAST PRIMARY, THAT HAS NOT BEEN SATISFIED AND ENTER IT INTO IEVNT ARRAY.
100
IS PRESENT ESN A PRIMARY?
NO
YES
120
ARE THERE SECONDARIES TO THIS PRIMARY?
NO
170
ENTER INTO IEVNT ARRAY AND LOOK FOR ANY MORE SECONDARIES TO PRESENT ESN
220
IS PRESENT ESN A SECONDARY TO A ROVING EVENT ?
NO
170
YES
300
IS NEXT ESN IN LIST A SECONDARY THAT HAS NOT BEEN SATISFIED?
NO
350
YES
ENTER INTO IEVNT ARRAY LOOK AT NEXT ESN
170
ADD FIRST PRIMARY AFTER PRESENT TO IEVNT ARRAY
IS IT A ROVING PRIMARY ?
NO
430
YES
190
LOOK FOR NEXT PRIMARY
350
ARE THERE ANY SECONDARIES TO LAST SATISFIED PRIMARY IN ESN LIST - IX1 -?
NO
170
YES
ADD SECONDARY TO IEVNT ARRAY AND LOOK FOR OTHERS
170
430
SAVE LAST VALUE OF TIME, INTEGRALS, AND FUNCTIONS TO BE MONITORED DURING CURRENT PHASE
RETURN

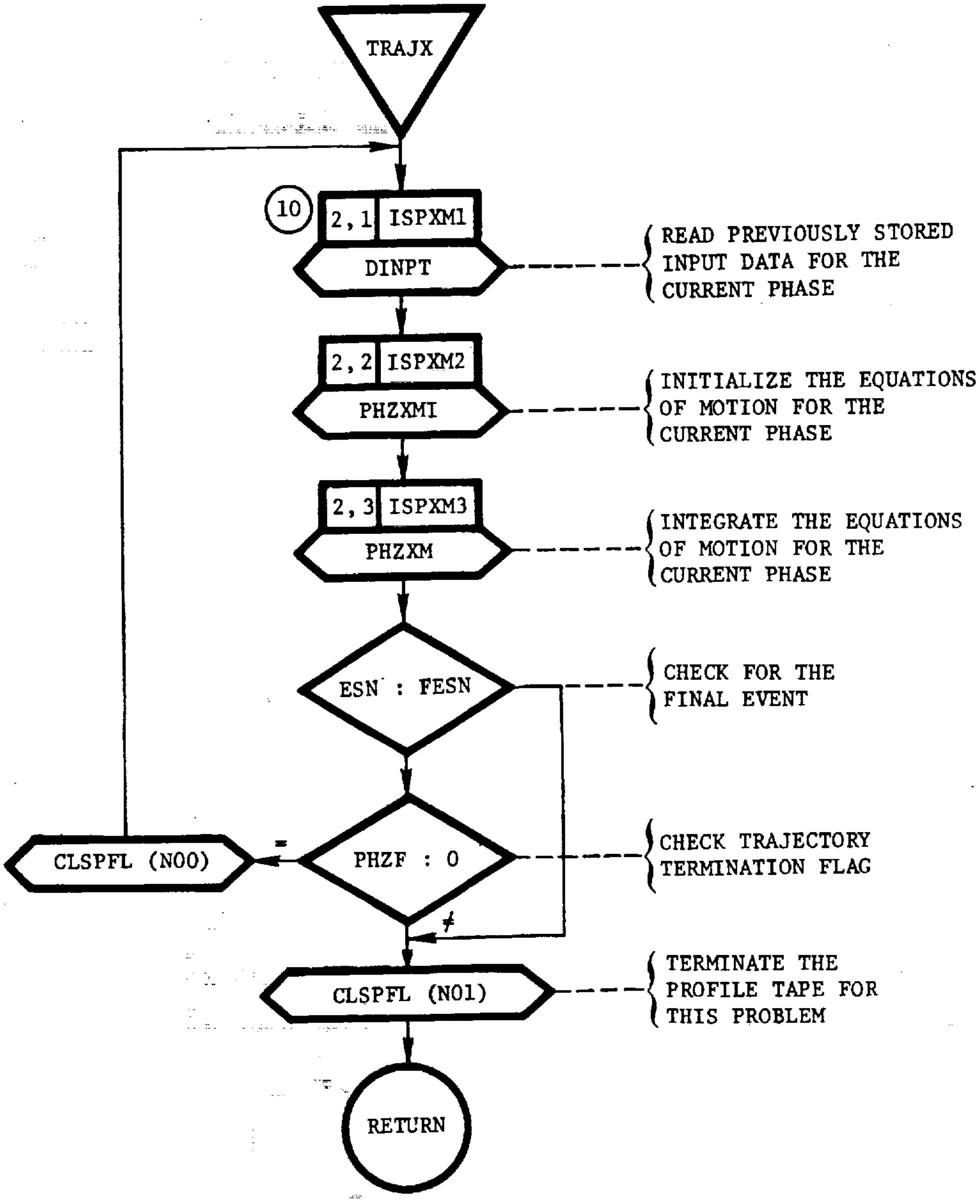
TRACKER: This routine computes the slant range, angle of elevation, azimuth angle, look angles, and space losses between specified tracker stations and the vehicle.


TRACKER
DETERMINE WHICH TRACKERS ARE ACTIVE
COMPUTE THE PLANET-TO-INERTIAL TRANSFORMATION FOR EACH TRACKER
COMPUTE THE SLANT RANGE VECTOR FOR EACH TRACKER
COMPUTE THE ELEVATION AND AZIMUTH ANGLES
COMPUTE LOOK ANGLES
COMPUTE SPACE LOSSES
RETURN

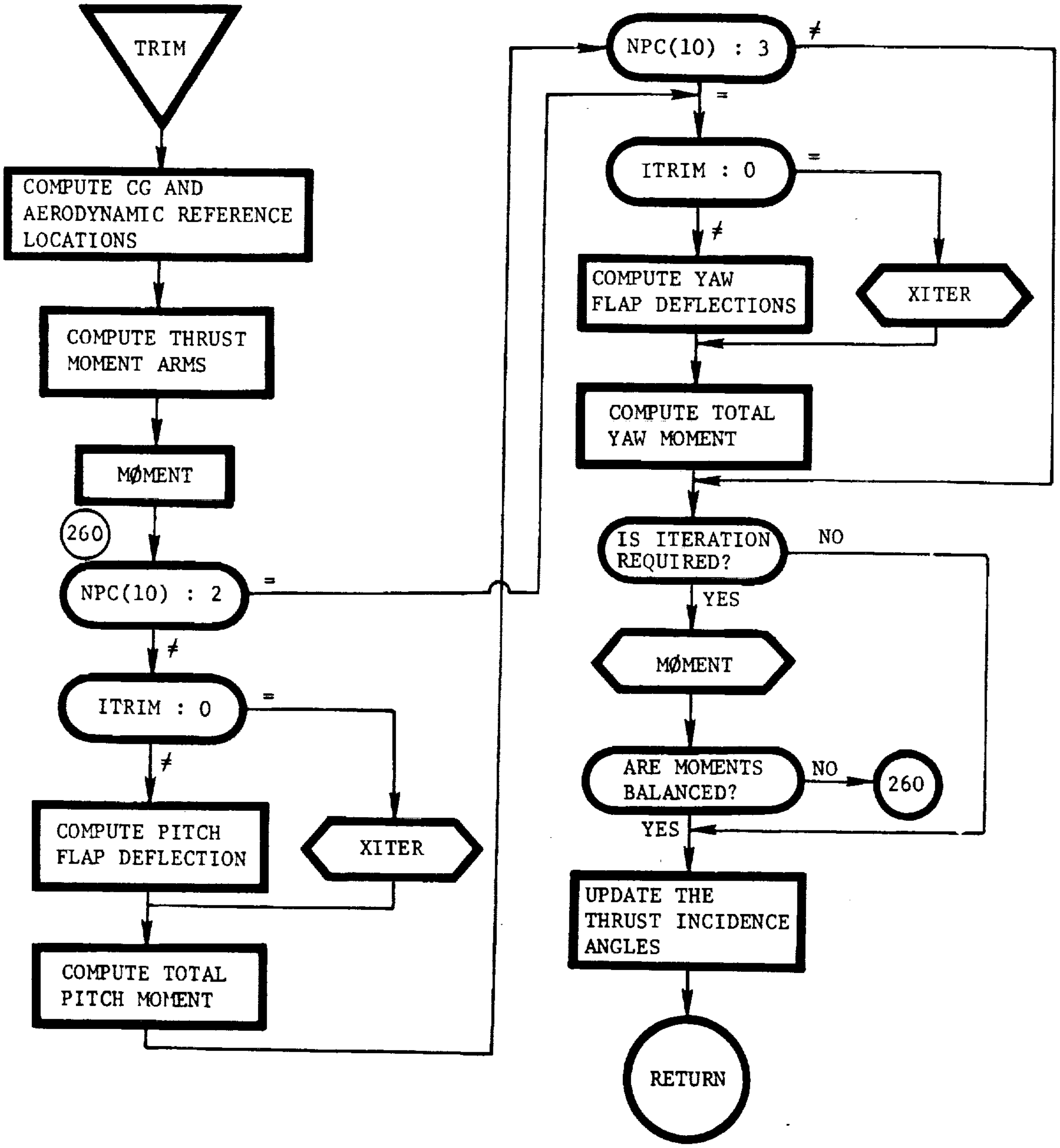
TRAJ: This routine propagates a trajectory from the beginning of the specified phase to the final cutoff condition.


TRAJ
NOMINAL TRAJECTORY?
NO
YES
FIRST NOMINAL?
NO
20
YES
READ INPUTS FOR CURRENT PHASE
40
NOMINAL TRAJECTORY?
NO
YES
SAVE THE INITIAL CONDITIONS
60
SET THE INITIAL CONTROL VARIABLES
INITIALIZE THE EQUATIONS OF MOTION
COMPUTE THE TARGET ERRORS
INTEGRATE THE EQUATIONS OF MOTION
LAST PHASE?
YES
NO
WAS TRAJECTORY TERMINATED?
YES
NO
INITIALIZE THE PROFILE TAPE
70
TRAJECTORY PRINT?
NO
YES
LAST PHASE TO BE INTEGRATED?
NO
80
YES
CALCULATE TARGET ERRORS
SET INDICES FOR THE ACTIVE CONSTRAINTS
COMPUTE THE ERROR FUNCTION
100
TERMINATE THE PROFILE TAPE
RETURN

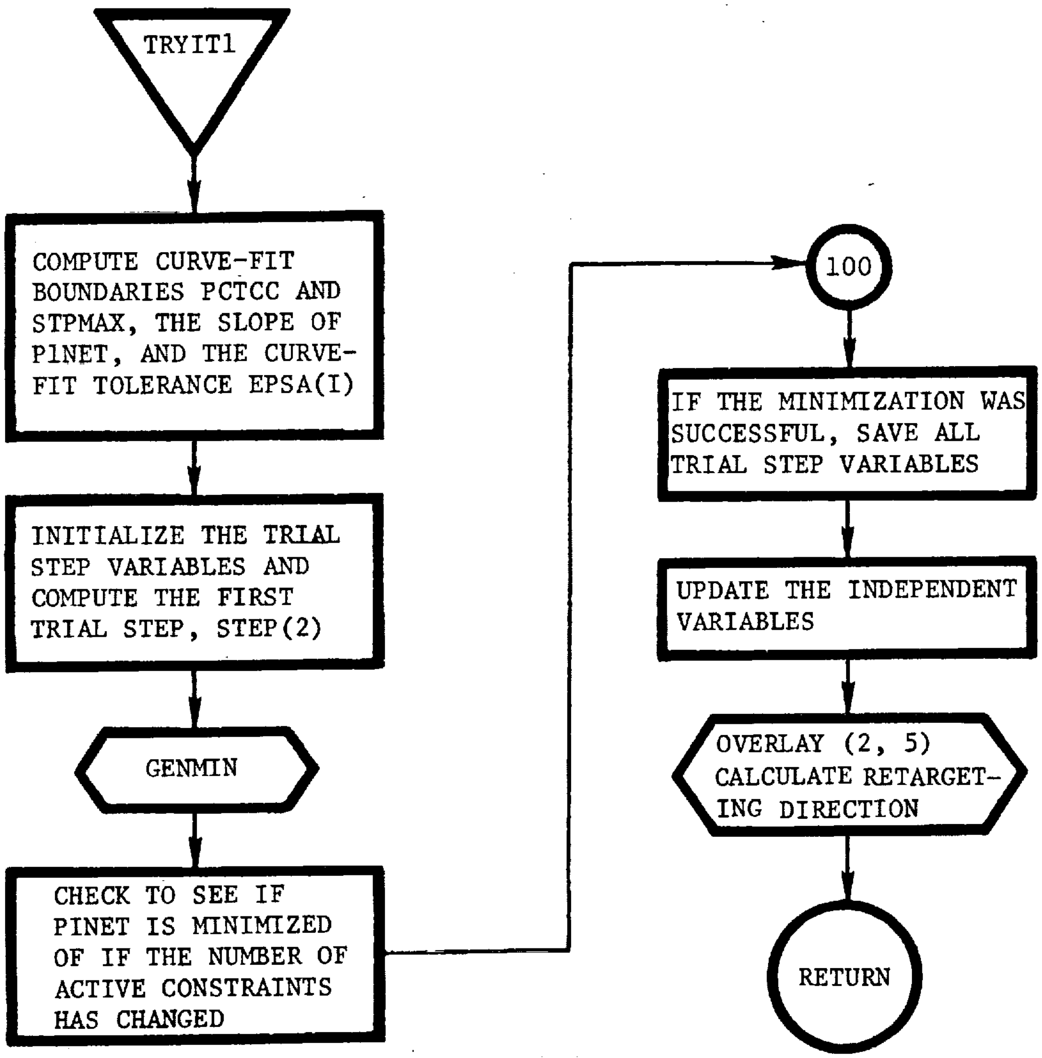
TRAJX: This routine is called when the single trajectory option is requested (SRCHM = 0). This routine calls overlays (2, 1), (2, 2), and (2, 3) in sequence to implement the trajectory simulation desired.


