N70-27151
CR-109777

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W., WASHINGTON, D.C. 20024

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE- INTAP - Interplanetary Trajectory Analysis Program

TM-70-1032-2

DATE-January 26, 1970

FILING CASE NO(S)- 105-3

AUTHOR(S)- R. W. Grutzner

FILING SUBJECT(S)- (ASSIGNED BY AUTHOR(S))- Trajectory Analysis
Interplanetary Mission Analysis

ABSTRACT

The computer program INTAP is intended as an aid in studying point-to-point, non-integrated, heliocentric conic section trajectories. It is not restricted to direct planet-to-planet flight but may include arbitrary points (defined by a radius vector and Julian date) in a sequence of planets to be linked by conic section trajectories. The individual conics are generated as solutions to Lambert's problem and a set of constraint equations may be imposed upon a sequence of such conics, the most common constraint being a ballistic flyby of one or more planets in the sequence. The trajectories are computed at points within a two dimensional parameter grid, the constraint equations having reduced the number of degrees of freedom of the system to two, each of which is associated with an axis of the grid. Contour plotting is then utilized to generate level surfaces of the dependent variables.

In addition to the trajectory characteristics, the information concerning near-planet encounters is computed using relative velocity vectors of approach and departure (hyperbolic excess velocities) to determine the flyby orbits within the planetary sphere of influence.

Abort trajectories, which return directly to Earth may be generated from any point(s) on a given heliocentric orbit.

SEE REVERSE SIDE FOR DISTRIBUTION LIST

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

SUBJECT: INTAP - Interplanetary Trajectory Analysis Program

DATE: January 26, 1970

FROM: R. W. Grutzner

TM-70-1032-2

TECHNICAL MEMORANDUM

1. INTRODUCTION

INTAP, an application of the Parametric Analysis Program(1) aids the systematic study of heliocentric point-to-point conic section trajectories. The system considered by INTAP consists of a sequence of "time points" (each of which is defined by a heliocentric position vector and an associated Julian date) and the conic section trajectories which link the time-wise adjacent points. Certain of the dependent variables (characteristics) of the system may be subject to constraint equations, the satisfaction of which defines a solution vector in the independent variable (parameter) space. If the proper number of characteristics are constrained, then the solutions, as well as any unconstrained characteristics, can be expressed as functions of two parameters. These two parameters may be associated with the coordinate axes of a grid, whose size is determined by the domains of interest of the two parameters and whose mesh size is determined by the degree of thoroughness with which these domains are to be studied. The non-grid solution parameters and the unconstrained characteristics are then representable by surfaces, the behavior of which may be analyzed by means of contour plotting.

As an example of the above description consider a function f_1 of three parameters:

$$f_1 = f_1(x_1,x_2,x_3) \tag{1}$$

which is constrained by an equation of the form:

$$f_1(x_1,x_2,x_3) - K_1 = 0 \tag{2}$$

If x_1 and x_2 are chosen as grid parameters, then at any grid point $(\bar{x}_1,\bar{x}_2)$, the solution of equation (2) yields:

$$x_3 = x_3(\bar{x}_1,\bar{x}_2) \tag{3}$$

Repeating this process at each grid point then generates points on the solution surface $x_3 = x_3(x_1,x_2)$. The variable x_3 is termed a search parameter. Any other function of the three parameters is then also representable as a function of the two grid parameters.

By obvious extension a function of n parameters subject to n-2 constraints may be represented as a function of any two of the parameters chosen to define the grid.

INTAP uses the above techniques in (generally) the following manner, although several variations are possible: given the grid and mesh sizes, and information concerning constraints and non-grid parameters, an initial solution to the constraint set is sought via a secant method iteration scheme. The initial solution is propagated over the entire grid, following which numerous functions of interest are computed and printed for each grid point. Plotting options are included for graphically displaying the characteristics as either contours or slices. Using this information, the feasibility of particular sets of trajectories may be assessed.

Several auxiliary programs exist for further study of the solutions. Among these are provisions for generating abort trajectories which define the requirements for effecting a direct return to Earth from points on a given heliocentric orbit. Other programs enable the connecting of trajectories generated within one grid to those of another grid, assuming a common grid parameter.

2. METHOD

As stated above, the basic quantity considered by INTAP is a "time-point", which consists of a heliocentric position vector and an associated Julian date. A conic section joining two time points is referred to as a "trajectory leg," and a sequence of legs will constitute a "segment". In the most frequently used application of this program, that of direct planet-to-planet flight, the four quantities which define a time-point are not independent but are related through Kepler's laws -- specification of a particular planet and Julian date uniquely determines the three coordinates of the planet. However, provision exists for specifying an arbitrary time-point (other than a planet), whose four coordinates are independent. This arbitrary point may then be treated as an element in the set of points to be linked by a sequence of heliocentric trajectories (segment). This option may be useful for applications wherein direct planet-to-planet flight is inefficient, e.g., transfers requiring high inclinations, thus necessitating a "plane change" maneuver enroute. This plane change is effected by a totally self-induced impulse (as distinguished from an alteration provided by the gravitational field of an encountered planet) which generally modifies both the magnitude and direction of the heliocentric velocity vector.

The computation of the trajectory leg between two time-wise adjacent points is achieved by means of an iterative routine which solves Lambert's problem, i.e., given two radius vectors (magnitudes and included angle) and the flight time between them find the unique* conic section (in terms of eccentricity and semi-traverse axis) which joins them. In terms of parameters and constraints, the two time points permit a maximum specification of eight parameters $(r_1, \lambda_1, \beta_1, t_1, r_2, \lambda_2, \beta_2, t_2)$.** If all eight of these parameters are set to some fixed values, the conic connecting the points is determined. If one parameter is free, then one constraint (either semi-transverse axis or eccentricity) may be imposed, and if two are free (excluding the combinations $(t_1, t_2), (\lambda_1, \lambda_2), (\beta_1, \beta_2), (\lambda_i, \beta_j, i=1,2; j=1,2)$), both orbit elements may be constrained. If the time point is a planet, then the Julian date is the logical choice to be the parameter since the three other variables (r, λ, β) are all single valued functions of the date. If two time-wise adjacent points are both planets, then the two position vectors are specified by the Julian dates and the Lambert leg joining the two planets is determined and hence neither the semi-transverse axis nor the eccentricity may be prescribed. However, there is another type of constraint which may be satisfied in this case and which is often of interest. Consider three consecutive planets. Let t_1 and t_3 be the grid parameters and t_2 be a search parameter which will be used in satisfying the ballistic flyby constraint

$$\Delta V_2(t_1, t_2, t_3) = 0 \tag{4}$$

an explanation of which follows: for a combination t_1, t_2, t_3 the two legs linking planet 1 with planet 2 and planet 2 with planet 3 may be computed. The velocity of planet 2 is also computed at t_2 as are the velocities on the conics inbound to and outbound from planet 2. If the velocity of planet 2 is subtracted (vectorially) from the other two velocities, the results are the relative velocities of approach and departure

---

*Assuming prograde transfers of less than 2π rad.

**λ = celestial longitude
β = celestial latitude.

from planet 2 or the "hyperbolic excess velocities" (v_∞). When the difference in the magnitudes of the two v_∞ vectors, ΔV_2, is zero, a continuous hyperbola may be passed from one to the other, the passage distance from the planet's center being determined by the mass of the planet, the magnitudes of and the angle between the v_∞ vectors. Thus, for a given t_1 and t_3, the non-grid parameter t_2 is used in searching for a configuration such that equation (4) is satisfied.

In more general terms then, given an initial guess at the state of a set of "time-points" and defining information as to which elements of the set are parameters and which are fixed, the simulation portion of INTAP will, based upon the particular functions for which constrained values are sought, generate those conics necessary to evaluate the functions and compute a vector of errors. This vector is fed back to the convergence routines for use in subsequent iterations. Assuming that the procedure converges to at least one solution of the constraint set, the solution will (at the user's option) be propagated to each point within the solution grid defined by the user.

As an illustration, consider a hypothetical situation in which it is desired to depart from Venus, pass Mars ballistically, fly to a time-point, three of whose coordinates (t, λ, β) are fixed, but whose heliocentric radius is a parameter, execute a plane change maneuver at the time-point, and then arrive at Earth on an ellipse with a fixed semi-major axis. The significant characteristics of such a segment might be Venus departure velocity, pericentron altitude at Mars passage, delta-V required for plane change, and Earth arrival velocity.

The problem could be defined in terms of parameters and constraints as follows:

<u>grid parameters</u>

a) Julian date at Venus departure
b) Julian date at Earth arrival

<u>search parameters</u>

c) Julian date at Mars passage
d) heliocentric radius at plane change

constraints

a) ballistic passage of Mars
b) fixed semi-major axis on third leg conic.

The user must have supplied an initial trial parameter vector and information concerning the ranges of the parameters. These data are entered through the various NAMELISTS which are considered in Appendix A.

At this point, program control will transfer to the search and convergence routines in an attempt to find an initial solution to the constraint set. Using the given parameters, the heliocentric position vectors of the four time-points are computed in the Ecliptic System.* Then, by means of an iterative routine, Lambert's problem is solved for the three unique conics which join the time-wise adjacent points. The heliocentric velocity vectors on the conics approaching and departing Mars are found and from these is subtracted the velocity of Mars on the given Julian date. As stated, the resulting relative velocity vectors, when equal in magnitude, define a ballistic flyby of the planet. Assuming they are unequal on the first trial, their difference is entered as an element of the error array.

The second element in this array is the difference between the desired semi-major axis on the third leg and that which resulted from solution of Lambert's problem on that leg. The search parameters are incremented and the process repeated. The resulting three error vectors provide a starting point for an iteration routine which will force the search parameters to a solution assuming:

a) such exists and
b) the search range includes points sufficiently close to said solution.

If this process does indeed converge, the resulting solution(s) may then be automatically propagated over the entire grid in a similar manner with the solution at one point used as a starting guess for adjacent points.

Having filled the grid, the characteristics may be computed, if desired. In this phase of the program, all the conics are generated, and for each point in the parameter grid the data pertaining to the conics and the time point encounters (flybys, plane changes, etc.) can be printed. Coincident with the printing, save files may be written to retain the data for future use, e.g., plotting of certain of the characteristics. Plotting may also be done within the present execution.

---

*See Figure 1.

Had there been another set of trajectories arriving at Venus over the same range of dates as the Venus departure range just considered, these two segments could be connected by either an impulsive or stopover encounter and the total set of trajectory characteristics for the two-segment flight plotted and printed.