TRAJX
10
2,1 ISPXM1
DINPT
READ PREVIOUSLY STORED INPUT DATA FOR THE CURRENT PHASE
2,2 ISPXM2
PHZXMI
INITIALIZE THE EQUATIONS OF MOTION FOR THE CURRENT PHASE
2,3 ISPXM3
PHZXM
INTEGRATE THE EQUATIONS OF MOTION FOR THE CURRENT PHASE
ESN : FESN
CHECK FOR THE FINAL EVENT
CLSPFL (N00)
=
PHZF : 0
≠
CHECK TRAJECTORY TERMINATION FLAG
CLSPFL (N01)
TERMINATE THE PROFILE TAPE FOR THIS PROBLEM
RETURN

TRIM: This routine calculates the engine deflection angle required to balance the aerodynamic moments when using rocket engines, or the flap deflection angle required to balance the aerodynamic moments when using jet engines.

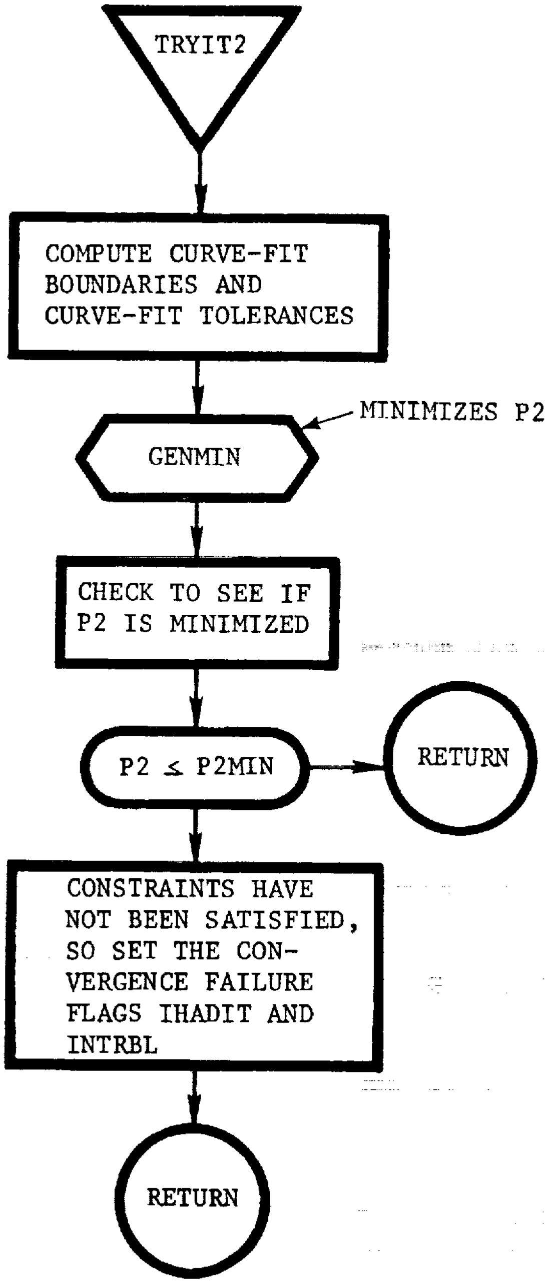

TRIM
COMPUTE CG AND AERODYNAMIC REFERENCE LOCATIONS
COMPUTE THRUST MOMENT ARMS
MØMENT
260
NPC(10) : 2
=
≠
ITRIM : 0
=
≠
COMPUTE PITCH FLAP DEFLECTION
XITER
COMPUTE TOTAL PITCH MOMENT
NPC(10) : 3
≠
=
ITRIM : 0
=
≠
COMPUTE YAW FLAP DEFLECTIONS
XITER
COMPUTE TOTAL YAW MOMENT
IS ITERATION REQUIRED?
NO
YES
MØMENT
ARE MOMENTS BALANCED?
NO
260
YES
UPDATE THE THRUST INCIDENCE ANGLES
RETURN

TRYIT1: This routine minimizes the estimated net cost function as a function of the step-size parameter. The principal function of the routine is to setup of the data required by GEMNIN, where GENMIN is the routine that actually minimizes the function.

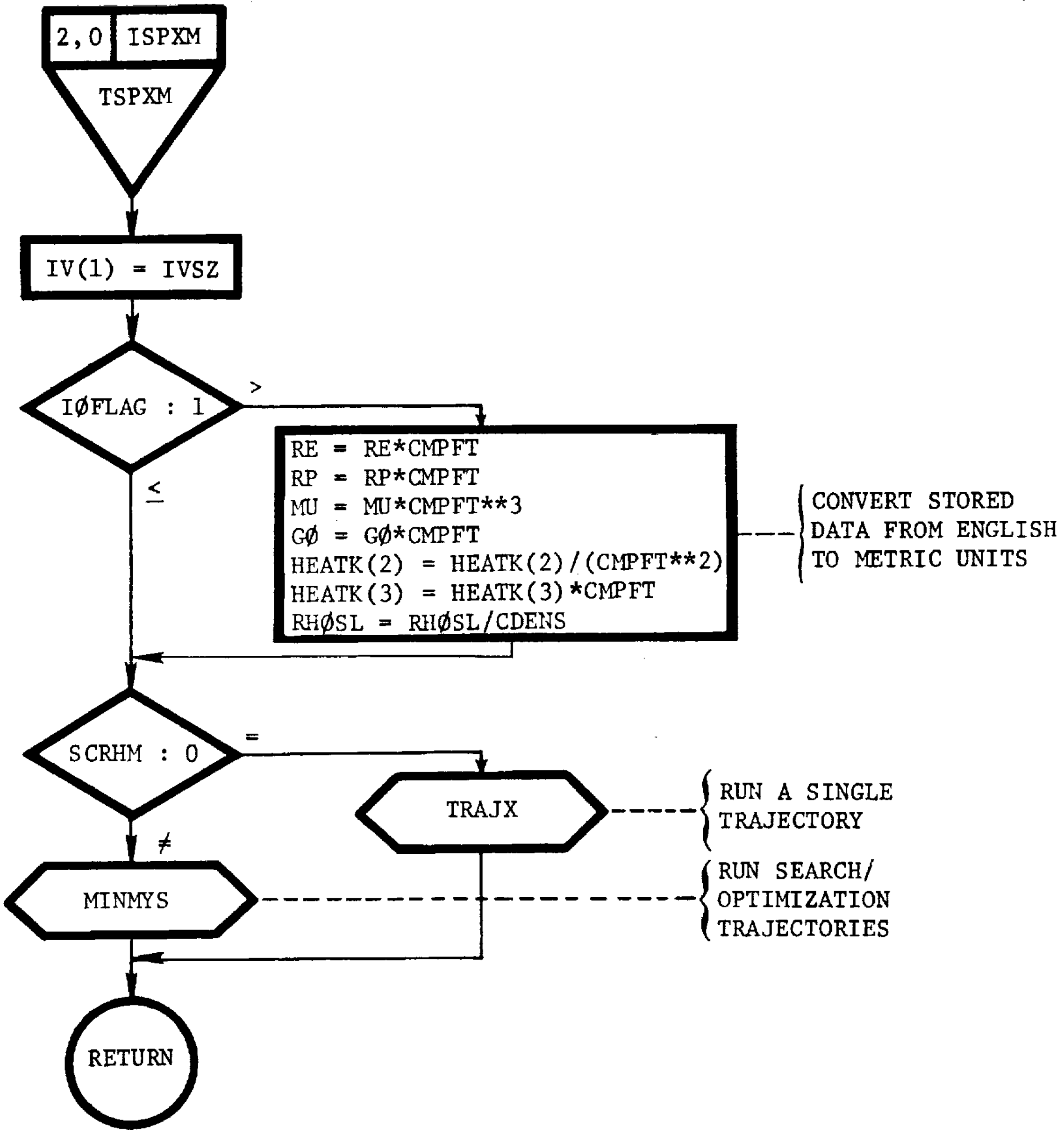

TRYIT1
COMPUTE CURVE-FIT BOUNDARIES PCTCC AND STPMAX, THE SLOPE OF P1NET, AND THE CURVE-FIT TOLERANCE EPSA(I)
INITIALIZE THE TRIAL STEP VARIABLES AND COMPUTE THE FIRST TRIAL STEP, STEP(2)
GENMIN
CHECK TO SEE IF PINET IS MINIMIZED OF IF THE NUMBER OF ACTIVE CONSTRAINTS HAS CHANGED
100
IF THE MINIMIZATION WAS SUCCESSFUL, SAVE ALL TRIAL STEP VARIABLES
UPDATE THE INDEPENDENT VARIABLES
OVERLAY (2, 5) CALCULATE RETARGETING DIRECTION
RETURN

C - 13

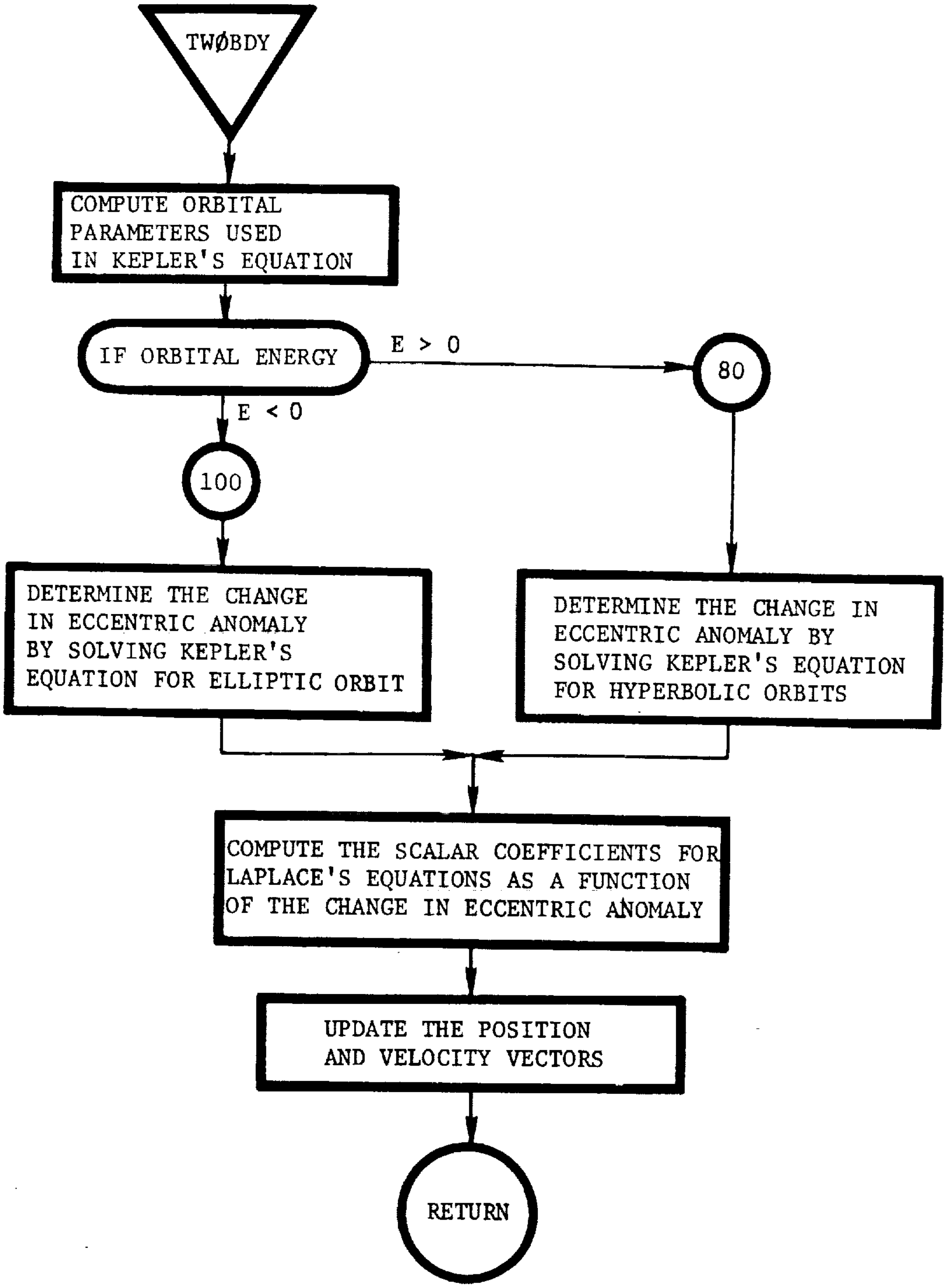
TRYIT2: This function minimizes the constraint error function P2.


TRYIT2
COMPUTE CURVE-FIT BOUNDARIES AND CURVE-FIT TOLERANCES
MINIMIZES P2
GENMIN
CHECK TO SEE IF P2 IS MINIMIZED
P2 ≤ P2MIN
RETURN
CONSTRAINTS HAVE NOT BEEN SATISFIED, SO SET THE CONVERGENCE FAILURE FLAGS IHADIT AND INTRBL
RETURN

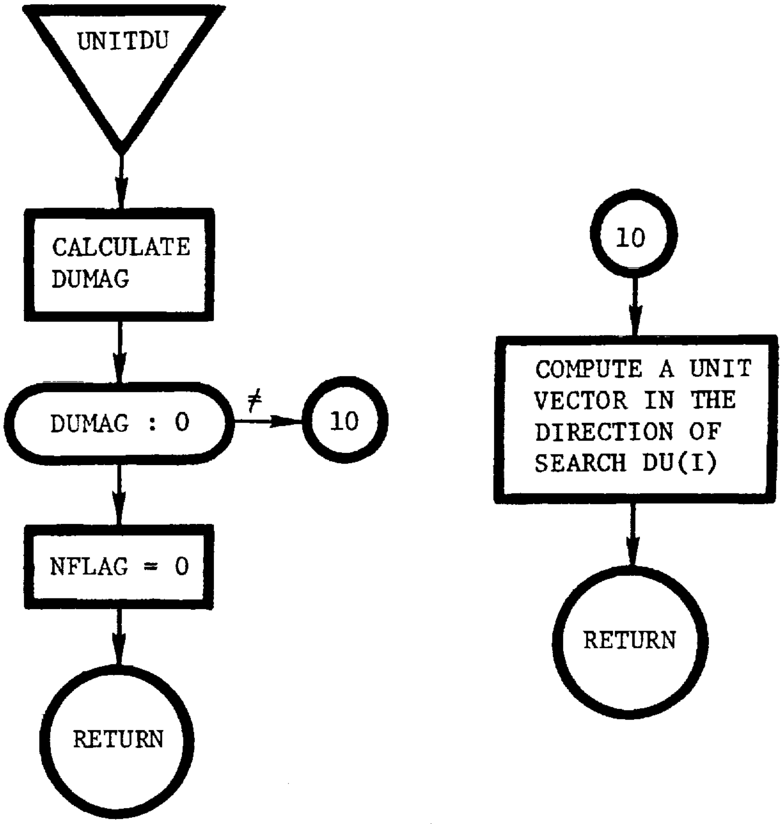
TSPXM: This routine is the main program of overlay (2, 0) and controls the overall operation of the trajectory simulation routines.