An impulsive encounter is generated when the last planet in one segment is the same as the first planet in the next segment and the times of arrival at this planet in the first segment match the departure times for the second segment. In this case the inbound and outbound v_∞ vectors, although unequal in magnitude, define a plane in the planetocentric coordinate system and the minimum single impulse transfer from one asymptote to the other may be computed in this plane. If the arrival times in the first segment differ from the second segment departure times by a fixed amount, the segments may be connected by a deboost-boost stopover wherein the spacecraft approaches the planet hyperbolically, deboosts at pericentron into an elliptical parking orbit for the required stay time and then boosts back out onto the outbound hyperbola. These features will be clarified by means of specific examples in Section 4.

Within the present framework, the program is capable of dealing with the following parameters if the time-point is a planet:

a) Julian date
b) radius of a point relative to planet's center
c) latitude of a point defined in the local planetocentric coordinate system
d) longitude of a point defined in the local planetocentric coordinate system.

If the time-point is other than a planet, the following may be considered parameters:

a) Julian date
b) heliocentric radius
c) celestial latitude
d) celestial longitude.

The constraints which may be satisfied are:

a) ballistic flyby of any body in heliocentric orbit
b) fixed eccentricity of any leg when at most one end-point is a planet

c) fixed semi-major axis of any leg when at most one end-point is a planet.

3. USAGE

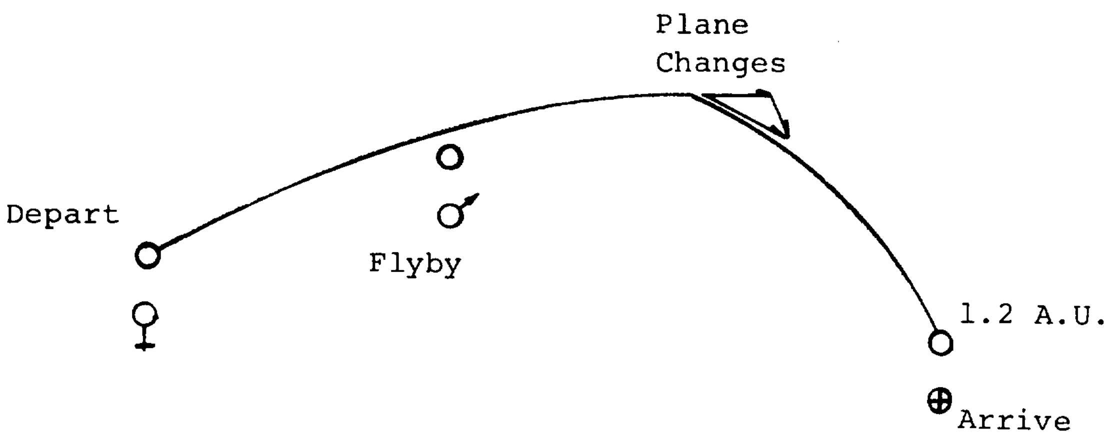
The program is structured into five subdivisions, and a run may invoke any logical subset or all five of the subdivisions. If a run is to start anywhere but at the initial subdivision, previously generated data must be made available to the program through either tape or saved FASTRAND files. Brief definitions of the primary functions of the five subdivisions follow:

1.	INISOL	seeks an initial solution to the constraints for specified values of the grid parameters
2.	FILGRD	propagates the initial solution to each point within the grid. If convergence cannot be achieved at a particular point, that point is marked to indicate this fact and the program continues.
3.	COMCHA	computes and prints characteristics of the trajectories at each grid point.
4.	AUXPRO	combines segments at common dates, computes and prints impulsive flyby or stopover characteristics. Generates aborts.
5.	PLTCON	generates plots of any of the characteristics.

The inputs necessary to run INTAP are discussed in the User's Guide (Appendix A).

4. TEST CASES

Sample outputs from two test cases are included. The first case, that which was discussed in Section 2, is a single segment incorporating a plane change maneuver, a ballistic flyby, and a semi-major axis constraint on one of the Lambert legs. The following figure conceptually displays the sequence:



The grid parameters were defined to be the times of departure from Venus and arrival at Earth. Two other parameters were the Mars passage date and the radial distance from the sun at the time-point. The constraints were a ballistic passage of Mars and a semi-major axis of 1.2 A.U. on the inbound leg to Earth. A list of the inputs and samples of the output for both the extensive and condensed printout options can be found in Appendix F. Figures 2 and 3 are contours of Mars passage radius and plane-change delta-v.

The second sample case illustrates the manner in which various segments are combined to form complete trips. A segment was generated which consisted of Earth departure, (grid Parameter 1), ballistic flyby of Mercury, and then Venus arrival (grid parameter 2). It can be seen (Appendix G) that a ballistic flyby of Mercury would pass well beneath the planet's surface, obviously precluding the use of this particular set of trajectories but they are intended here only as an illustration of certain features of the program. The Venus arrival dates covered the same span as did the Venus departures in the first example, thereby enabling the connecting of segments via an impulsive flyby at common grid points, i.e., for each Venus arrival (departure) there are ND1xNA2 complete trajectories consisting of:

a) Earth Departure
b) Mercury Flyby
c) Venus Impulsive Transfer
d) Mars Flyby
e) Plane Change
f) Earth Arrival

where ND1 is the number of Earth departures in the first segment and NA2 the number of Earth arrivals in the second segment. Appendix G contains some sample output of the connected segments, and Figure 4 is a contour plot of minimum transfer velocity holding the Venus impulsive transfer time fixed at its minimum value (JD=43286), the grid parameters being Earth departure and arrival times. The figure indicates that the transfer velocity is relatively independent of the Earth arrival time in the second segment.

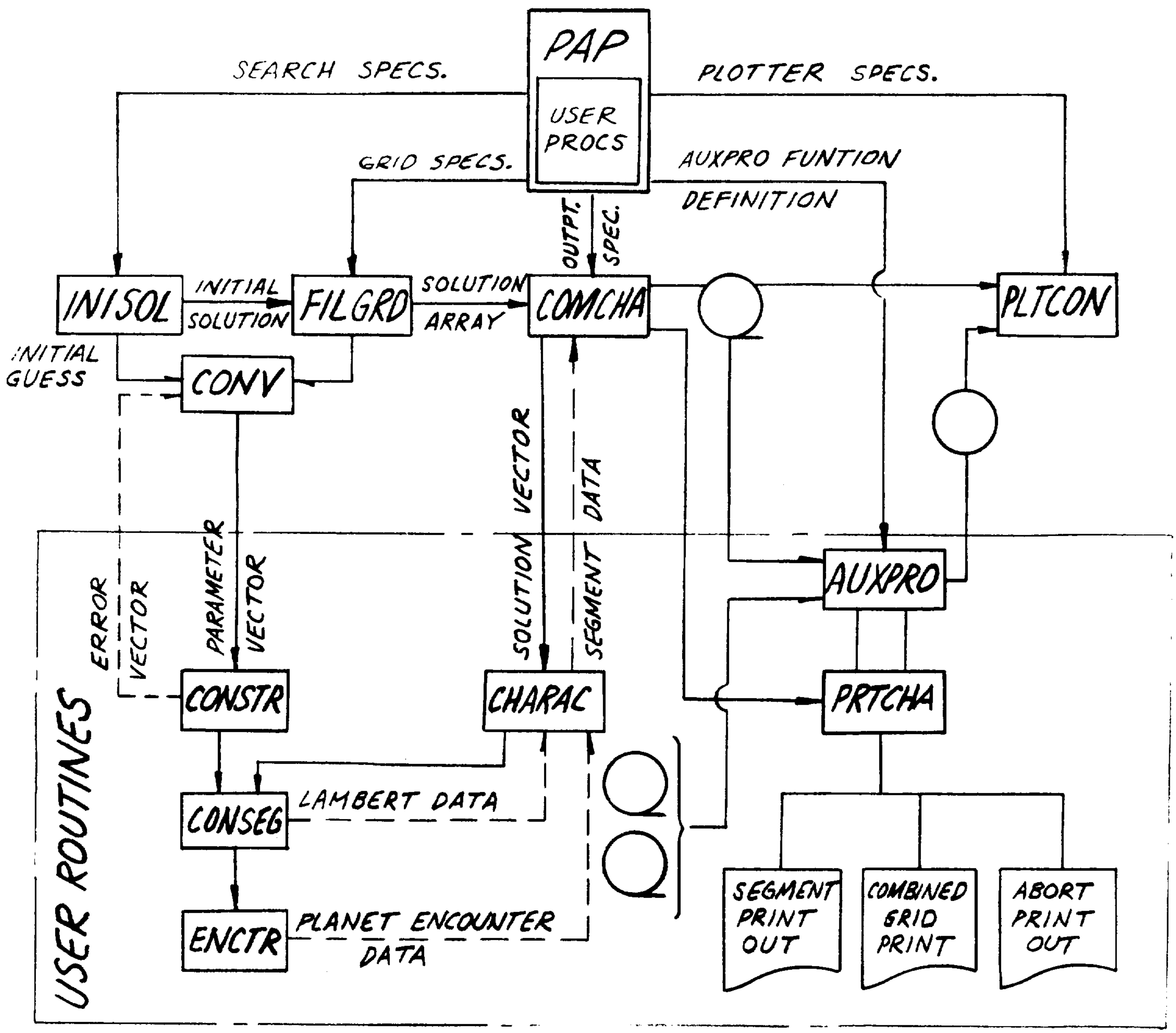
The run which produced this second case, i.e., generate the two grids, combine them and plot contours required a computer charge of about 65 units.

5. PROGRAM DESCRIPTION

An attempt has been made to impart a modular structure to the program. In anticipation of future applications, provisions have been made for the inclusion of additional operations with a minimum of programming effort. Scattered throughout the program there are 'unassigned options' which are presently not used but which provide space for the insertion of new code as the need arises. The logic is set up to handle an optimization feature although none such exists presently. There are three specific types of constraint errors that may be evaluated within the current framework but, again, the user may expand upon this by inserting either a line of code or by adding additional entry points to an existing routine, depending upon the type of constraint he seeks to satisfy. In either event, the logical structure has already been included.

5.1 Subroutine Interface

Appendices C and D cross-reference the subroutines and common sections within INTAP (exclusive of those provided by PAP). The following functional block diagram displays the hierachical structure of the major subroutines:



5.2 Subroutine Operations

Referring to the previous figure it is seen that the significant input to the user routines is a parameter vector. From the first two PAP functions, INISOL and FILGRD, this vector is either an initial or an iterated guess at the solution while from the third function, COMCHA the vector is a solution. This vector is transmitted as an argument by either subroutine CONSTR or CHARAC to subroutine SETUP. The output from SETUP is a dimensioned variable containing the components of the heliocentric position vectors for each of the time-points in the segment. The position vectors are computed directly from (r,λ,β) if the time point is other than a planet. In the case of a planet, given the Julian date, the program sequentially determines the mean anomaly, eccentric anomaly, true anomaly whence the radius which is decomposed along the periapse and semi-latus rectum directions. The Euler angles of the planet's orbit determine the elements of a rotation matrix to transform the vector to the Ecliptic system. The planet numbers and encounter dates are also stored in COMMON for each of the points. Those time-points with no associated parameter will have been stored in the pass through PAP (wherein the simulator is defined), and as they are invariant, need be computed only once. The program logic branches at the return from SETUP to reduce the amount of computation required. CONSTR checks each pair of adjacent points and determines if a constraint must be satisfied at either of the points or on the Lambert leg connecting them. If so, CONSEG is called to join the points; otherwise the next pair is checked. This procedure is continued until all pairs of adjacent points have been considered. The constrained functions are then evaluated, an error vector generated, and control returned to the convergence routines. In CHARAC, all the points are connected since the parameter vector represents a solution. Using the relative velocity vectors as input, ENCTR is called to compute the near planet trajectory data; this data and the Lambert data are stored in another vector, termed the "characteristic vector," which is subsequently printed by subroutine PRTCHA.

5.3 Subroutine Description

This section briefly describes the function of each subroutine (exclusive of those contained within PAP). Explanatory discussion is included where appropriate. A cross reference of the routines is to be found in Appendix C.

AUXPRO	controlling routine for calling AUX1 and AUX2
AUX1	combines grids and generates plot files
AUX2	generates abort trajectories
CHARAC	controlling routine for computing Lambert legs and encounter data.
CONSEG	computes Lambert leg data and relative velocity vectors.

CONST sets all constants (including planet data) used by the program.

CONSTR evaluates present state of functions to be constrained and generates an error vector.

ECANOM iteratively solves Kepler's Equation for the eccentric anomaly.

ENCONT computational routine for generating planetary encounter data. Since the various encounter types require different outputs, the subroutine utilizes the FORTRAN V entry point feature such that for each encounter type ENCONT is called several times, entering the program at various points. The entries are defined as follows:

ENTRY1 - stores the relative velocity vectors in COMMON and computes the sub-solar point coordinates and the bend angle.

ENTRY2 - computes the spatial characteristics of the flyby trace.

ENTRY3 - computes the arrival, departure, and deboost velocity requirements.

ENTRY4 - computes minimum single impulse hyperbolic transfers.

ENTRY5 - computes additional bend angle information.

ENTRY6 - computes characteristics unique to a plane change maneuver.

The reason for using this modular approach is that the user, rather than having to write a computational routine for any new type encounter, e.g., an optimum two impulse transfer around a planet, can set up a sequence of calls to ENCONT using the various entry points and a good deal of the computational outputs will already have been provided. He need only supply the code for generating the double impulse. The COMMON section associated with the encounter data, PENCTR, has several presently unused variables which have been included in anticipation of future expansion.

ENCTR	controlling routine for calling ENCONT.
EQRTR	computes the location of a transfer from one asymptote to a second such that the resulting hyperbolii have equal periapse radii.
FIXPLC	computes the components, in the Ecliptic system, of the radius vector from the sun to a planet's center, given the Julian Date.
FIXREL	converts a relative position vector from a planetocentric system to the Ecliptic system.
LAMBRE	iterative routine which solves Lambert's problem.
LODCHA	in much the same way as ENCONT is structured, subroutine LODCHA is set up to provide a modular capability that permits the user to place those encounter outputs he requires into the 'characteristic vector.' The entries into this routine load those elements of COMMON PENCTR which the user wishes to output, into the vector CHA in the order he prescribes. If a new encounter type is desired the user will provide the code for an additional entry point and must of course modify the output routine (PRTCHA) to print the newly created characteristic vector. The 'unassigned option' within ENCTR and LODCHA permit inclusion of new code with a minimum of concern regarding the logical structure of the routines.
MANOM	computes the mean anomaly given the Julian Date.
MATELM	computes elements of transformation matrices.
OPTRAN	computes the location of the minimum Δv single impulsive transfer between asymptotic velocity vectors.
ORBIT	given heliocentric position and velocity vectors, computes orbital parameters.

PAP	main program.
PERXFR	computes location of a periapse transfer between asymptotes.
PRTCHA	output routine.
RADMAG	computes heliocentric radius components along perihelion and semi-latus rectum directions.
SETUP	transforms the parameter vector to a series of heliocentric position vectors
SLOPE	computes the flight path angle.
TRUANM	converts eccentric anomaly to true anomaly.
VASTOT	computes the virtual asymptote for arrivals and departures, given the desired periapse declination.
VELMAG	computes velocity components in the orbit plane.

6. Auxiliary Routines

At present there are three functions which may be performed by auxiliary routine AUXPRO. The first of these connects segments which have a common grid parameter. The second function involves generation of plot files and the third computes abort trajectories. Regarding the first two functions, consider the situation with two independent segments which were generated such that the arrival time at the final point of the earlier segment and the departure time from the initial point of the second segment were both grid parameters (sample case 2). It is then possible to connect these two segments by means of an impulsive flyby if the two aforementioned times correspond. If the two grids were separated in time by a fixed amount, then a deboost stopover could be used in combining them. AUXPRO performs this function by reading the 'characteristic files' generated by either two or three successive passes through COMCHA. In operating on the data files, AUXPRO will also generate a plot file if so requested, although, since the plot information can be a function of only two variables, the user must fix the value of one or two (depending on whether 2 or 3 grids are being connected) of the three or four independent grid variables associated with the entire trip. The input specifications to AUXPRO are listed in Appendix A. If, when connecting the segments, ISAA is set equal to one, a save file will be generated which contains all the characteristics of the trip for each point in the combined grids. This file may then be processed to produce a plot file wherein any two of the grid parameters on the original files may be considered the total grid parameters and the values of the remaining parameters set at fixed values the user chooses.

The third function, abort trajectory generation requires an initial pass through INISOL and COMCHA to compute the characteristics of the trajectory from which aborts are to be made. Having generated this reference "orbit" the input specifications defined in NAMELIST $ABORT then enable the calculation of abort trajectories.

7. OUTPUT ROUTINES

The output resulting from the convergence routines is defined in Reference 2. The output which describes the solution characteristics takes either of two forms. The first is a condensed printout which, it is hoped, will give the user sufficient information to assess the feasibility of a given solution grid without requiring an undue amount of printed output. The more complete output lists all the Lambert leg data and the encounter characteristics defined in Appendix B. The encounter data are printed sequentially whether the output represents only a single segment or is the result of combining segments via AUXPRO.

8. ACKNOWLEDGMENTS

The author is indebted to P. F. Long for the descriptions of several of the input specifications, and for much of the problem definition.