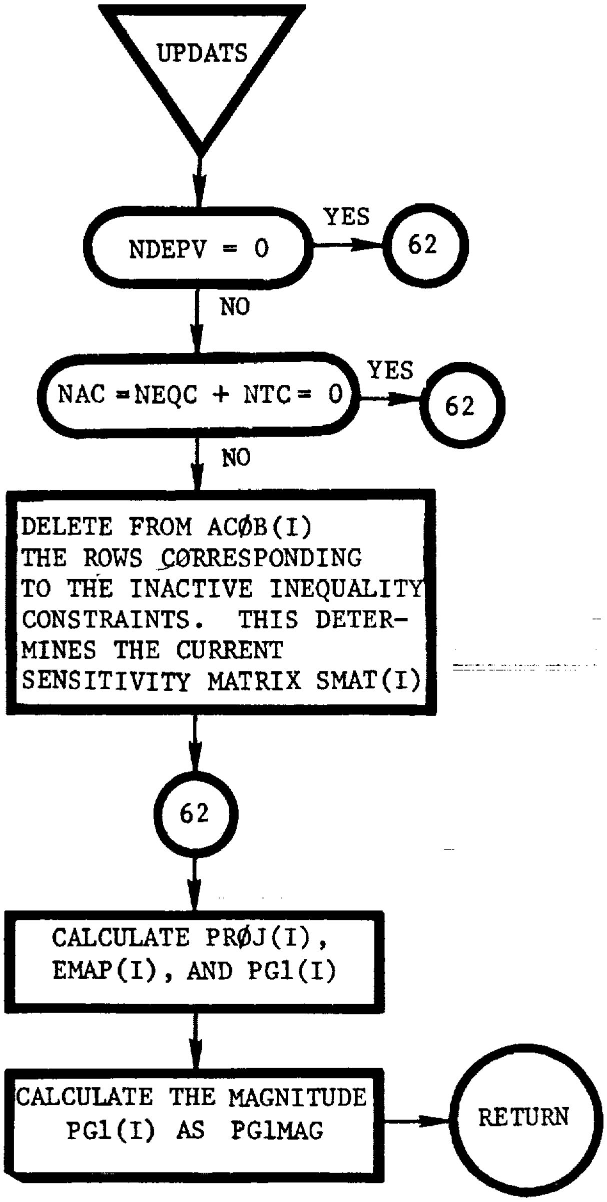
TWØBDY: This routine solves the orbital equations of motion about a spherical planet using the method of Laplace.


TWØBDY
COMPUTE ORBITAL PARAMETERS USED IN KEPLER'S EQUATION
IF ORBITAL ENERGY
E > 0
80
E < 0
100
DETERMINE THE CHANGE IN ECCENTRIC ANOMALY BY SOLVING KEPLER'S EQUATION FOR ELLIPTIC ORBIT
DETERMINE THE CHANGE IN ECCENTRIC ANOMALY BY SOLVING KEPLER'S EQUATION FOR HYPERBOLIC ORBITS
COMPUTE THE SCALAR COEFFICIENTS FOR LAPLACE'S EQUATIONS AS A FUNCTION OF THE CHANGE IN ECCENTRIC ANOMALY
UPDATE THE POSITION AND VELOCITY VECTORS
RETURN

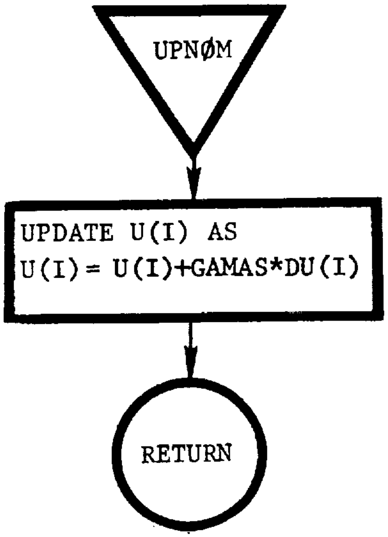
UNITDU: This routine unitizes the control correction vector and computes the magnitude of the control correction vector.



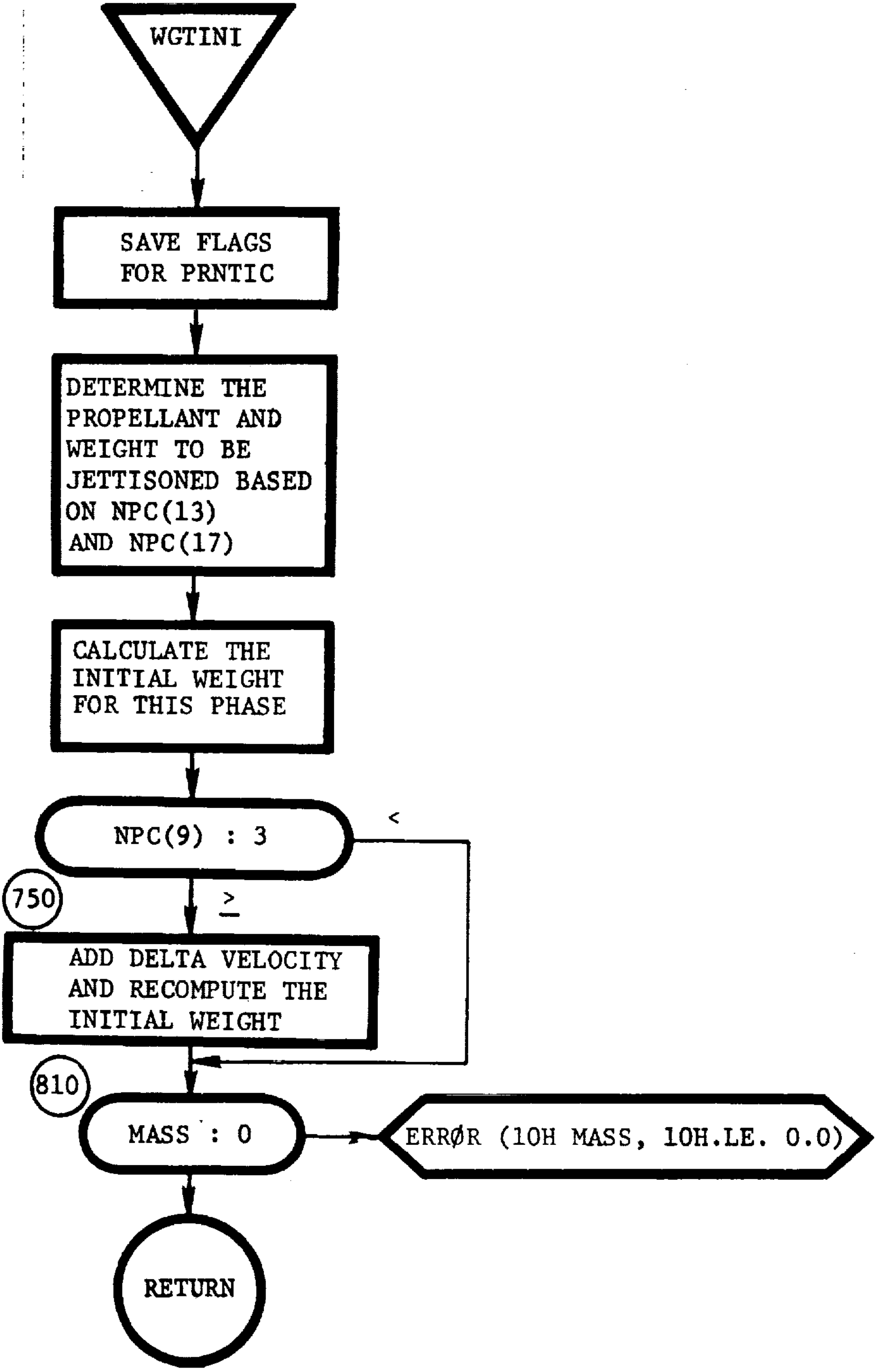
UPDATS: This routine updates the sensitivity matrix by deleting those rows that can be dropped. The routine also computes the projection matrix PRØJ(I) and the error map matrix in subroutine PGM to determine the direction of search.


UPDATS
NDEPV = 0
YES
62
NO
NAC =NEQC + NTC= 0
YES
62
NO
DELETE FROM ACØB(I) THE ROWS CORRESPONDING TO THE INACTIVE INEQUALITY CONSTRAINTS. THIS DETERMINES THE CURRENT SENSITIVITY MATRIX SMAT(I)
62
CALCULATE PRØJ(I), EMAP(I), AND PG1(I)
CALCULATE THE MAGNITUDE PG1(I) AS PG1MAG
RETURN

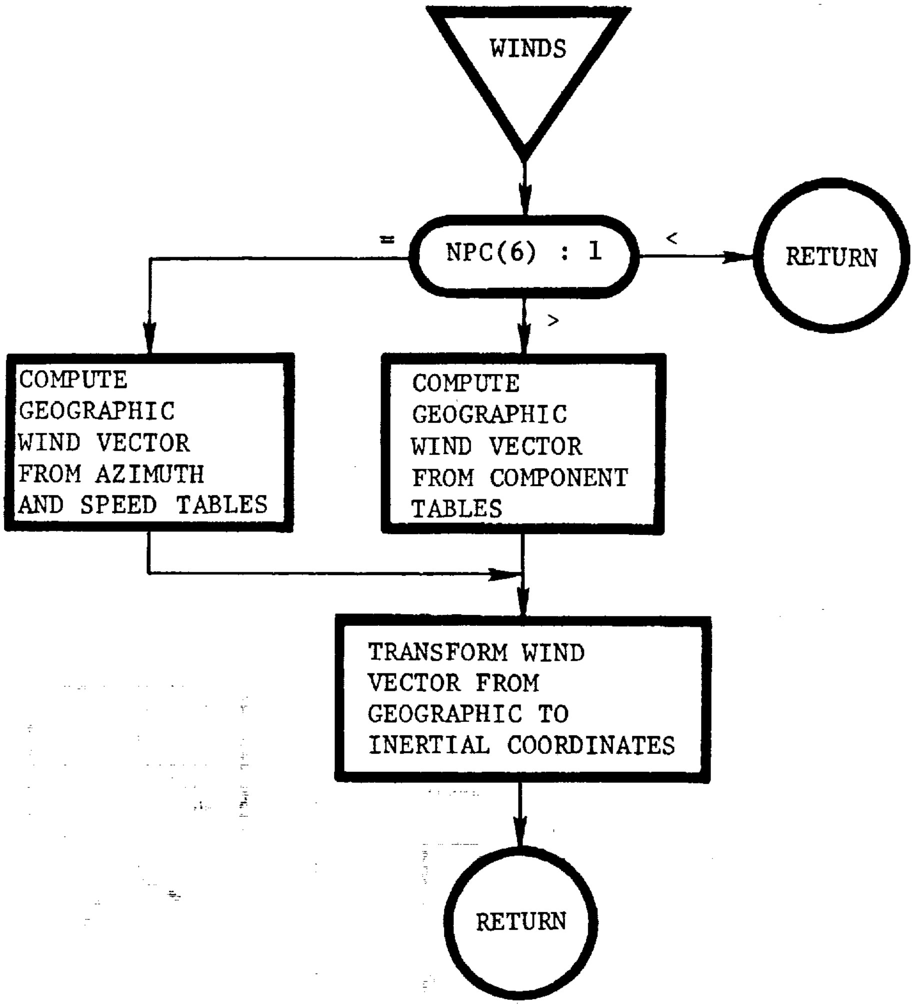
UPNØM: This routine updates the nominal values of the independent variables according the direction of search DU(I) and the step size GAMAS.