R. W. Grutzner

1032-RWG-dmu

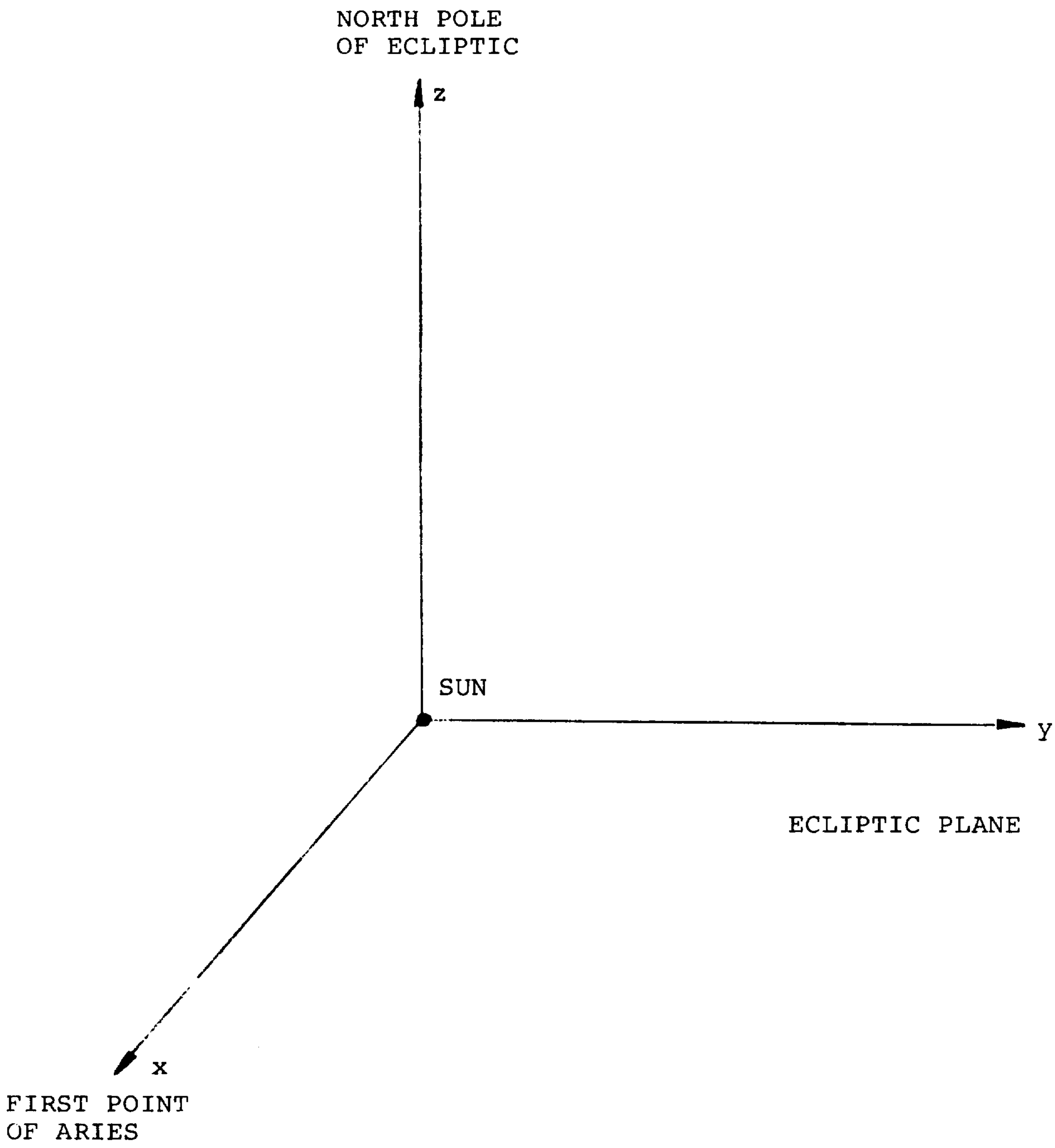
R. W. Grutzner

REFERENCES

1. Long, P. F., "A Method of Parametric Analysis," Memorandum for File, Case 720, April 16, 1968.

2. Long, P. F., "PAP - Parametric Analysis Program," (in preparation).

3. NASA SP-35, Planetary Flight Handbook, Volume III, Part I.



INERTIAL COORDINATE SYSTEM

FIGURE 1

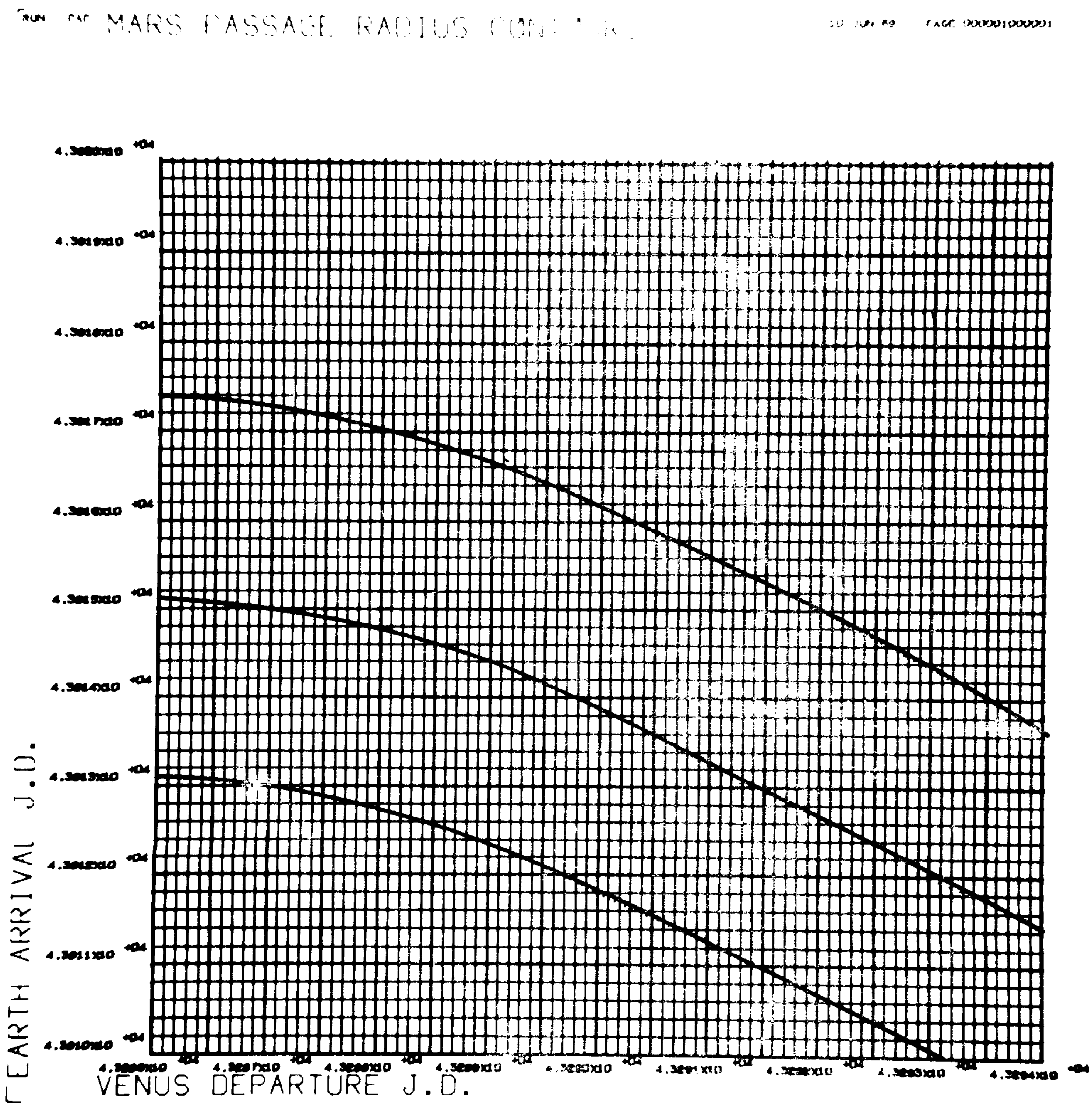


FIGURE 2

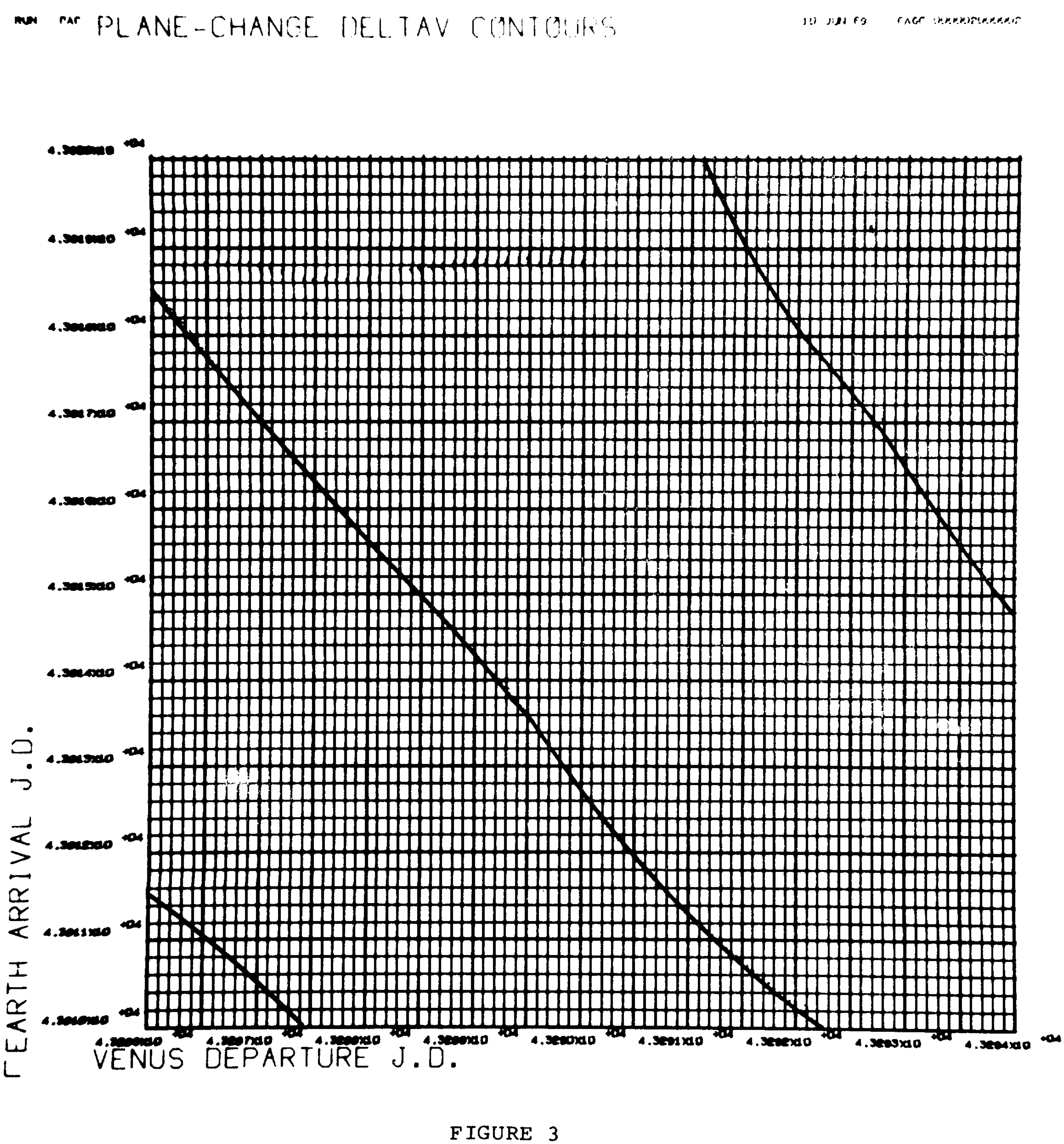

PLANE-CHANGE DELTAV CONTOURS
EARTH ARRIVAL J.D.
VENUS DEPARTURE J.D.