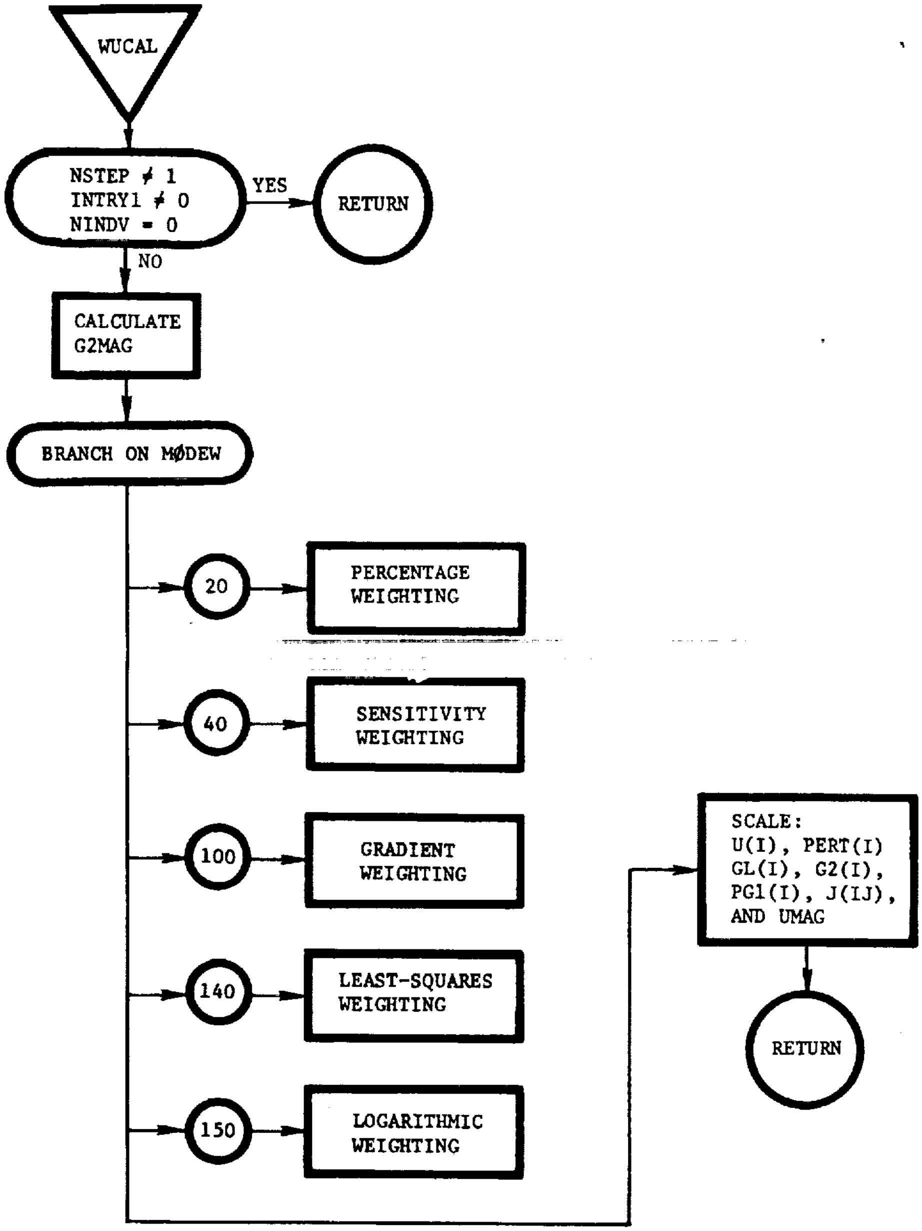
WGTINI: This routine initializes the vehicle's weight for the current phase and adds the specified delta velocity if requested by user input.


WGTINI
SAVE FLAGS FOR PRNTIC
DETERMINE THE PROPELLANT AND WEIGHT TO BE JETTISONED BASED ON NPC(13) AND NPC(17)
CALCULATE THE INITIAL WEIGHT FOR THIS PHASE
NPC(9) : 3
<
750
≥
ADD DELTA VELOCITY AND RECOMPUTE THE INITIAL WEIGHT
810
MASS : 0
ERRØR (10H MASS, 10H.LE. 0.0)
RETURN

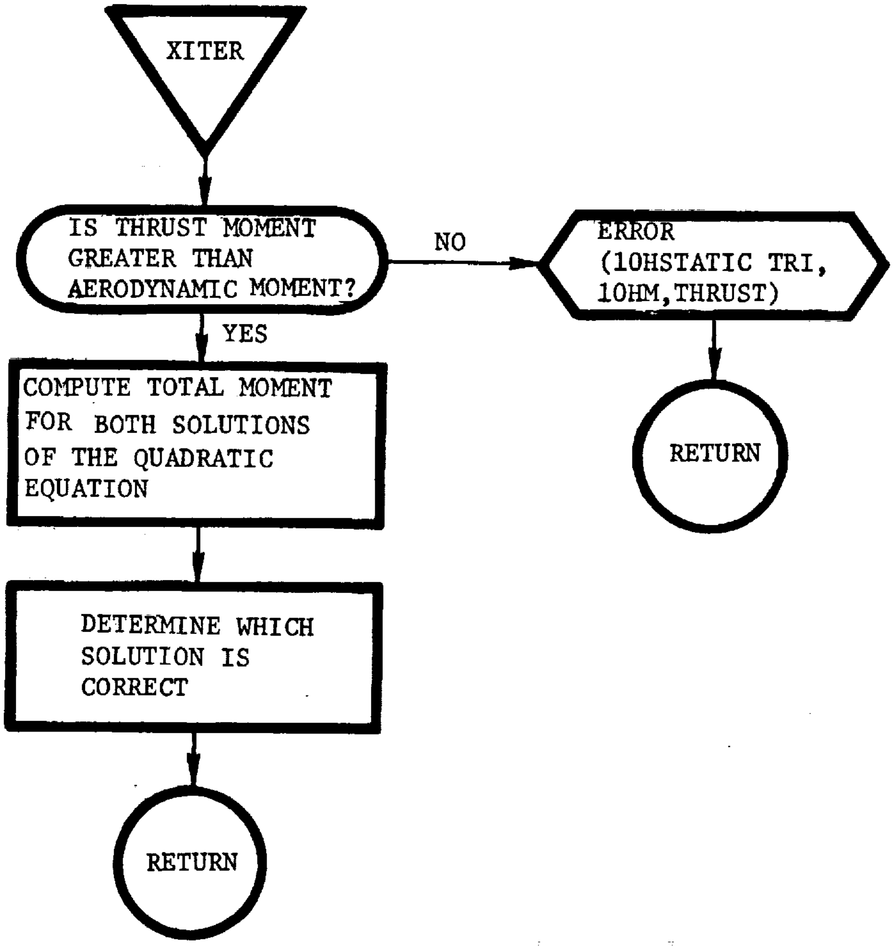
WINDS: This routine calculates the components of the wind velocity vector in the Earth-centered inertial coordinate system.


WINDS
NPC(6) : 1
=
<
>
RETURN
COMPUTE GEOGRAPHIC WIND VECTOR FROM AZIMUTH AND SPEED TABLES
COMPUTE GEOGRAPHIC WIND VECTOR FROM COMPONENT TABLES
TRANSFORM WIND VECTOR FROM GEOGRAPHIC TO INERTIAL COORDINATES
RETURN

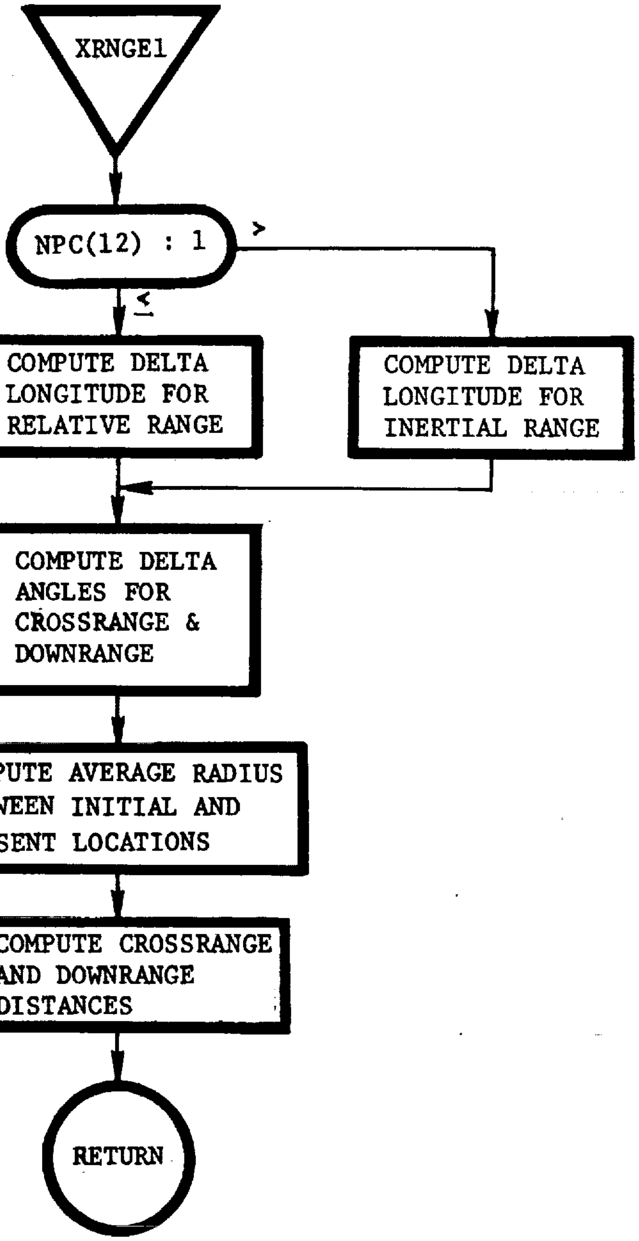
WUCAL: This routine calculates the weighting matrix for the independent variables.



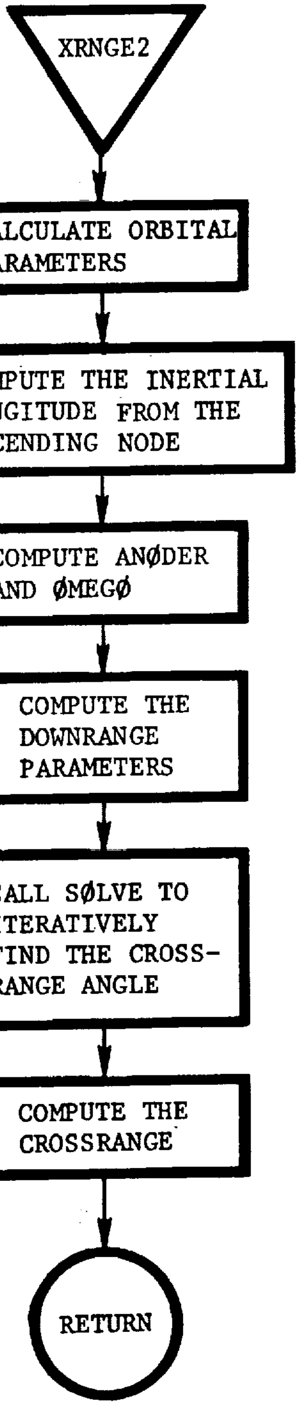
XITER (A, B, C, SIP, CIP, TTMXB): This routine solves the quadratic equation to determine the sine and cosine of the thrust incidence angle (SIP and CIP) and the thrust moment required to balance the moments (TTMXB), given the aerodynamic and thrust moment to be balanced (A) and the components of thrust (B and C).


XITER
IS THRUST MOMENT GREATER THAN AERODYNAMIC MOMENT?
NO
ERROR (10HSTATIC TRI, 10HM,THRUST)
RETURN
YES
COMPUTE TOTAL MOMENT FOR BOTH SOLUTIONS OF THE QUADRATIC EQUATION
DETERMINE WHICH SOLUTION IS CORRECT
RETURN

XRNGE1: This routine calculates the downrange and crossrange distances, based on relative great circles.


XRNGE1
NPC(12) : 1
>
≤
COMPUTE DELTA LONGITUDE FOR RELATIVE RANGE
COMPUTE DELTA LONGITUDE FOR INERTIAL RANGE
COMPUTE DELTA ANGLES FOR CROSSRANGE & DOWNRANGE
COMPUTE AVERAGE RADIUS BETWEEN INITIAL AND PRESENT LOCATIONS
COMPUTE CROSSRANGE AND DOWNRANGE DISTANCES
RETURN

XRNGE2: This routine calculates the crossrange distance based on a ground track of a reference circular orbit.


XRNGE2
CALCULATE ORBITAL PARAMETERS
COMPUTE THE INERTIAL LONGITUDE FROM THE ASCENDING NODE
COMPUTE ANØDER AND ØMEGØ
COMPUTE THE DOWNRANGE PARAMETERS
CALL SØLVE TO ITERATIVELY FIND THE CROSS-RANGE ANGLE
COMPUTE THE CROSSRANGE
RETURN

IV. SERVICE ROUTINES

The following service routines are used to perform arithmetic tasks that are used frequently throughout the program.

ABT (A,B,C,L,M,N): This routine computes matrix C given matrices A and B, the dimensions of A (L by M), and the dimensions of B (M by N) as matrix A times the transpose of matrix B.

ADDREL: This routine prints a namelist like statement for a single variable name. This output is for use in ODINEX executed programs.

ANGLE2 (X,Y,ALPHA): This routine computes the angle ALPHA, in degrees, between vectors X and Y. ALPHA is measured counter-clockwise from X to Y and can range from 0 to 360 deg.

ATANH (A): This function computes the angle, from 0 to 2 pi radians, whose arc-tangent is A over B.

BTW (B,W,D,L,M.N): This routine computes matrix D as the transpose of matrix B times matrix W, where L, M, and N are the dimensions of matrices B and W.

BUCKET (B,W,D,L,M,N): This routine rearranges an array X and the corresponding elements of Y in ascending order. N is the number of elements in each array. NP is a pointer indicating the first element of Y that is less than the next element. NP is zero if it does not exist. The ordered arrays are returned as XX and YY.

CØNICT: This routine calculates the Keplerian conic for the target vehicle. Same flow chart as CØNIC.

CØSH (A): This function returns as the hyperbolic cosine of A.

CUBMIN (A,XMIN,YMIN): This routine evaluates a cubic polynominal coefficients (A), and returns the minimum values of the function (YMIN) and the corresponding value of the argument (XMIN).

CYCYM: This routine performs the cycling functions for the variable step/order predictor-corrector integrator.

DIFTAB (TABLE,DELT): This function computes the derivative of a monovariant table (TABLE) using finite differences over delta X (DELT).

DIGDIF (M,N,NDIF): This routine finds the number of different digits (NDIF) between M and N, where M and N are base-10 numbers.

DISPLY: This routine displays certain parameters from the search/optimization algorithm on the cathode-ray tube of the 6400 computer console.

DPØLY3 (A,N): This function evaluates the derivative of polynomial: A(1) + A(2)*X + A(3) x X**2 +A(4)*X**3 where N is the address of X in the dictionary.

DTAB (XS,YS,N,D): This function performs the table lookup tasks for double-valued (bivariant) tables. XS is the value of the X argument, YS is the value of the Y argument, N is the pointer array, and D is the table array.

ERRØR (I,J): This routine writes an error message and determines if it is fatal or nonfatal. I and J are Hollerith words. If the first letter in I or J is blank, the error is fatal; otherwise it is nonfatal.

EXPN (IBKT,IX,IL,IXBKT): This routine moves the contents of IBKT from IXBKT down IL locations, and shifts the contents of IBKT from IX down IL locations. IXBKT is the last occupied cell in IBKT.

FITER (FPDATA,IDATA,XN,FN,N,FCT): This routine is called by subroutine SØLVE to find the zero of the function by either the Regula Falsi or secant methods. The arguments FPDATA and IDATA are the same as those defined for subroutine SØLVE. XN is the current value of X and FN is the corresponding function value. N is the current iteration number and FCT is the external function defining the function.

FØPMIN (X,Y,XMIN,YMIN,IERR): This routine calculates the minimum of a polynomial based on four ordered pairs (X,Y). The abscissa value that minimizes this cubic polynomial is returned as XMIN and the corresponding ordinate value is returned as YMIN.

FØRMN: This routine computes the nutation in right ascension.

GENTAB (TABLE): This function is the executive table look-up routine. It determines whether the table is constant-valued, monovariant, bivariant, or trivariant, and calls functions TAB, DTAB, and TRITAB as required.

GSAI: This subroutine sets a flag (ISSEPS=0) to indicate no steering sensitivity matrix has been calculated. It is called once a phase.

INVM (A,N): This routine inverts an N by N matrix A and returns A as the inverse.

IRMTRX (R,V,ETR): This subroutine computes the transformation matrix (ETR) from ECI coordinates to target-centered relative coordinates where R = ECI position and V = ECI velocity.

LEVEL: This routine calculates the angle of attack and throttling parameter required to maintain level flight.

MADD (A,SA,B,SB,C,K,L): This routine adds matris A and matrix B to form matrix C, where SA and SB are scalars that are applied to A and B before the addition is performed. The dimensions of A and B are given as K by L.

MATPY (A,B,C,L,M,N): This routine multiplies matrix A by matrix B to produce matrix C, where A is an L by M matrix and B is a M by N matrix. Thus, C is an L by N matrix.

MTRXM (A,B,C): This routine multiplies matrix A by matrix B to produce matrix C, where the maximum size of A or B is 3 by 3.

MTRXT (A,B,C): This routine multiplies matrix A by the transpose of matrix B to form matrix C, where A, B, and C are 3 by 3 matrices.

MTRXTV (AT,V,W): This routine multiplies the transpose of matrix A by vector V to form vector W, where A is a 3 by 3 matrix and V and W are 3 by 1 vectors.

MTRXV (A,B,C): This routine multiplies a 3 by 3 matrix A by a vector B, and returns the answer as C.

ØRBTRT: This routine transforms the target orbit parameters to earth centered inertial coordinates. The flow chart is the same as ØRBTR.

PAD (A,B,MØD): This routine determines the delta X that produces the most precise derivative without losing significance in being rounded off.

Argument	Calls	Call
	1 to (n - 1)	n
A	f(X)	ΔX
B	f(X + ΔX)	Dummy
MOD	0	1

PØLY (N1,CØFI,ARGI): This function evaluates a polynomial of degree N1 with coefficients CØFI as a function of ARGI.

PØLY1 (A,X): This function evaluates a polynomial of degree 1 with coefficients A as a function of X.

PØLY2 (A,X): This function evaluates a polynomial of degree 2 with coefficients A as a function of X.

PØLY3 (A,X): This function evaluates a polynomial of degree 3 with coefficients A as a function of X.

PPT (P,C,M,N,S): This routine computes matrix C as matrix P times the transpose of matrix P, where P is an M by N matrix and S is a scalar that must be set to zero in the calling program.

RRTBR: This subroutine calculates body rates, given relative Euler rates.

REVOAT: This routine contains date of last program revision.

SERCH (K,IV): This routine locates the address of variable I (Hollerith) with respect to the beginning of the common reference.

SETESN: This routine resets the valves of all variables in thedictionary to their values at the minus side of event ESNI.

SINH (A): This function returns as the hyperbolic sine of A.

SØLVE (FPDATA, IDATA, XSTAR, FCT): This routine is a generalized one-dimensional iteration scheme that finds the zero of the function FCT. The argments required are:

FPDATA(1) = initial guess on the value of the argument X
FPDATA(2) = the minimum value of X
FPDATA(3) = the maximum value of X
FPDATA(4) = the tolerance on the function value
FPDATA(5) = the tolerance on the value of X
FPDATA(6) = the increment of X in the secant method
= 1 (Regula Falsi)
= 2 (secant method)
IDATA(2) = the maximum number of iterations allowed
IDATA(3) = iteration flag
= 1 (iteration limit was reached)
= 2 (X within tolerance)
= 3 (the iteration converged)
IDATA(4) = type of solution to accept
= 0 (ignore slope of the function)
= 1 (positive slope of function only)
IDATA(5) = debugging print selector
= 0 (no debugging printout)
= 1 (debugging printout)

XSTAR = the value of X that satisfies the iteration
FCT = the external function whose zero is to be found.

SP (X,Y,N): This function computes the scalar product of two N-dimensional vectors X and Y.

SREL (LIST): This routine locates the addresses of the variables in the array LIST, where LIST is constructed as follows:

NAME1,NAME2,X_1,

NAME3,NAME4,X_2,

NAME5,NAME6,X_3,

The routine is also used to find the addresses of the derivatives and their corresponding integrals.

SVDQ: This routine performs the variable step/order predictor-corrector integrator.

SVDQI: This routine performs the initialization for SVDQ.

TAB (XS,N,X,Y): This function is a generalized table lookup scheme for single-valued (monovariant) tables. The table lookup is performed using either linear or cubic interpolation. XS is x* (the value of x for which the table value is desired). N is the number of pairs of points in the table. X and Y are the values of x and f(x), respectively. If the value of x* lies beyond the table values of x, extrapolation will occur. If linear interpolation is desired, the x values immediately above and below x* are used. If cubic interpolation is desired the two x values immediately below x* and the two values immediately above x* are used. The array N in the argument list is defined as:

N(1) = number of pairs of points in the table
N(2) = type of interpolation desired
= 1 (linear interpolation)
= 3 (cubic interpolation)
N(3) = type of x's
= -1 (decreasing)
= 1 (increasing)
N(4) = pointer to the value of x that is just below the last value of x*

THPM (X,Y,XMIN,YMIN): This routine fits a quadratic polynomial through three points. It returns the minimum of this polynomial as YMIN and the minimizing value of X as XMIN.

THPØSM (X,Y,DYDX1,XMIN,YMIN): This routine fits a cubic polynomial using three points and the slope of the function. The routine returns the minimum of the polynomial as YMIN and the minimizing value of X as XMIN.

TPØSM (X,Y,DYDX1,XMIN,YMIN): This routine fits a quadratic polynomial using two points and the slope of the function. It returns the minimum of the function as YMIN and the minimizing value of X as XMIN.

TRITAB (XS,YS,ZS,N,D): This routine is a trivariant table interpolator that finds F(X*,Y*,Z*).

D = the array containing the table to be interpolated.
N(1) = number of x's
N(2) = number of y's
N(3) = number of z's
N(4) = type of interpolation
N(5) = type of x's
N(6) = type of y's
N(7) = type of z's
N(8) thru (20) = pointers for the last used point on the table.
X*, Y*, Z* = values of the table arguments

UNIT (X,XBAR,XMAG): This routine computes the magnitude, XMAG, and unit components XBAR, of a 3-dimensional vector X.

UPNØMS: This subroutine stores the current value of the steering controls (US) into the dictionary.

VCRØSS (A,B,C): This routine computes the vector cross-product, C, between two three-dimensional vectors A and B.

VDØT (X,Y): This function computes the dot product between two three-dimensional vectors X and Y.

ZERØ (A,M,N): This routine zeroes the M by N vector A.

EPHEM (DATE, GULIE, GHA, RAS, DECL): This routine computes the Greenwich hour angle, and the right ascension and declination of the sun given the month/day/year date and the floating point Julian date from 1950.0 .

FORMIN (CENJUL, GULIE1, RANUT): This routine computes the nutation of the right ascension of the Sun given the true Julian Century and the Julian date.

SUN (AJP, ARA, DEC): This routine computes the right ascension and declination of the Sun.

RUK8: This routine performs eight-order Runge-Kutta integration.

V. DEFINITION OF INTERNAL FORTRAN SYMBOLS

This section presents the symbols used internally in the program. Most variables in the program are located in COMMON to conserve core locations. Certain variables are local variables to a specific routine. These types of variables are not shown in this list but are presented along with the flow chart for that routine in the flow chart section. Variables that are either input or output variables are defined in Volume II.

Mathematical symbols are presented for each variable where applicable. The common that contains the variable is also shown, along with a definition of the variable and the routine that defines it.

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

A(I)
I=1,2
	A	LOCAL	TOTAL AERO AND THRUST MOMENTS IN PITCH AND YAW TO BE CANCELLED BY THE TRIMMING ENGINES/ TRIM

AB(I)
I=1,9
	-	DPGVC	THE TRANSFORMATION MATRIX FROM THE ATMOSPHERIC RELATIVE SYSTEM TO THE BODY SYSTEM/ IBMTRX

ACOB(I)
I=1,625
	-	SEARC	THE JACOBIAN OF THE CONSTRAINT VECTOR/ GRAD

ACOBS(I)
I=1,16
	-	LOCAL	THE JACOBIAN OF THE STEERING CONSTRAINT VECTOR/ GRADS

AR(I)
I=1,15
		MOTVC	THE TABLE LOOK-UP VALUE OF EXIT AREA FOR ENGINE I/ PROP

AXL(I)
I=1,3
	-	MOTVC	THE INERTIAL ACCELERATION VECTOR DUE TO GRAVITY AS PROPAGATED BY THE LAPLACE TWO-BODY EQUATIONS/ MOTENC

B(I)
I=1,3
		LOCAL	THE ELEMENTS OF THE B VECTOR USED TO COMPUTE THE QUATERNION RATES/ GUID2

C(I)
I=1,2
	C	LOCAL	THRUST TIMES THE PITCH AND YAW MOMENT ARMS, RESPECTIVELY/ TRIM

CADPS	-	LOCAL	THE TABLE LOOK-UP OF CADPT/ AERO
CADYS	-	LOCAL	THE TABLE LOOK-UP OF CADYT/ AERO
CDDPS	-	LOCAL	THE TABLE LOOK-UP OF CDDPT/ AERO
CDDYS	-	LOCAL	THE TABLE LOOK-UP OF CDDYT/ AERO
CLDPS	-	LOCAL	THE TABLE LOOK-UP OF CLDPT/ AERO
CMDPS	-	LOCAL	THE TABLE LOOK-UP OF CMDPT/ AERO
CNDPS	-	LOCAL	THE TABLE LOOK-UP OF CNDPT/ AERO

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

CNP1(I)
I=1,30
	-	DYTEM	CORRECTED SOLUTION IN THE FOURTH ORDER PREDICTOR-CORRECTOR INTEGRATION FORMULA/ FOPC

CTHAT	-	SEARC	COSINE OF CONEPS(1)/ TEST
CWDYS	-	LOCAL	THE TABLE LOOK-UP OF CWDYT/ AERO
CYCF	-	CYCVC	CYCLING FLAG SUCH THAT IF =0, THIS IS A DERIVATIVE PASS WITH DELT=0/ CYCXMI, CYCXM
CYDYS	-	LOCAL	THE TABLE LOOK-UP OF CYDYT/ AERO

D(I)
I=1,3
		LOCAL	THE ELEMENTS OF THE D VECTOR USED TO COMPUTE THE QUATERNION RATES/ GUID2

DAX(I)
I=1,3
	-	MOTVC	THE DIFFERENCE BETWEEN THE INERTIAL ACCELERATION COMPUTED VIA MOTION AND THE INERTIAL ACCELERATION COMPUTED VIA TWOBDY/ MOTENC

DELT	-	CYCVC	CURRENT INTEGRATION STEP SIZE/ CYCXM

DG(I)
I=1,25
	-	SEARC	THE DIFFERENCE BETWEEN THE COST GRADIENT ON TWO SUCCESSIVE ITERATIONS/ DGM

DFVLH(I)
I=1,3
	-	MOTVC	THE HISTORY VALUES OF THE FUNCTIONAL INEQUALITY CONSTRAINT DERIVATIVES/ MOTION

DMI(I)
I=1,15
	-	MOTVC	THE TABLE LOOK-UP VALUE OF FLOWRATE FOR ENGINE I/ PROP

DPR	-	SERVC	DEGREES PER RADIAN CONVERSION FACTOR/ BLKDAT
DTIME	-	CYCVC	THE DERIVATIVE OF TIME, I.E., 1.0
DTO	-	CYCVC	THE INTEGRATION STEP SIZE IN THE CURRENT PHASE/ CYCXMI

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
DTSAV	-	DYTEM	THE LAST INTEGRATION STEP SIZE. USED TO CHECK FOR CHANGES IN STEP SIZE WHEN USING THE PREDICTOR-CORRECTOR FORMULA/ DYNS2

DXP(I)
DYP(I)
DZP(I)
I=1,15
		LOCAL	THE THRUST MOMENT ARM OF ENGINE I/ TRIM

DXR(I)
I=1,3
		LOCAL	THE AERODYNAMIC MOMENT ARM/ TRIM

DYNIL(I)
I=1,148
	-	DYNIL	AN ARRAY WHICH CONTAINS THE INTEGRALS, DERIVATIVES, AND A FLAG FOR EACH VARIABLE TO INDICATE WHETHER IT IS TO BE INTEGRATED/ BLKDAT

EA(I)
I=1,25
	-	SEARC	THE ERROR VECTOR FOR THE ACTIVE CONSTRAINTS/ PGM

EMAP(I)
I=1,625
	-	SEARC	THE TRANSFORMATION THAT DETERMINES THE DIRECTION OF SEARCH FOR CONSTRAINT SATISFACTION/ UPDATS

ENOIS	-	CYCVC	A SMALL NUMBER USED AS A TOLERANCE TEST

ENOM(I)
I=1,25
		SEARC	THE NOMINAL TARGET ERRORS/ NOMINL

ENOMS(I)
I=1,4
	-	DPGVC	THE NOMINAL STEERING ERRORS/ GRADS

EPSA(I)
I=1,4
	-	SEARC	THE STEPSIZE CONTROL FOR THE ONE-DIMENSIONAL MINIMIZATION ROUTINE, I.E., UPPER AND LOWER BOUNDS AND CURVEFIT ERROR TOLERANCES/ FGAMA, TRYIT1, TRYIT2