FIGURE 3

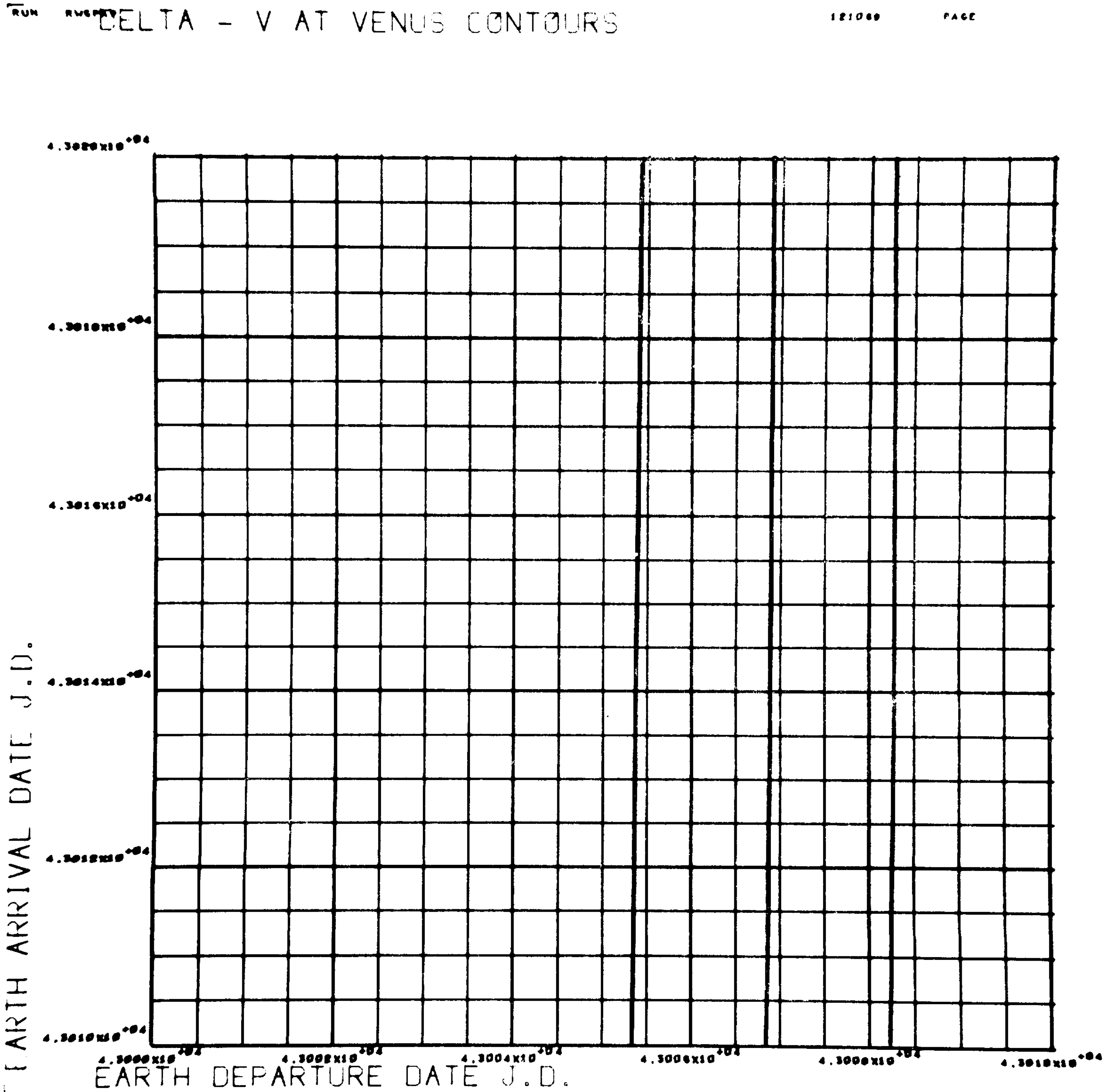

RUN
DELTA - V AT VENUS CONTOURS
121069
PAGE
4.3020x10+04
4.3018x10+04
4.3016x10+04
4.3014x10+04
4.3012x10+04
4.3010x10+04
EARTH ARRIVAL DATE J.D.
4.3000x10+04
4.3002x10+04
4.3004x10+04
4.3006x10+04
4.3008x10+04
4.3010x10+04
EARTH DEPARTURE DATE J.D.

FIGURE 4

APPENDIX A

USER'S GUIDE

This outline defines the various NAMELISTs and the input variables which must be specified by the user in order to run INTAP.

Input Lists

Data are entered through five separate FORTRAN NAMELISTs which perform functions as follows:

$RUNDAT	reads logical inputs unique to a particular approach to the machine.
$SAVDAT	reads inputs which define the system under consideration.
$PLANDT	reads additional planetary or asteroid data if desired. Data for all nine planets are included within the program.
$CONECT	reads inputs relevant to the combining of grids.
$PLTDAT	reads inputs defining the plotting operations.
$ABORT	reads inputs defining abort trajectories.

$RUNDAT

Name	Type/Size	Description
KON	INT/8	Control sequence indicator. KON(J)=I $(1 \le I \le 5)$ causes subroutine I to be called Jth in the sequence. The subroutines are defined in Section 3. KON(J)=6 returns control to the beginning of the program. KON(J)$\ge$7 causes normal termination of the execution.
IUNIT	INT/	Initial data unit. Defines the logical unit from which SAVDAT and the appropriate previously generated data are read if the program is to be started at a point other than INISOL, i.e., if KON(1)>1.

Name	Type/Size	Description
ICHSA	INT/	Data change indicator. If ICHSA=1 and KON(1)>1, namelist SAVDAT is read from logical unit 5 after it is read from logical unit IUNIT. If ICHSA≠1 and KON(1)>1, SAVDAT is read from IUNIT only. When KON(1)=1, ICHSA has no effect.
NFR	INT/	Number of frames to be plotted. Causes namelist PLTDAT to be read NFR times.
NF14 NF15 NF16 NF17	INT/	Plot data file flags. Plot data is to be read from the logical unit indicated by the last two symbols of the variable name when that variable is set to 1. If only one logical unit is being used, these variables may be ignored.
ISAI ISAF ISAC ISAA	INT/	Data set save flags. These flags, when set to 1, cause the data set of the subroutine whose first initial is the same as the last letter of the variable name to be saved.
IPRI	INT/	INISOL print flag. The solutions found are always printed. If IPRI=1, the search progress is also printed.
IPRF	INT/	FILGRD print flag. A display of the grid points for which solutions were found and the percentage of points converging is always printed. If IPRF=1, solution mapping progress is printed also.
IPRC	INT/	COMCHA print flag. =0, no results printed. =K, (K≥1) subrouting PRTCHA is called at the corner points of the grid and at points of the grid defined by the intersections of the Kth, 2Kth, 3Kth,...,grid lines.
IPRA	INT/	Not used.

Name	Type/Size	Description
IPRP	INT/	PLTCON print flag. ≤0, no printing. ≥1, frame summary table, drum storage information, and function arrays printed. ≥2, function value and availability arrays printed. >3, point selection map printed by interpolation subroutine.
IPRINT	INT/	Controls type of COMCHA and AUXPRO output. If IPRINT=0, condensed print is generated. If IPRINT>0, an expanded printout is produced.
NAUX	INT/	If NAUX=1, AUXPRO will be used for combining segments or file manipulation. If NAUX=2 abort trajectories are to be generated.
IREAD	INT/	If IREAD=1, program expects $PLANDT namelist to immediately follow the $SAVDAT list. $PLANDT permits modification of or addition to existing planetary data. Trips to the asteroids or a fictitious planet may be generated by reading the appropriate data which is defined in $PLANDT. The existing data for the nine planets are taken from Reference 3.

$SAVDAT (Items marked '*' are input)

Name	Type/Size	
*ARATS(I)	REAL/30	Ratio of semi-major axis of parking orbit to planet radius at Ith encounter. Only applicable to encounter types 1,6,8. (See IENC below).
*DELSTP(I)	REAL/30	Stopover duration at Ith time-point within segment. Applicable to encounter type 6 only. When connecting segments in AUXPRO, this time is inferred from data on files and need not be specified.
*EPS(I)	REAL/8	Maximum acceptable difference between Ith constrained function and its target value.
*ICCH(I)	INT/5	If the Ith set of characteristics is to be computed from the grid mapped from the Jth solution set ICCH(I)=J; note that the Jth solution must have been mapped.
*ICM	INT/	Convergence method selector (see Reference 2 for descriptions of methods).
*ICS(I)	INT/8	ICS(I)=jkm, a three digit constraint identifier with following interpretation for Ith constraint: j=0; constraint is at an encounter j=1; constraint is on Lambert leg; k; time-point number (j=0) or Lambert leg number (j=1) with which constraint is associated; m; type of constraint where m=1 → ΔV constraint m=2 → semi-major axis constraint m=3 → eccentricity constraint
*IENC(I)	INT/30	IENC(I)=j, defines the Ith encounter as type j where; j=1 → departure j=2 → ballistic flyby j=3 → impulsive flyby j=6 → deboost stopover j=7 → plane change j=8 → arrival

Name	Type/Size	Description
*IP1	INT/	If IP1=ij, the first grid parameter is from time-point i and is type j where; j=2 → Julian date j=3 → radius j=4 → longitude or right ascension j=5 → latitude or declination
*IP2	INT/	Same as above for second grid parameter.
*ISGR(I)	INT/5	If ISGR(I)=j, use Jth solution to map Ith grid.
*ISM	INT/	Search method selector. See Reference 2 for descriptions of methods.
*NCG	INT/	Number of grids for which characteristics are to be developed.
*NCS	INT/	Number of equality constraints to be satisfied.
*NSG	INT/	Number of initial solutions to be mapped.
*PERDEC(I)	REAL/30	Desired periapse declination (deg) at Ith time-point. Specified only for departures and arrivals. If desired value is not realizable, program will select its own value.
*PSGCON(I)	REAL/30	Passage radius or altitude at Ith encounter. If PSGCON(I)<10, value is interpreted as PASSAGE RADIUS/ PLANET RADIUS; otherwise it is assumed to be altitude in N.M; disregarded if specified for ballistic flyby or plane change encounter types.
*P1I	REAL/	The fixed value of the first grid parameter while the parameter space is being searched for initial solutions.

Name	Type/Size	Description
*P2I	REAL/	Same as above for second grid parameter.
*TARV(I)	REAL/8	Target value for Ith constraint.
*TPD	REAL/ (13,10)	Time-point definitions. See Appendix A1 for the variable definitions within TPD.

The following $SAVDAT variables are computed internally and constitute additional information which is written on all files to be saved.

LEGS	INT/	Number of Lambert legs.
LENGTH	INT/30	Length of a total characteristic record
LNENCR	INT/30	Length of an encounter record.
LNLBRT	INT/30	Length of a Lambert leg record.
NCUSR	INT/	Total number of characteristics.
NPONTS	INT/	Total number of time-points in segment.
NPUSR	INT/	Number of parameters.

$PLANDT

AMU(I)	REAL/20	Gravitational constant of Ith planet (ft**3/sec**2). I is ordered with increasing distance from sun, e.g., 1=Mercury, 2=Venus, etc.
RHO(I)	REAL/20	Radius of Ith planet (N.M.)
SMA(I)	REAL/20	Semi-major axis of Ith planet's orbit (A.U.)
ECC(I)	REAL/20	Eccentricity of Ith planet's orbit.
TPHPAS(I)	REAL/20	Time of perihelion passage (JD-2400000).
FINC(I)	REAL/20	Inclination (deg) of Ith planet's orbit to the Ecliptic.

Name	Type/Size	Description
OMEGA(I)	REAL/20	Celestial longitude (deg) of Ith planet's ascending node through Ecliptic.
PERMIN(I)	REAL/20	Angle (deg) from Ith planet's ascending node to its perihelion.
VEFRPH(I)	REAL/20	True anomaly (deg) of Ith planet's Vernal Equinox.
EQATOR(I)	REAL/20	Inclination (deg) of Ith planet's equator to its orbit plane.
$CONECT		
IGV1	INT/	If IGV1=jk, one parameter of combined grid is from segment j and is departure time (k=1) or arrival time (k=2).
IGV2	INT/	Same as above for second grid parameter.
KONECT	INT/	If KONECT=ijklmn, connect grid developed from solution i (segment j) to grid developed from solution k (segment l) to grid developed from solution m (segment n). If only two segments are to be connected, set KONECT=ijkl with same interpretations.
LIA1		Logical input unit for first segment data.
LIA2	INT/	Logical input unit for second segment data.
LIA3		Logical input unit for third segment data.
TFIXED(I) I=1, IMAX	REAL/2	Values of fixed times to be used in generating multisegment plots. IMAX is either 1 or 2 according as there are 2 or 3 grids. If there are 3 grids, TFIXED(1)<TFIXED(2). If a stopover is included between grids and is not a grid parameter for the combined grid, the arrival time (as opposed to departure) at the stopover is to be specified as TFIXED(I).