ESN	-	TGOVC	THE CURRENT EVENT SEQUENCE NUMBER/ DINPT
ESNPR1	-	INFVC	THE CURRENT EVENT SEQUENCE NUMBER FOR PRINTOUT/ INFXMI

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

EVNT(I)
I=1,8
	-	REDAT	THE EVENT SEQUENCE NUMBER AND CRITERIA ARRAY/ READAT

EVTF	-	PHZVC	EVENT FLAG/ PHZXMI =0, NOT AN EVENT =1, ON MINUS SIDE OF AN EVENT =2, ON PLUS SIDE OF AN EVENT
EXTRAP	-	INFVC	THE VALUE OF TIME WHEN THE LAST PROFIL TIME SLICE WAS WRITTEN. USED TO COMPUTE THE PROFIL WRITE INTERVAL/ INFXM
FPP5	-	SERVC	FLOATING POINT NUMBER = .5
FP00 • • • FP15	-	SERVC	FLOATING POINT NUMBERS 0,1,...,15
FP60	-	SERVC	FLOATING POINT 60
FP90	-	SERVC	FLOATING POINT 90
FP180	-	SERVC	FLOATING POINT 180
FP270	-	SERVC	FLOATING POINT 270
FP360	-	SERVC	FLOATING POINT 360

FUXN(I)
I=1,10
	-	TGOVC	THE HISTORY VALUES OF THE CRITERIA VARIABLES CURRENTLY BEING MONITORED/ TGOEM

GAMASS	-	SEARC	THE STEPSIZE FOR THE NON-UNITIZED DIRECTION OF SEARCH/ TRYIT1
GAMAX	-	SEARC	THE MAXIMUM STEPSIZE ALLOWED CONSIDERING THE INACTIVE INEQUALITY CONSTRAINTS, AND THE MAXIMUM ALLOWED PERCENTAGE CHANGE IN THE CONTROL PARAMETERS/ TRYIT1

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

GB(I)
I=1,9
	-	DPGVC	THE TRANSFORMATION MATRIX FROM THE GEOGRAPHIC SYSTEM TO THE BODY SYSTEM/ IBMTRX

GCON(I)
I=1,3
	-	LOCAL	CONSTANTS USED BY GRAV IN COMPUTING THE GRAVITATIONAL POTENTIAL/ MOTIAL

GP(I)
I=1,25
	-	SEARC	THE COST GRADIENT ON THE PREVIOUS ITERATION/ PGM, DGM, DGMP2

GSQOLD	-	SEARC	THE SQUARE OF THE GRADIENT MAGNITUDE ON THE PREVIOUS ITERATION/ CGM

HARG(I)
I=1,3
	-	DPGVC	HISTORY VALUE OF THE ARGUMENT OF THE GDF(I) TABLE USED IN GENERALIZED LINEAR FEEDBACK STEERING/ GUIDX

HDG(I)
I=1,25
	-	SEARC	THE ESTIMATE OF THE HESSIAN MATRIX GENERATED IN THE DAVIDON ALGORITHM/ DGM

HEADER(I)
I=1,10
	-	INFIC	THE TITLE TO BE PRINTED OUT AT THE TOP OF EACH PAGE/ INFXMI

HEATC	-	LOCAL	CONSTANT USED IN AERO HEATING CALCULATION/ MOTIAL

HERROR(I)
I=1,3
	-	DPGVC	HISTORY VALUE OF ERROR TERM IN GENERALIZED LINEAR FEEDBACK STEERING EQUATIONS/ GUIDX

HESS(I)
I=1,325
	-	SEARC	THE APPROXIMATION TO THE HESSIAN MATRIX GENERATED IN THE DAVIDON ALGORITHM/ DGM, DGMP2, PGM

I,J,K,L,M	-	SERVC	INTEGER VARIABLES FOR TEMPORARY USE

IA(I)
I=1,9
	-	DPGVC	TRANSFORMATION MATRIX FROM PLANET CENTERED INERTIAL TO ATMOSPHERIC RELATIVE FRAME/ IBMTRX

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

IACS(I)
I=1,25
	-	SEARC	THE SAVED VALUES OF THE SUBSCRIPTS OF THE ACTIVE CONSTRAINTS/ TRYIT1

IBKT(I)
I=1,2000
	-	BLANK	A DATA BUFFER WHICH CONTAINS THE EVENT CRITERIA AND THE TABLE INPUT DATA/ READAT, DINPT

IBTC(I)
I=1,25
	-	OVRLY25	A COMBINATION OF ACTIVE CONSTRAINT INDICES USED TO DETERMINE IF SOME CONSTRAINTS CAN BE MADE INACTIVE/ DROP

ICASE	-	MULTRC	CURRENT CASE (PROBLEM) NUMBER/ READAT
ICD	-	OVRLY25	INDEX OF THE ACTIVE CONSTRAINT WHICH WAS DROPPED/ REVISE
ICGM	-	SEARC	A FLAG TO INDICATE THAT INITIALIZATION MUST BE DONE IN THE CONJUGATE GRADIENT ROUTINE (CGM) IF ICGM IS NON-ZERO/ DELTU
IDAV	-	SEARC	NOT USED.

IDENT(I)
I=1,9
	-	SERVC	THE IDENTITY MATRIX (3X3)/ BLKDAT

IDTAB(I)
I=1,5
	-	CYCVC	ADDRESSES OF THE TABLES THAT ARE TO BE USED IN COMPUTING THE INTEGRATION STEP SIZE DURING THE CURRENT PHASE/ CYCXM, DTMDL

IDX	-	REDAT	INDEX USED IN READAT TO MERGE MULTIPLE RUN DATA/ READAT
IENT	-	REDAT	CURRENT ESN BEING MATCHED DURING GENERATION OF MULTIPLE RUN DATA/ READAT
IERRX	-	REDAT	USED TO SAVE FATAL ERROR FLAG BEFORE EVALUATING NON-FATAL ERRORS/ READAT

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
IESN	-	PHZVC	THE INITIAL EVENT SEQUENCE NUMBER/ DINPT

IEVNT(I)
I=1,10
	-	TGOVC	THE ARRAY OF EVENT LOCATIONS CURRENTLY BEING MONITORED/ TGOEMI

IFLG	-	CYCVC	A FLAG USED TO INDICATE THAT AN EVENT HAS BEEN INITIATED/ CYCXM
IFRST	-	REDAT	A FLAG TO INDICATE THAT THIS IS THE BEGINNING OF A PROBLEM/ READAT

IG(I)
I=1,9
	-	DPGVC	THE TRANSFORMATION MATRIX FROM THE PLANET CENTERED INERTIAL TO THE GEOGRAPHIC SYSTEM/ MOTION

IGEN(I)
I=1,2000
	-	GENRL	A DATA BUFFER WHICH CONTAINS THE GENERAL INPUT DATA/ READAT, DINPT

IHADIT	-	SEARC	A FLAG WHICH IS SET TO 1 IF THE PROGRAM COULD NOT GET TARGETED ON THE LAST OPTIMIZATION STEP/ TRYIT1, TRYIT2
II	-	REDAT	USED AS INDEX AND COUNTER DURING DATA PROCESSING IN READAT/ READAT

IL(I)
I=1,9
	-	DPGVC	THE TRANSFORMATION MATRIX FROM THE PLANET CENTERED INERTIAL TO THE LAUNCH INERTIAL SYSTEM/ MOTIAL

IMAX	-	SEARC	THE LAST PHASE NUMBER OF DEPPH(I) AND OPTPH TO OCCUR IN TIME/ MINMYS
IMIN	-	SEARC	THE FIRST PHASE NUMBER OF DEPPH(I) AND OPTPH TO OCCUR IN TIME/ MINMYS
IMLT	-	REDAT	POINTS TO THE BEGINNING OF LABELED COMMON MNMMLT IN IV/ READAT
IMULT	-	MULTRC	SIZE OF GENERAL DATA RECORD SAVED FOR MULTIPLE RUNS/ READAT

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
IN IO	-	MULTRC	CURRENT INPUT AND OUPUT FILE FOR MULTIPLE RUN - FLIP FLOPS BETWEEN 3 AND 4/ READAT

IND(I)
I=1,25
	-	OVRLY25	COMBINATION OF CONSTRAINTS SELECTED BASED ON COMBINATORIAL PROCEDURES/ COMBINE

INFF	-	INFVC	A PRINT FLAG WHICH FORCES A PRINT WHEN SET NONZERO/ PHZXM
INIT	-	DYTEM	A COUNTER IN PREDICTOR-CORRECTOR TO START THE ALGORITHM/ DYNXB, DYNS2
INSRT	-	REDAT	FLAG USED TO INDICATE IF A TABLE HAS BEEN INSERTED FOR THE CURRENT PHASE BEING PROCESSED IN MULTIPLE RUN PORTION OF READAT/ READAT
INTRBL	-	SEARC	A FLAG TO INDICATE THAT THE PROGRAM COULD NOT GET TARGETED ON THE CURRENT OPTIMIZATION STEP/ TRYIT2, TEST
INTRY1	-	SEARC	A FLAG TO INDICATE THAT OVERLAY (2,5) IS TO BE CALLED WHEN SET NON-ZERO/ TRYIT1
IOPT	-	SEARC	A FLAG WHICH IS SET NON-ZERO TO INDICATE THAT THERE WERE NO PREVIOUS OPTIMIZATION STEPS/ TRYIT1
IPRT	-	GENIC	A FLAG WHICH SUPPRESSES THE TRAJECTORY PRINTOUT WHEN SET TO ZERO/ MINMYS
IPRNTB	-	INFVC	THE NUMBER OF FULL PRINT LINES IN THE PRINT BLOCK/ INFXM
IPRNTR	-	INFVC	THE NUMBER OF REMAINING PRINT VARIABLES IN THE LAST LINE THAT IS NOT FULL/ INFXM

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
IRANGE	-	AUXVC	A FLAG TO INDICATE THAT THE RANGE CALCULATION HAS BEEN INITIALIZED/ AUXFMI
IREVDT	-	GENIC	REVISION DATE PRINTED AT THE BEGINNING OF EACH RUN IN READAT/ BLKDAT
IRUNF	-	REDAT	FLAG USED TO INDICATE IF CURRENT CASE IS TO BE BYPASSED. IRUNF=0 IMPLIES CASE WILL BE BYPASSED/ READAT
IR1(1)	-	SERVC	A TEMPORARY REUSABLE ARRAY

ISCORE(I)
I=1,2
	-	MULTRC	WORDS IN WHICH SUCCESS OR FAILURE OF RUNS ARE PACKED/ RSCORE

ISFLG	-	SEARC	FLAG WHICH INDICATES THAT THIS IS A RESTART ITERATION FOR THE DAVIDON OPTION/ RSEARCH
ISTART	-	SEARC	A FLAG TO INDICATE THAT THE DAVIDON ALGORITHM IS TO BE RE-STARTED IF SET NON-ZERO/ TRYIT1, PGM

ISTC(I)
I=1,25
	-	OVRLY25	INDICES OF THE SAVED TIGHT CONSTRAINTS/ DROP

ISTEPS	-	DPGVC	A FLAG WHICH IS SET NON-ZERO IF A STEERING SENSITIVITY MATRIX HAS BEEN COMPUTED IN THIS PHASE/ GSA

ISTOP(I)
I=1,4
	-	SEARC	AN ARRAY TO INDICATE HOW THE CURRENT PROBLEM TERMINATED/ NOMINL, TEST
			ISTOP(1)=77B, PROBLEM SOLVED
			ISTOP(2)=77B, ITERATION LIMIT
			ISTOP(3)=77B, NO CHANGE IN STATE
			ISTOP(4)=77B, TIME LIMIT

ISZBLK	-	MULTRC	SIZE OF SEARCH DATA RECORD SAVED FOR MULTIPLE RUNS/ READAT
ISZEV	-	TGOVC	THE NUMBER OF EVENTS BEING MONITORED/ TGOEMI

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
ISIZ	-	REDAT	THE SIZE OF THE TABLE CURRENTLY BEING STORED IN IBKT/ READAT
ISV	-	MOTVC	A POINTER TO THE LAST POINT USED IN THE ATMOSPHERE TABLE LOOK-UP/ ATMOS2

ITC(I)
I=1,25
	-	SEARC	THE INDICES OF THE TIGHT INEQUALITY CONSTRAINTS/ NOMINL, REVISE, DROP

ITERF	-	SEARC	A FLAG TO INDICATE THE TYPE OF ITERATION STEP/ MINMYS =0, TARGETING ONLY =1, OPTIMIZATION ONLY =2, TARGETING AND OPTIMIZATION
ITRIM	-	LOCAL	FLAG TO INDICATE IF TRIM IS BY ENGINES (= 0) OR BY FLAPS (= 1)/ MOTIAL

IVCRT(I)
I=1,3
	-	DPGVC	ARRAY OF INDICES POINTING TO THE EVENT CRITERIA IN IV USED TO REACH THE DESIRED ANGLE WHEN USING THE GENERALIZED LINEAR COMMANDS/ GUIDI, GUID1

IVETA	-	DPGVC	INDEX POINTING TO EVENT CRITERIA USED TO REACH DESIRED ETA WHEN USING GENERALIZED LINEAR COMMANDS/ MOTIAL, PROP
IVSZ	-	SERVC	THE SIZE OF THE COMPUTATIONAL COMMON REGION (END-IV+1)/ READAT
IWTFLG	-	SEARC	A FLAG TO INDICATE THAT THE DEPENDENT AND INDEPENDENT VARIABLES ARE WEIGHTED OR UNWEIGHTED/ WUCAL
IXBKT	-	REDAT	THE NUMBER OF CELLS OCCUPIED BY THE CONTENTS OF IBKT/ READAT

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
IXE	-	REDAT	A POINTER TO THE EVENT IN IBKT/ READAT
IXEVG	-	REDAT	A POINTER USED IN THE GENERAL DATA ARRAY (IGEN)/ READAT
IXEVT	-	REDAT	A POINTER USED IN THE TABLE DATA ARRAY (IBKT)/ READAT
IXEVT	-	TGOVC	AN INDEX ON IEVNT/ TGOEMI
IXG	-	REDAT	POINTS TO BEGINNING OF CURRENT PHASE IN NEW CASE DATA SET/ READAT
IXGEN	-	REDAT	THE NUMBER OF CELLS OCCUPIED BY THE CONTENTS OF IGEN/ READAT
IXT	-	REDAT	AN INDEX USED TO PACK THE TABLES INTO IBKT/ READAT
IX1 IX2	-	REDAT	INDICES USED TO BUILD THE DATA BUFFERS/ READAT
IXTBL	-	REDAT	THE CORE LOCATION OF THE TABLE BEING INSERTED INTO IBKT/ READAT