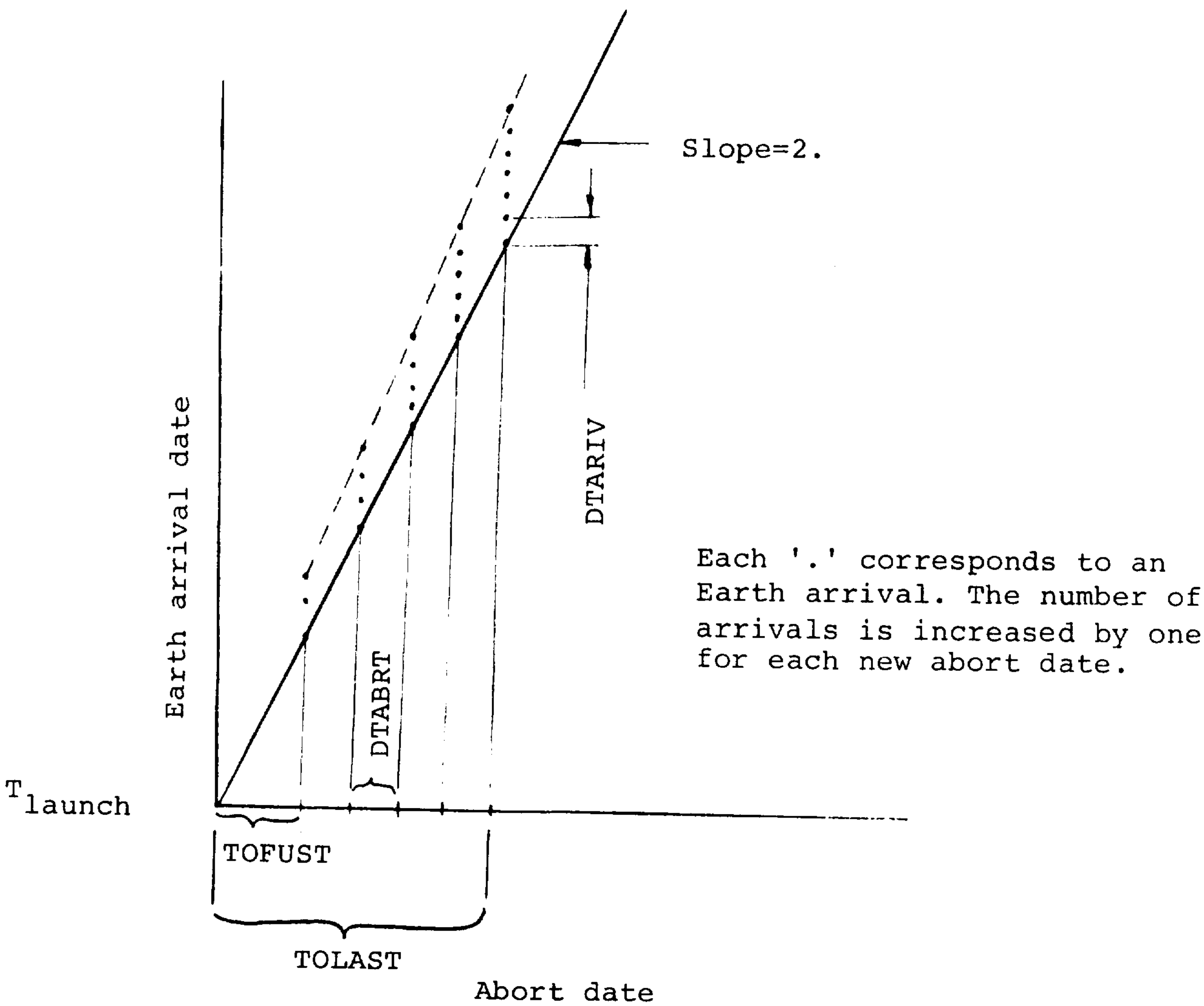
$PLTDAT

Name	Type/Size	Description
IOS	INT/	Plotting option selector. =0, off-line plots only. =1, printer plots only. =2, both types.
FTLT	HOL/8	48 character frame title.
ALAB	HOL/8	48 character abscissa label.
OLAB	HOL/8	48 character ordinate label.
LABC	INT/	Not used at present.
AMI	REA/	Minimum value of the frame abscissa.
AMA	REA/	Maximum value of the frame abscissa.
OMI	REA/	Minimum value of the frame ordinate.
OMA	REA/	Maximum value of the frame abscissa.
NFU	INT/	Number of functions on this frame (≤ 10).
IUN*	INT/10	Logical unit indicators. IUN(J)=I means that the Jth function of this frame is to be read from logical unit I.
ISO	INT/10	Solution numbers. ISO(J)=I means that the Jth function of this frame is from the Ith solution.
IVA	INT/10	Variable numbers. IVA(J)=I means that the Jth function of this frame is the Ith dependent variable of the solution. (See Appendix A2).
P1IN	REA/10	Interpolation increments for the first independent variable.

*The remaining variables in this table require a value corresponding to each function to be plotted on the present frame.

Name	Type/Size	Description
P2IN	REA/10	Interpolation increments for the second independent variable.
ICD	INT/10	Contour plane indicators. =0, constant function contours. =1, first variable constant contours. =2, second variable constant contours.
IAOS	INT/10	Abscissa-ordinate selectors. =0, P(1)/P(2),P(2)/F, or P(1)/F is the abscissa/ordinate. =1, the reverse is true.
CMI	REA/10	Minimum contour values (used when ICD=0).
CMA	REA/10	Maximum contour values (used when ICD=0).
CIN	REA/10	Step size between contours (used when ICD=0).
INS	INT/10	Indices of the initial contours (used when ICD=1 or 2).
LAS	INT/10	Indices of the final contours (used when ICD=1 or 2).
ISI	INT/10	Increments of indices (used when ICD=1 or 2).
$ABORT		
TOFUST	REAL/	Time, with respect to launch, at which aborts are to begin (J.D.).
TOLAST	REAL/	Time, with respect to launch, at which aborts are to end (J.D.).
DTABRT	REAL/	Increment in time of abort.
DTARIV	REAL/	Increment in earth arrival time.
NARIV	INT/	Number of earth arrivals corresponding to earliest abort time.

The following diagram serves to define the above inputs;



APPENDIX A1

DEFINITIONS OF INPUT VARIABLES TPD(13,10)

A segment may contain up to ten time-points and for each such point the following data must be entered:

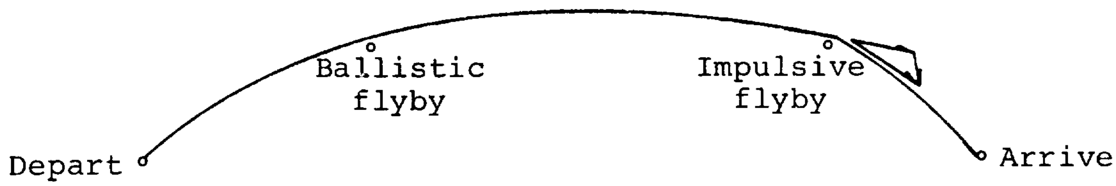
TPD(1,I)	0. for non-planetary time-point. a. where a is ath planet from Sun, e.g., a=1. Mercury, a=2. for Venus, etc.
TPD(2,I)	minimum Julian Date (JD-2400000.) associated with time-point I.
TPD(3,I)	maximum Julian Date (JD-2400000.) associated with time-point I.
TPD(4,I)	increment in Julian Date associated with time-point I. If TPD(4,I)=0 (in which case TPD(2,I)=TPD(3,I)), the time associated with time-point I is considered fixed. If TPD(4,I)≠ 0 and this time has been designated as a grid parameter via IP1 or IP2 then TPD(4,I) defines one increment in the solution grid. If TPD(4,I)≠ 0 and this time has not been designated a grid parameter then TPD(4,I) defines the search increment.
TPD(5,I)	minimum radius (A.U.) associated with time-point I. If I is a planet, this value is read in as the number of planet radii and refers to an incremental radius which is added to the heliocentric vector to the planet center.
TPD(6,I)	maximum radius (A.U.) associated with time-point I. If I is a planet remark concerning TPD(5,I) is pertinent.
TPD(7,I)	increment in radius. Interpretation is same as given for TPD(4,I).
TPD(8,I)	minimum celestial longitude (deg) associated with time-point I. If I is a planet TPD(8,I) may be specified as right ascension in the local planetocentric coordinate system.

TPD(9,I)	maximum celestial longitude (deg) associated with time-point I. If I is a planet, remark concerning TPD(8,I) is pertinent.
TPD(10,I)	increment in celestial longitude (deg). Interpretation is same as given for TPD(4,I).
TPD(11,I)	minimum celestial latitude (deg) associated with time-point I. If I is a planet TPD(11,I) may be specified as declination in local planetocentric coordinate system.
TPD(12,I)	maximum celestial latitude (deg) associated with time-point I. If I is a planet, remark concerning TPD(11,I) is pertinent.
TPD(13,I)	increment in celestial latitude (deg). Interpretation is same as given for TPD(4,I).

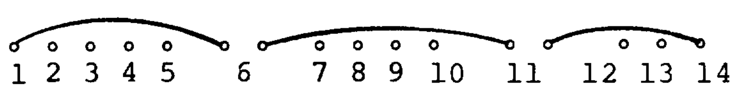
APPENDIX A2

VARIABLE ASSIGNMENTS WITHIN DATA RECORDS

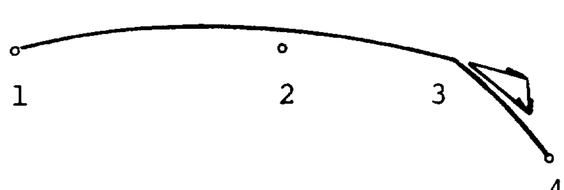
If any of the characteristics are to be plotted, the variable IVA(J) must be set in the $PLTDAT namelist. The means of so doing is best illustrated by an example. Assume the following sequence of encounters:



If the passage radius at the second time-point were to be plotted, IVA(J)=220 where the first '2' indicates the second time-point and the 20 indicates the twentieth variable in the encounter record for a ballistic flyby. Appendix B lists the locations within the data records of the various trajectory characteristics. As another case, for plotting the departure velocity from the third point, set IVA(J)=307. If plots are to be generated from multisegment trajectories, the value of IVA should be preceded by a 1(one). For example, if three segments were connected which had the following configuration:



and departure velocity from point 11 (assume an impulsive flyby encounter), were to be plotted, set IVA(J)=11107. If the same variable were to be plotted for point 6, set IVA(J)=10607. If non-grid parameters are to be plotted, the parameter number should be preceded by a minus sign. For example, in the following configuration point 3 might be a time-point whose heliocentric radius is the parameter to be plotted:



the times associated with points 1 and 4 might be grid parameters and a ballistic flyby is to be found at point 2, so that for each grid point the record would appear:

ICONV	P1	P2	P3	P4	C1	C2	C3 - - - - - - - - -

if radius at point 3 had been defined as the fourth parameter set IVA(J)=-4.

The value of IVA read in via the data deck is decomposed by PAP and then reconstructed to determine the location, within the characteristic record, of the particular variable to be plotted. This is done as follows: PAP, based upon the variable IENC associated with each time-point, assigns to that point a variable 'LENGTH' which specifies the number of words written for the particular encounter type plus the number written for the Lambert leg elements defining the conic connecting the present point with the next point in the segment. Then if IVA(J)=nlm, the value of IVA(J) stored internally is,

$$\sum_{k=1}^{n-1} \text{LENGTH}(k)+\text{lm}+\#\text{parameters}-2$$

APPENDIX B

VARIABLE ASSIGNMENTS WITHIN ENCOUNTER RECORDS

Departure, Arrival, Stopover Encounter types (IENC=1,8,6).

Var. No.	Name	Definition
1	NPLNO	planet number
2	TIMEIN	arrival time at encounter
3	VIN	magnitude of v_∞ inbound
4	RASIN	right ascension of v_∞ inbound
5	DECLIN	declination of v_∞ inbound
6	TIMOUT	departure time from encounter
7	VOUT	magnitude of v_∞ outbound
8	RASOUT	right ascension of v_∞ outbound
9	DECOUT	declination of v_∞ outbound
10	BA	bend angle
11	RASCSP	right ascension of subsolar point
12	DECLSP	declination of subsolar point
13	DELPSI	excess or deficiency in natural bend angle
14	PSI1	natural bend of inbound hyperbola
15	PSI2	natural bend of outbound hyperbola
16	PLINC	inclination of flyby trace to planetary equator
17	POMEGA	longitude of ascending node of flyby trace measured with respect to planet's vernal equinox.

Var. No.	Name	Definition
18	OMEGA1	angle between ascending node and inbound asymptote
19	OMEGA2	angle between ascending node and outbound asymptote
20	OMEGAP	angle between ascending node and periapse
21	ALPHP	right ascension of periapse
22	DECLP	declination of periapse
23	PALT	altitude of periapse
24	PRAD	radius of periapse
25	ARATIO	ratio of semi major axis of deboost ellipse to radius of circular parking orbit
26	AENM	semi major axis of deboost ellipse
27	VAPO	velocity at apoapse of deboost ellipse
28	VELC	circular velocity at periapse of deboost ellipse
29	VELPE	elliptic velocity at periapse of deboost ellipse
30	VELPH1	velocity at periapse of inbound hyperbola
31	VELPH2	velocity at periapse of outbound hyperbola
32	DELVC1	velocity impulse required for circular deboost
33	DELVC2	velocity impulse required for circular boost
34	SUMVC	total Δv required for circular stopover
35	DELVE1	
36	DELVE2	same as three previous definitions applied to elliptic stopover.
37	SUMVE	

The following 10 variables pertain to the Lambert transfer orbit connecting this time-point (I) and the one following:

Var. No.	Name	Definition
38	ELPSD (1,I)	semi-major axis
39	ELPSD (2,I)	eccentricity
40	ELPSD (3,I)	outbound heliocentric velocity magnitude
41	ELPSD (4,I)	outbound flight path angle
42	ELPSD (5,I)	true anomaly of departure
43	ELPSD (6,I)	inclination of transfer with respect to departure planet's orbit
44	ELPSD (7,I)	inbound heliocentric velocity mag.
45	ELPSD (8,I)	π-inbound flight path angle
46	ELPSD (9,I)	true anomaly of arrival
47	ELPSD (10,I)	inclination of transfer with respect to arrival planet's orbit

Ballistic Flyby (IENC=2)

1	NPLNO	
2	TIMEIN	
3	VIN	
4	RASIN	Same definitions as previously given
5	DECLIN	
6	TIMOUT	
7	VOUT	
8	RASOUT	

Var. No.	Name	Definition
9	DECOUT	
10	BA	
11	RASCSP	
12	DECLSP	
13	POMEGA	
14	PLINC	Same definitions as previously given
15	OMEGA1	
16	OMEGA2	
17	OMEGAP	
18	ALPHP	
19	DECLP	
20	PRAD	
21	PALT	
22	VELPER	velocity at periapse
23	DELVBL	$\|v_\infty in\| - \|v_\infty out\|$
24-33		Lambert Data

Single Impulse Flyby (IENC = 3)

1	NPLNO	
2	TIMEIN	
3	VIN	Same as previously defined
4	RASIN	
5	DECLIN	
6	TIMOUT	
7	VOUT	
8	RASOUT	

Var. No.	Name	Definition
9	DECOUT	
10	BA	
11	RASCSP	Same as previously defined
12	DECLSP	
13	DELPSI	
14	PSI1	
15	PSI2	
16	ALTTR1	altitude of optimum transfer
17	RADTR1	radius of optimum transfer
18	DELVT1	minimum Δv to execute transfer
19	GAMT1	flight path angle of transfer velocity
20	THE1T1	true anomaly of transfer on inbound hyperbola
21	THE2T1	true anomaly of transfer on outbound hyperbola
22	VTINT1	velocity on inbound hyperbola at transfer point.
23	VTOUT1	velocity on outbound hyperbola at transfer point
24	PLINC	
25	POMEGA	Same as previously defined
26	OMEGA1	
27	OMEGA2	
28	OMEGT1	angle between ascending node and transfer point
29	ALPH1	right ascension of transfer point
30	DECL1	declination of transfer point
31-40		Lambert Data

Plane Change Maneuver (IENC=7)

Var. No.	Name	Definition
1	NPLNO=0	
2	TIMEIN	time of arrival at point
3	VIN	heliocentric inbound velocity
4	RASIN	not defined
5	DECLIN	not defined
6	TIMOUT	time of departure from point
7	VOUT	heliocentric outbound velocity
8	RASOUT	not defined
9	DECOUT	not defined
10	BA	bend angle between heliocentric vectors
11	RASCSP	not defined
12	DECLSP	not defined
13	ALPH1	angle between extended radius vector and projection of $\Delta\bar{v}$ into arrival plane
14	DECL1	angle between $\Delta\bar{v}$ and projection of same into plane of arrival orbit.
15	DINC	dihedral angle between inbound and outbound planes
16	DELVT1	$\|\Delta v\|$
17-26		Lambert data

APPENDIX C

SUBROUTINE CROSS REFERENCE

Subroutine	Calls
AUXPRO	AUX2 AUX2
AUX 1	ENCTR PRTCHA
AUX 2	CONSEG ENCTR PRTCHA SETUP
CHARAC	CONSEG ENCTR FIXPLC SETUP
CONSEG	BSTST* LAMBRE MATELM
CONST	NONE
CONSTR	CONSEG FIXPLC SETUP
ECANOM	NONE
ENCONT	BSTST EQRTR MATELM OPTRAN PERXFR SUBSOL VASTOT

*BSTST is a set of routines which perform vector and matrix operations. Written by J. E. Holcomb.

Subroutine	Calls
ENCTR	ENCONT LODCHA
EQRTR	NONE
FIXPLC	BSTST ECANOM MANOM MATELM RADMAG SLOPE TRUANM VELMAG
FIXREL	BSTST
LAMBRE	BSTST
LODCHA	NONE
MANOM	NONE
MATELM	NONE
OPTRAN	NONE
PAP (main program)	AUXPRO CONST FIXPLC FIXREL
PERXFR	NONE
PRTCHA	NONE
RADMAG	NONE
SETUP	FIXPLC FIXREL
SLOPE	NONE
SUBSOL	BSTST ECANOM MANOM MATELM TRUANM

Subroutine	Calls
TRUANM	NONE
VASTOT	BSTST
VELMAG	NONE

Subroutine	Called by
AUXPRO	PAP
AUX 1	AUXPRO
AUX 2	AUXPRO
CHARAC	COMCHA
CONSEG	AUX 2 CHARAC CONSTR
CONST	PAP
CONSTR	INISOL routines FILGRD routines
ECANOM	FIXPLC SUBSOL
ENCONT	ENCTR
ENCTR	AUX 1 AUX 2 CHARAC
EQRTR	ENCONT
FIXPLC	CHARAC CONSTR PAP SETUP
FIXREL	PAP SETUP

Subroutine	Called by
LAMBRE	CONSEG
LODCHA	ENCTR
MANOM	FIXPLC SUBSOL
MATELM	CONSEG ENCONT FIXPLC SUBSOL
OPTRAN	ENCONT
PERXFR	ENCONT
PRTCHA	AUX 1 AUX 2 COMCHA
RADMAG	FIXPLC
SETUP	AUX 2 CHARAC CONSTR
SLOPE	FIXPLC
SUBSOL	ENCONT
TRUANM	FIXPLC SUBSOL
VASTOT	ENCONT
VELMAG	FIXPLC

APPENDIX D

COMMON SECTION CROSS REFERENCE

Subroutine	COMMON Sections
AUX 1	AUXPLT CONFAC MCL USER
AUX 2	USER
CHARAC	USER
CONSEG	CONFAC PLANET USER VHELIO
CONST	CONFAC PLANET
CONSTR	USER
ECANOM	NONE
ENCONT	CONFAC PENCTR PLANET USER VHELIO
ENCTR	USER
EQRTR	CONFAC PLANET
FIXPLC	PLANET USER
FIXREL	PLANET USER
LAMBRE	CONFAC
LODCHA	CONFAC PENCTR PLANET

Subroutine	COMMON Sections
MANOM	CONFAC
MATELM	NONE
OPTRAN	CONFAC PENCTR PLANET
PAP (main program)	CONFAC MCL PLANET USER AUXPLT
PERXFR	CONFAC
PRTCHA	USER
RADMAG	NONE
SETUP	USER
SLOPE	CONFAC
SUBSOL	CONFAC PENCTR PLANET
TRUANM	CONFAC
VASTOT	CONFAC PENCTR PLANET
VELMAG	NONE