JAC(I)
I=1,26
	-	SEARC	THE INDICES OF THE ACTIVE CONSTRAINTS/ TRYIT1

JMLT	-	REDAT	POINTS TO END OF LABELED COMMON HOLINC IN IV/ READAT
JMULT	-	MULTRC	SIZE OF TABLE DATA RECORD SAVED FOR MULTIPLE RUN/ READAT
KREEP	-	SEARC	A FLAG WHICH INDICATES IF THE ITERATION IS PROGRESSING TOWARDS A SOLUTION/ TEST

LB(I)
I=1,9
	-	DPGVC	THE TRANSFORMATION MATRIX FROM THE LAUNCH INERTIAL TO THE BODY SYSTEM/ IBMTRX

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
NFLAG	-	SEARC	A FLAG WHICH IS SET POSITIVE TO INDICATE THAT THE PROBLEM HAS CONVERGED/ TEST
NINDVX	-	REDAT	NUMBER OF INDEPENDENT PARAMETERS SAVED FROM LAST RUN/ READAT

NLDADR(I)
I=1,25
	-	REDAT	ADDRESS OF THE CONTROLS SAVED FROM THE LAST SEARCH RUN/ READAT

NLDPH(I)
I=1,25
	-	REDAT	PHASE NUMBERS ASSOCIATED WITH THE CONTROLS SAVED FROM THE LAST SEARCH RUN/ READAT

NOMF	-	SEARC	A FLAG WHICH IS SET NON-ZERO IF THE TRAJECTORY BEING RUN IS A NOMINAL TRAJECTORY/ NOMINL
NOO	-	SERVC	FIXED POINT ZERO
NO1 • • • NO15	-	SERVC	FIXED POINT NUMBERS 1,...,15
NPAGE	-	MULTRC	PAGE COUNTER/ PAGER
NPASS	-	DYNVC	AN INTEGRATION PASS FLAG/ DYNXA, RUK
NPC9 NPC13 NPC17	-	LOCAL	INPUT VALUES OF NPC(9), NPC(13), AND NPC(17) THAT ARE REQUIRED FOR PRINTOUT/ PRINTIC
NSTEP	-	SEARC	AN ITERATION COUNTER FOR THE CURRENT PROBLEM/ MINMYS
NTC	-	SEARC	THE NUMBER OF TIGHT CONSTRAINTS/ REVISE

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
NULL	-	SERVC	A VARIABLE USED TO DETECT IF INPUT VARIABLES HAVE BEEN INPUT. THE STORED VALUE OF NULL IS 1HU/ BLKDAT
OLDG2		SEARC	G2MAG FROM THE PREVIOUS ITERATION/ TEST
OLDP1 OLDP2		SEARC	P1NOM AND P2NOM FROM THE PREVIOUS ITERATION/ TEST
OLDU		SEARC	UMAG FROM THE PREVIOUS ITERATION/ TEST
OMGSLT	-	LOCAL	OMEGA*SIN(LATC)/ MOTIAL, MOTION
ONE	-	MNMMLT	THE NAME OF A CELL WHICH CONTAINS FLOATING POINT ONE/ NOMHOL
PCTGC	-	TGOVC	PERCENT OF NOMINAL INTEGRATION STEP ALLOWED BY TIME-TO-GO LOGIC TO BRACKET THE DESIRED FUNCTION VALUE/ TGOEM
PCTOLD	-	SEARC	THE MAXIMUM PERCENTAGE CHANGE ALLOWED ON THE PREVIOUS OPTIMIZATION STEP/ TRYIT1

PE(I)
I=1,198
	-	INFIC	THE CORE ADDRESSES AND NAMES OF THE CURRENT PRINT VARIABLES/ INFXMI, INFXM

PGT(I)
I=1,45
	-	TRACKC	PLANET TO TRACKER GEOGRAPHIC TRANSFORMATION MATRIX. (3X3 MATRIX FOR 5 TRACKERS)/ AUXFMI, TRACKER

PHZF	-	PHZVC	A FLAG TO INDICATE THAT THE TRAJECTORY IS TO BE TERMINATED/ PHZXM =0, DO NOT TERMINATE THE TRAJECTORY NE 0, TERMINATE THE TRAJECTORY
PI		SERVC	PI/ BLKDAT

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
LIMGEK	–	REDAT	THE MAXIMUM SIZE OF THE GENERAL DATA ARRAY (IGEN)/ READAT
LIMBKT	–	REDAT	THE MAXIMUM SIZE OF THE TABLE DATA ARRAY (IBKT)/ READAT
LISTD(1) I=1,30	–	DYTEM	THE ADDRESSES OF THE CURRENT DERIVATIVES/ DYS11, DLOOK
LISTI(I) I=1,30	–	DYTEM	THE ADDRESSES OF THE CURRENT INTEGRALS/ DYSI1, DLOOK
LMBKT	–	REDAT	THE CURRENT SIZE OF IBKT/ READAT
LMEVT	–	REDAT	THE SIZE OF THE CURRENT EVENT BEING INPUT/ READAT
LNGTH	–	REDAT	THE LENGTH OF THE COMPUTATIONAL COMMON AREA/ READAT
LPRNT	–	INFVC	THE LAST PRINT TIME/ INFXMI, INFXM
N	–	DYTEM	THE NUMBER OF INTEGRALS IN THE CURRENT PHASE/ DYSI1, DLOOK
NACS	–	SEARC	THE NUMBER OF ACTIVE CONSTRAINTS INCLUDING THE EQUALITY CONSTRAINTS/ TRYIT1
NAMSVR(I) I=1,51	–	SEARC	THE HOLLERITH NAMES OF THE CONTROL, TARGET, AND OPTIMIZATION VARIABLES/ READAT
NDISC	–	DYNVC	A FLAG TO SIGNAL THE INTEGRATION ALGORITHM THAT THERE IS A DISCONTINUITY/ DYNXA
NEQC	–	SEARC	THE NUMBER OF EQUALITY CONSTRAINTS/ MINMYS
NETF	–	SEARC	A FLAG WHICH IS SET NON-ZERO IF SRCHM=4 AND OPT IS NON-ZERO/ MINMYS

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
PIF	-	PHZVC	A PROGRAM INITIALIZATION FLAG/ PHZXMI =0, THIS IS THE INITIALIZATION PASS =1, THIS IS NOT THE INITIALIZATION PASS
PIO2		SERVC	PI OVER TWO/ BLKDAT
PITMOM YAWMOM	-	LOCAL	THE TOTAL MOMENTS IN PITCH AND YAW/ TRIM

PNMCN(I)
I=1,30
	-	DYTEM	FIRST DIFFERENCE BETWEEN PREDICTED SOLUTION AND CORRECTED SOLUTION USED IN FOURTH ORDER PREDICTOR-CORRECTOR FORMULA/ FOPC

PNP1(I)
I=1,30
	-	DYTEM	PREDICTED SOLUTION IN FOURTH ORDER PREDICTOR-CORRECTOR INTEGRATION FORMULA/ FOPC

PROJ(I)
I=1,625
	-	OVRLY25	THE MATRIX WHICH DETERMINES THE PROJECTED GRADIENT/ UPDATS, REVISE

P1NOM P2NOM	-	SEARC	THE VALUES OF P1 AND P2 ON THE NOMINAL TRAJECTORY/ MINMYS
P2MIN	-	SEARC	THE LOWER BOUND ON THE WEIGHTED ERROR MAGNITUDE/ DELTU

OLDU(I)
I=1,25
	-	REDAT	CONTROLS SAVED FROM LAST SEARCH RUN/ READAT

QSREF	-	LOCAL	DYNAMIC PRESSURE TIMES THE AERODYNAMIC REFERENCE AREA/ MOTION
RERP2	-	LOCAL	(RE/RP)**2/ MOTIAL
REVNT	-	REDAT	A VARIABLE USED TO SET A FLAG FOR A ROVING EVENT/ READAT
RPD	-	SERVC	RADIANS PER DEGREE CONVERSION FACTOR/ BLKDAT

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

S(I)
I=1,25
	-	SEARC	THE VALUE OF DU FROM THE PREVIOUS ITERATION/ CGM

SALFA CALFA	-	LOCAL	THE SINE AND COSINE OF THE ANGLE OF ATTACK/ GUID1, MOTION

SAVE(I)
I=1,70
	-	TGOVC	THE VALUES OF THE STATE VARIABLES AT THE LAST INTEGRATION STEP/ TGOEM

SAVIT(I,J)
I=1,27
J=1,5
	-	SEARC	AN ARRAY IN WHICH THE RESULTS OF THE CURVEFIT OF P1 AND P2 ARE SAVED/ FGAMA

SBETA CBETA	-	LOCAL	THE SINE AND COSINE OF THE ANGLE OF SIDESLIP/ GUID1, MOTION

SFCT(I)
I=1,15
	-	LOCAL	THE TABLE LOOK-UP VALUE OF THE SPECIFIC FUEL CONSUMPTION FOR ENGINE I/ PROP

SGAM CGAM	-	LOCAL	THE SINE AND COSINE OF THE PATH ANGLE RELATIVE TO THE ATMOSPHERE/ GUID1, GUID2
SIDEAL	-	AUXVC	THE SAVED VALUE OF VIDEAL USED TO COMPUTE THE REQUIRED VELOCITY MARGIN/ AUXFM1

SINDPH(I)
I=1,25
	-	SEARC	THE VALUES OF INDPH(I) WHICH ARE SAVED IN READAT FOR PRNTIC/ READAT

SIPT CIPT	-	LOCAL	THE SINE AND COSINE OF THE THRUST INCIDENCE ANGLE IN PITCH REQUIRED TO TRIM THE VEHICLE/ TRIM
SIYT CIYT	-	LOCAL	THE SINE AND COSINE OF THE THRUST INCIDENCE ANGLE IN YAW REQUIRED TO TRIM THE VEHICLE/ TRIM
SLAM CLAM	-	LOCAL	THE SINE AND COSINE OF THE AZIMUTH ANGLE RELATIVE TO THE ATMOSPHERE/ GUID1, GUID2

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
SLAMRF CLAMRF	-	LOCAL	THE SINE AND COSINE OF THE REFERENCE AZIMUTH ANGLE USED IN THE RANGE CALCULATIONS/ AUXFMI
SLATC CLATC TLATC	-	LOCAL	THE SINE, COSINE, AND TANGENT OF THE GEOCENTRIC LATITUDE OF THE VEHICLE/ MOTIAL, MOTION
SLATRF CLATRF	-	LOCAL	THE SINE AND COSINE OF THE REFERENCE LATITUDE USED IN THE RANGE CALCULATIONS/ AUXFMI
SLG CLG	-	LOCAL	SINE AND COSINE OF GEODETIC LATITUDE/ MOTIAL
SLONG CLONG	-	LOCAL	SINE AND COSINE OF INERTIAL LONGITUDE/ MOTIAL, MOTION
SOLVED	-	MULTRC	FLAG SET TO INDICATE HOW THE LAST PROBLEM TERMINATED/ SCORE
SSIGMA CSIGMA	-	LOCAL	THE SINE AND COSINE OF THE BANK ANGLE/ GUID1, MOTION

SSTI(I)
I=1,625
	-	OVRLY25	A MATRIX USED IN COMPUTING THE PROJECTION MATRIX AND IS EQUAL TO (S*S PRIME)INVERSE/ UPDATS, PGM

SSVIDL	-	AUXVC	THE ACCUMULATED SUM SQUARED VALUE OF VIDEAL FOR EACH STAGE. COMPUTED AT THE BEGINNING OF EACH PHASE/ AUXFMI

STEMP(I)
I=1,25
	-	SERVC	AN ARRAY USED FOR TEMPORARY STORAGE

STEP(I)
I=1,6
		SEARC	THE TRIAL STEP LENGTH FOR EACH TRIAL STEP/ GENMIN

STKLAT(I)
CTKLAT(I)
I=1,5
	-	TRACKC	SINE AND COSINE OF GEOCENTRIC LATITUDE OF TRACKING STATION I/ AUXFMI, TRACKER

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

STKLON(I)
CTKLON(I)
I=1,5
	-	TRACKC	SINE AND COSINE OF THE LONGITUDE OF TRACKING STATION I/ AUXFMI TRACKER

STMINP(I)
I=1,2
		SEARC	THE MINIMUM STEP SIZE DECREASE FROM THE PREVIOUS STEP WHEN GENERATING THE CURVEFIT/ GENMIN

STPMAX		SEARC	THE LENGTH OF THE STEP IN THE DIRECTION OF SEARCH REQUIRED TO REACH THE BOUNDARY FOR THE NEAREST INEQUALITY CONSTRAINT/ TRYIT1, TRYIT2

TABLE(I)
I=1,1500
	-	REDAT	THE CURRENT TABLE BEING INPUT. USED TO TRANSFER THE TABLE FROM INPUT TO THE STORAGE ARRAY IBKT/ RTAB, READAT

TEMP(I)
I=1,50
	-	SERVC	AN ARRAY USED FOR TEMPORARY STORAGE

TGO	-	TGOVC	THE TIME TO GO TO THE NEXT EVENT/ TGOEM
TGRAD	-	SEARC	TIME REQUIRED TO COMPUTE ALL TRAJECTORY SENSITIVITIES/ MINMYS
THRSTT	-	LOCAL	TOTAL THRUST OF THE TRIMMING ENGINES/ PROP

TI(I)
I=1,15
	-	MOTVC	THE VALUE OF NET THRUST FOR ENGINE I/ PROP

TIMIN	-	SEARC	THE TIME AT WHICH THE EARLIEST PHASE IN INDPH(I) OCCURS/ NOMINL, SAVIC
TIMX	-	TGOVC	THE LAST VALUE OF TIME/ TGOEM

TJD(I)
I=1,15
	-	LOCAL	THE TABLE LOOK-UP VALUE OF THRUST FOR ENGINE I/ PROP

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

TKLATC(I)
I=1,5
	-	TRACKC	GEOCENTRIC LATITUDE OF TRACKING STATION I/ AUXFMI

TKRAD(I)
I=1,5
	-	TRACKC	GEOCENTRIC RADIUS TO TRACKING STATION I/ AUXFMI

TREF	-	CYCVC	A TIME REFERENCE USED TO COMPUTE THE NEXT STEP SIZE (DELT)/ CYCXMI, CYCXM
TREFP	-	DYNVC	THE TIME REFERENCED TO THE LAST PRIMARY EVENT, CYCXMI, TGOEMI
TREFS	-	DYNVC	THE TIME REFERENCED TO THE LAST SECONDARY EVENT/ CYCXMI, TGOEMI

TRKXRI(I)
I=1,15
	-	TRACKC	THE EARTH CENTERED INERTIAL COORDINATES OF TRACKING STATION I. (X,Y,Z) FOR 5 TRACKING STATIONS/ AUXFMI, TRACKER

TTRIAL	-	SEARC	TIME REQUIRED FOR A TRIAL STEP IN THE ONE DIMENSIONAL MINIMIZATION ROUTINE/ FGAMA
TWOPI		SERVC	TWO TIMES PI/ BLKDAT
TVACA	-	LOCAL	TOTAL VACUUM THRUST OF THE ENGINES USED TO LIMIT ACCELERATION/ PROP

VXL(I)
I=1,3
	-	MOTVC	THE INERTIAL VELOCITY VECTOR PROPAGATED BY THE LAPLACE EQUATIONS IN TWOBODY/ MOTENC

VXLO(I)
I=1,3
	-	MOTVC	INITIAL INERTIAL VELOCITY VECTOR TO BE PROPAGATED BY THE LAPLACE EQUATIONS/ MOTENC

WGTO	-	LOCAL	THE CALCULATED VALUE OF INITIAL STAGE WEIGHT/ WGTINI
WJETTO	-	LOCAL	THE SAVED VALUE OF JETTISON WEIGHT FOR PRINTOUT PURPOSES/ WGTINI

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE
WPROPO	-	LOCAL	INITIAL STAGE PROPELLANT WEIGHT/ WTGINI, PROP
WPUSDH	-	LOCAL	PROPELLANT CONSUMED UP TO AND INCLUDING LAST PHASE/ WTGINI

XB(I)
I=1,2
	B	LOCAL	THRUST TIMES THE PITCH AND YAW MOMENT ARMS/ TRIM

XINF	-	SERVC	FLOATING POINT INFINITY (10.E10)/ BLKDAT

XISAV(I)
I=1,3
	-	AUXVC	THE SAVED VALUES OF THE VEHICLE POSITION VECTOR RELATIVE TO THE EARTH FOR USE IN COMPUTING DPRNG1 AND DPRNG2/ AUXFMI

XL(I)
I=1,3
	-	MOTVC	THE INERTIAL POSITION VECTOR PROPAGATED BY THE LAPLACE EQUATIONS/ MOTENC

XLO(I)
I=1,3
	-	MOTVC	INITIAL INERTIAL POSITION VECTOR TO BE PROPAGATED BY THE LAPLACE EQUATIONS/ MOTENC

XYOMA(I)
XYOME(I)
XYOMS(I)
I=1,3
	-	DPGVC	CONSTANTS USED TO COMPUTE COMMANDS WHEN USING GENERALIZED LINEAR ANGLE OF ATTACK, SIDESLIP, AND BANK ANGLE COMMANDS/ GUIDI, GUIDI
			XYOMA(1)=INITIAL VALUE OF CRITR
			XYOMA(2)=INITIAL ANGLE VALUE
			XYOMA(3)=SLOPE

XYOME(I)
I=1,3
	-	DPGVC	CONSTANTS USED TO COMPUTE THROTTLE COMMAND WHEN USING GENERALIZED LINEAR COMMANDS/ MOTIAL, PROP
			XYOME(1)=INITIAL VALUE OF CRITR
			XYOME(2)=INITIAL ETA VALUE
			XYOME(3)=SLOPE

DEFINITION OF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBOL	MATH SYMBOL	COMMON	DEFINITION/SUBROUTINE

XYOM1(I)
XYOM2(I)
XYOM3(I)
I=1,3
	-	DPGVC	CONSTANTS USED TO COMPUTE COMMANDS WHEN USING GENERALIZED LINEAR INERTIAL OR RELATIVE EULER ANGLE COMMANDS/ GUIDI, GUIDI
			XYOM1(1)=INITIAL VALUE OF CRITR
			XYOM1(2)=INITIAL ANGLE VALUE
			XYOM1(3)=SLOPE

YAWRC	-	LOCAL	VALUE OF YAWR SAVED AT END OF A PHASE FOR PRINTOUT/ GUIDI
YAWRH	-	MOTVC	HISTORY VALUE OF YAW ATTITUDE. USED WHEN RE-INITIALIZING BODY ATTITUDES BETWEEN EVENTS/ GUIDI

YES(I)
I=1,4
	-	SEARC	THE VALUES OF THE POLYNOMIALS GENERATED DURING THE ONE DIMENSIONAL MINIMIZATION AT THEIR PREDICTED MINIMUMS/ TRYIT1

YP(I,J)
I=1,4
J=1,30
	-	DYTEM	BACK DERIVATIVES USED IN FOURTH ORDER PREDICTOR-CORRECTOR FORMULA/ FOPC

YPVAV(I)
I=1,30
	-	DYTEM	INITIAL VALUES OF INTEGRALS USED IN PREDICTOR-CORRECTOR FORMULA/ FOPC

Y1(I)
Y2(I)
I=1,30
	-	DYTEM	WORKING STORAGE USED BY THE RUNGE-KUTTA INTEGRATION ALGORITHM/ RUK

Z1TRY(I)
I=1,6
	-	SEARC	SAVED VALUES OF P1TRY(I)/ TRYIT1

Z2TRY(I)
I=1,6
	-	SEARC	SAVED CALUES OF P2TRY(I)/ TRYIT1

ZMAG	-	SEARC	SAVED VALUE OF UMAG/ TRYIT1
ZMAX	-	SEARC	SAVED VALUE OF STPMAX/ TRYIT1

DEFINITION CF INTERNAL FORTRAN SYMBOLS (CONTD)

FORTRAN SYMBCL	MATH SYMBCL	COMMON	DEFINITION/SUBROUTINE
ZP1DS	-	SEARC	SAVED VALUE OF DP1DS/ TRYIT1
ZP2DS	-	SEARC	SAVED VALUE CF DP2DS/ TRYIT1

ZSTEP(I)
I=1,6
	-	SEARC	SAVED VALUES OF THE TRIAL STEP SIZES/ TRYIT1

ZYES(I)
I=1,5
	-	SEARC	SAVED VALUES OF YES(I)/ TRYIT1

VI. POST SUBROUTINE INDEX

SUBROUTINE NAME	OVERLAY	OCTAL SIZE
AET	(0,0)	74
ADDREL	(2,6)	116
AERO	(2,3)	252
AEROHI	(2,3)	155
AERO4	(2,3)	106
ANGLE2	(0,0)	115
ANMPT	(2,3)	263
ARTBR	(2,0)	71
ATANH	(0,0)	32
ATAN3	(0,0)	27
ATMOS	(2,0)	34
ATMOS1	(2,0)	43
ATMOS2	(2,0)	275
ATMOS3	(2,0)	507
AUXFM	(2,3)	614
AUXFMI	(2,2)	374
BACKOI	(2,0)	66
BACKOR	(2,0)	66
BLKDAT	(0,0)	7
BTW	(0,0)	105
BUCKET	(0,0)	123
CALE	(2,0)	120
CALES	(2,3)	26
CALSPE	(2,3)	21
CGM	(2,5)	43
CLGM	(2,3)	30
CLSPFL	(2,0)	114
COMBIN	(2,5)	56
CONIC	(2,3)	544
CONICT	(2,3)	547
CONVO	(2,3)	226
COSH	(0,0)	32
CUBMIN	(0,0)	131
CYCXM	(2,3)	220
CYCXMI	(2,2)	116
CYCYM	(2,3)	67
DATA	(2,0)	7
DELTU	(2,5)	45
DERIV	(2,3)	21
DERVI	(2,2)	15

POST SUBROUTINE INDEX (CONTD)

SUBROUTINE NAME	OVERLAY	OCTAL SIZE
DGAMLAM	(2,3)	166
DGM	(2,5)	230
DGMP2	(2,5)	36
DICT	(1,0)	7
DIFTAB	(2,0)	103
DIGDIF	(2,0)	62
DINPT	(2,1)	164
DISPLY	(0,0)	143/10
DLOOK	(2,2)	37
DPOLY3	(2,0)	31
DPRNG	(2,3)	150
DROP	(2,5)	313
DTAB	(2,0)	554
DTMDL	(2,3)	137
DVADDM	(2,2)	200
DWINDS	(2,3)	56
DYNS1	(2,3)	11
DYNS3	(2,3)	35
DYNXA	(2,2)	14
DYNXB	(2,2)	13
DYNXM	(2,3)	31
DYNXMI	(2,2)	15
DYSI1	(2,2)	50
DYSI2	(2,2)	251
EPHEM	(2,2)	256
ERROR	(0,0)	57
EXPN	(1,0)	43
FGAMA	(2,0)	163
FORMN	(2,2)	247
FITER	(2,0)	153
FOPC	(2,3)	306
FOPMIN	(0,0)	232
FXRNG2	(2,3)	233
GABDD	(2,5)	63
GAMLAM	(2,0)	121
GCNTRL	(2,3)	10
GENMIN	(2,0)	527
GENTAB	(2,0)	113
GMAG	(2,5)	41
GGUID	(2,3)	10
GNAV	(2,3)	10
GRAD	(2,0)	134
GRADS	(2,3)	71

POST SUBROUTINE INDEX (CONTD)

SUBROUTINE NAME	OVERLAY	OCTAL SIZE
GRAV	(2,3)	114
GSA	(2,3)	225
GSAI	(2,2)	16
GSENSR	(2,3)	53
GUIDI	(2,2)	523
GUIDX	(2,0)	212
GUID1	(2,0)	600
GUID2	(2,3)	524
HSWGT	(2,3)	47
IBMTRX	(2,0)	525
INFXM	(2,3)	302/333
INFXMI	(2,2)	177
INPUTX	(1,0)	656
INTGRL	(2,2)	37
INVM	(0,0)	547
IRMTRX	(2,0)	63
IRTBR	(2,0)	54
ITERO	(2,6)	1,250/1,545
LEVEL	(2,0)	20
MADD	(0,0)	35
MASTER	(0,0)	7,343/11,431
MATPY	(0,0)	77
MINMYS	(2,0)	125
MOMENT	(2,3)	167
MONITR	(2,3)	46
MOTENC	(2,3)	152
MOTIAL	(2,2)	1,400
MOTION	(2,3)	554/725
MTRXM	(0,0)	66
MTRXT	(0,0)	66
MTRXTV	(0,0)	55
MTRXV	(0,0)	54
NMLTER	(1,0)	22
NOMHOL	(1,0)	117
NOMINL	(2,0)	220
OLGM	(2,0)	10
OLGOM	(2,0)	15/35
ORBTR	(2,2)	241
ORBTRT	(2,2)	241
PAD	(2,0)	166

POST SUBROUTINE INDEX (CONTD)

SUBROUTINE NAME	OVERLAY	OCTAL SIZE
PAGER	(0,0)	55/104
PBLOCK	(2,3)	1,043
PGM	(2,5)	235
PHZXM	(2,3)	172
PHZXMI	(2,2)	45
POLY	(0,0)	46
POLY1	(2,0)	22
POLY2	(2,0)	26
POLY3	(2,0)	33
PPT	(0,0)	107
PRNTIC	(2,2)	4,627/4,726
PROP	(2,3)	614
QMULT	(2,2)	74
QRTATE	(2,2)	72
QUATI	(2,2)	60
QUAT2	(2,2)	26
QUAT3	(2,2)	52
READAT	(1,0)	3,112
REVDAT	(0,0)	7
REVISE	(2,5)	136
RGENDA	(1,0)	773
RRTBR	(2,0)	101
RSCORE	(1,0)	100
RSEARC	(1,0)	367
RTAB	(1,0)	74
RTBLML	(1,0)	1,015
RUK	(2,3)	147
SAVIC	(2,0)	101
SCORE	(1,0)	31
SDM	(2,5)	20
SERCH	(1,0)	42
SETESN	(2,0)	74
SETIC	(2,0)	140
SETIV	(2,0)	76
SHRINK	(1,0)	37
SINH	(0,0)	32
SOLVE	(2,0)	150
SP	(0,0)	30
SREL	(1,0)	33

POST SUBROUTINE INDEX (CONTD)

SUBROUTINE NAME	OVERLAY	OCTAL SIZE
SUN	(2,2)	263
SVDQ	(2,3)	2,067
SVDQI	(2,3)	147
TAB	(2,0)	510
TEST	(2,0)	154
TGOEM	(2,3)	376
TGOEMI	(2,2)	331
THPM	(0,0)	125
THPOSM	(0,0)	160
TMOTM	(2,3)	122
TPOSM	(0,0)	64
TRACKM	(2,3)	275
TRAJ	(2,0)	202
TRAJX	(2,0)	120
TRIM	(2,3)	217
TRITAB	(2,0)	350
TRYIT1	(2,0)	323
TRYIT2	(2,0)	160
TSPXM	(2,0)	45
TWOBDY	(2,3)	356
UNIT	(0,0)	46
UNITDU	(2,0)	40
UPDATS	(2,5)	164
UPNOM	(2,0)	22
UPNOMS	(2,3)	25
VCROSS	(0,0)	41
VDOT	(0,0)	30
WGTINI	(2,2)	265
WINDS	(2,0)	74
WUCAL	(2,5)	272
X1TER	(2,3)	201
XRNGE1	(2,3)	114
XRNGE2	(2,3)	217
ZERO	(0,0)	27

U. S. Government Printing Office: 1975--637-464/74 Region No. 3-II