COMMON Section	Used In
AUXPLT	AUX1 PAP
CONFAC	CONSEG CONST ENCONT EQRTR PAP LAMBRE LODCHA MANOM OPTRAN PERXFR SLOPE SUBSOL TRUANM VASTOT
MCL	PAP
PENCTR	ENCONT LODCHA OPTRAN SUBSOL VASTOT
PLANET	CONSEG CONST ENCONT EQRTR FIXPLC FIXREL PAP LODCHA OPTRAN SUBSOL VASTOT
USER	AUX 1 AUX 2 CHARAC CONSEG CONSTR ENCONT ENCTR FIXPLC FIXREL PAP PRTCHA SETUP
VHELIO	CONSEG ENCONT

APPENDIX E

FILE ASSIGNMENTS

PAP Subfunction	Input Files	Output Files
INISOL	-	8,9,10
FILGRD	8,9,10	11,12,13
COMCHA	11,12,13	14,15,16
AUXPRO	14,15,16	17,18
PLTCON	14,15,16,17,18	-

```
KON=1,2,3,6,IPRINT=1,ISAC=1,
$END
ISGR=1,
ICCH=1,
NCS=2,
TARV=0.,1.2,
EPS=.0005,.0005,
IEQC=1,2,7,8,
IP1=12,
IP2=42,
ICS=21,155,
SSCON=100.,0.,0.,100.,
PERDEC(1)=18.,PERDEC(4)=18.,
RATS=1.,0.,0.,1.,
IPS=2.,43286.,43294.,2.,9*0.,
4.,43550.,43600.,40.,9*0.,
0.,43725.,43725.,0.,.7,2.5,.1,253.,253.,4*0.,
0.,43810.,43820.,2.,9*0.,
$END

*****CALL INISOL AT TIME 23:52:35.792*****
```

		PI =	.43286000+05	.43810000+05	.43350001+05	.7[illegible]-00
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.43[illegible]200+05	.7761[illegible]-00
		PI =	.43286000+05	.43810000+05	.43350001+05	.7999[illegible]-00
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.43445870+05	.7738[illegible]-00
		PI =	.43286000+05	.43810000+05	.43350001+05	.699999[illegible]-00
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.4340[illegible]+05	.72[illegible]-00
		PI =	.43286000+05	.43810000+05	.43350001+05	.999999[illegible]-00
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.43[illegible]+05	.7781941[illegible]-00
		PI =	.43286000+05	.43810000+05	.43[illegible]+05	.110[illegible]+01
OUTBUS	4 ITRS	PF =	.43286000+05	.43810000+05	.43453111+05	.8502[illegible]-00
		PI =	.43286000+05	.43810000+05	.43350001+05	.12000000+01
MAXITR	20 ITRS	PF	.43286000+05	.43810000+05	.434[illegible]+05	.1201248[illegible]+01
		PI =	.43286000+05	.43810000+05	.43350001+05	.13000000+01
OUTBUS	2 ITRS	PF =	.43286000+05	.43810000+05	.43420657+05	.1536[illegible]+01
		PI =	.43286000+05	.43810000+05	.43350001+05	.1399[illegible]+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43[illegible]126+05	.1700169[illegible]+01
		PI =	.43286000+05	.43810000+05	[illegible]	.14[illegible]+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43358813+05	.[illegible]1
		PI =	.43286000+05	.43810000+05	.43350001+05	.15[illegible]+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43[illegible]126+05	.[illegible]+01
		PI =	.43286000+05	.43810000+05	.43350001+05	.1[illegible]+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.4337[illegible]+05	[illegible]
		PI =	.43286000+05	.43810000+05	.433[illegible]+05	[illegible]
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.433[illegible]+05	[illegible]
		PI =	.43286000+05	.43810000+05	.433[illegible]+05	.1[illegible]
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.4336[illegible]+05	.3307[illegible]
		PI =	.43286000+05	.43810000+05	.43350001+05	.1[illegible]
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.44755250+05	.43[illegible]11[illegible]+01
		PI =	.43286000+05	.43810000+05	.4335[illegible]+05	.2[illegible]+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43344750+05	.5[illegible]+01
		PI =	.43286000+05	.43810000+05	[illegible]	[illegible]
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43100000+05	.2[illegible]
		PI =	.43286000+05	.43810000+05	.4335[illegible]	.2200[illegible]
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43328001+05	.6[illegible]
		PI =	.43286000+05	.43810000+05	.43350001+05	.2399[illegible]
UTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.433[illegible]+05	.72[illegible]+01

		PI =	.43286000+05	.43810000+05	.43550000+05	.16999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43219376+05	.28113022+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.17999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43142375+05	.32829819+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.18999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43061125+05	.38071060+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.19999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42977000+05	.43848267+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.20999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42889751+05	.50172577+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.21999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42800251+05	.57054901+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.22999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42711500+05	.64511719+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.23999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42618500+05	.72552185+01
		PI =	.43286000+05	.43810000+05	.43550000+05	.24999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42526501+05	.81186219+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.70000000-00
CONVER	6 ITRS	PF =	.43286000+05	.43810000+05	.43512332+05	.77413064-00
		PI =	.43286000+05	.43810000+05	.43590000+05	.79999999-00
CONVER	4 ITRS	PF =	.43286000+05	.43810000+05	.43513060+05	.77412859-00
		PI =	.43286000+05	.43810000+05	.43590000+05	.89999998-00
CONVER	7 ITRS	PF =	.43286000+05	.43810000+05	.43512098+05	.77413058-00
		PI =	.43286000+05	.43810000+05	.43590000+05	.99999998-00
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.43525090+05	.77893695-00
		PI =	.43286000+05	.43810000+05	.43590000+05	.11000000+01
OUTBUS	3 ITRS	PF =	.43286000+05	.43810000+05	.43425375+05	.27839923-00
		PI =	.43286000+05	.43810000+05	.43590000+05	.12000000+01
OUTBUS	2 ITRS	PF =	.43286000+05	.43810000+05	.43500017+05	.11999714+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.13000000+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43506329+05	.14280953+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.13999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43477719+05	.17001667+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.14999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43442813+05	.20209084+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.15999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43402313+05	.23909646+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.16999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43356750+05	.28112946+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.17999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43307251+05	.32829895+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.18999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43254625+05	.38070907+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.19999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43199750+05	.43848267+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.20999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43142750+05	.50172577+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.21999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43150000+05	.57055359+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.22999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.43025000+05	.64511719+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.23999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42966001+05	.72551880+01
		PI =	.43286000+05	.43810000+05	.43590000+05	.24999999+01
OUTBUS	1 ITRS	PF =	.43286000+05	.43810000+05	.42907500+05	.81186523+01

2 SOLUTION(S) FOUND

```
        1    .43286000+05    .43810000+05    .43434472+05    .77413071-00
        2    .43286000+05    .43810000+05    .43512549+05    .77420904-00

*****SYSTEM EVALUATED     0 TIMES IN INISOL*****


*****CALL FILGRD AT TIME 23:53:14.634*****


*****MAP SOLUTION NO.  1*****

                  PI =    .43286000+05    .43810000+05    .43434472+05    .77413071-00
CONVER  1 ITRS    PF =    .43286000+05    .43810000+05    .43434476+05    .77413073-00
                  PI =    .43288001+05    .43810000+05    .43434472+05    .77413071-00
CONVER  3 ITRS    PF =    .43288001+05    .43810000+05    .43434373+05    .77413072-00
                  PI =    .43286000+05    .43812001+05    .43434472+05    .77413071-00
CONVER  3 ITRS    PF =    .43286000+05    .43812001+05    .43434369+05    .78944232-00
                  PI =    .43288001+05    .43812001+05    .43434472+05    .77413071-00
CONVER  3 ITRS    PF =    .43288001+05    .43812001+05    .43434277+05    .78944241-00
                  PI =    .43288001+05    .43814000+05    .43434182+05    .80475411-00
CONVER  1 ITRS    PF =    .43288001+05    .43814000+05    .43434183+05    .80454280-00
                  PI =    .43290000+05    .43812001+05    .43434186+05    .78944200-00
CONVER  1 ITRS    PF =    .43290000+05    .43812001+05    .43434163+05    .78944779-00
                  PI =    .43286000+05    .43814000+05    .43434261+05    .80475491-00
CONVER  1 ITRS    PF =    .43286000+05    .43814000+05    .43434279+05    .80454280-00
                  PI =    .43290000+05    .43810000+05    .43434270+05    .77413072-00
CONVER  1 ITRS    PF =    .43290000+05    .43810000+05    .43434267+05    .77413070-00
                  PI =    .43286000+05    .43816000+05    .43434191+05    .81964278-00
CONVER  1 ITRS    PF =    .43286000+05    .43816000+05    .43434194+05    .81943307-00
                  PI =    .43292000+05    .43810000+05    .43434160+05    .77413069-00
CONVER  1 ITRS    PF =    .43292000+05    .43810000+05    .43434169+05    .77413072-00
                  PI =    .43286000+05    .43818001+05    .43434110+05    .83432335-00
CONVER  1 ITRS    PF =    .43286000+05    .43818001+05    .43434133+05    .83414283-00
                  PI =    .43288001+05    .43816000+05    .43434194+05    .81943307-00
CONVER  3 ITRS    PF =    .43288001+05    .43816000+05    .43434108+05    .81943070-00
                  PI =    .43290000+05    .43814000+05    .43434087+05    .80454281-00
CONVER  1 ITRS    PF =    .43290000+05    .43814000+05    .43434093+05    .80454031-00
                  PI =    .43292000+05    .43812001+05    .43434050+05    .78945318-00
CONVER  2 ITRS    PF =    .43292000+05    .43812001+05    .43434069+05    .78944787-00
                  PI =    .43294001+05    .43810000+05    .43434072+05    .77413075-00
CONVER  1 ITRS    PF =    .43294001+05    .43810000+05    .43434070+05    .77413068-00
                  PI =    .43286000+05    .43820000+05    .43434072+05    .84885259-00
CONVER  1 ITRS    PF =    .43286000+05    .43820000+05    .43434062+05    .84867859-00
                  PI =    .43288001+05    .43818001+05    .43434133+05    .83414283-00
CONVER  3 ITRS    PF =    .43288001+05    .43818001+05    .43434044+05    .83414081-00
                  PI =    .43290000+05    .43816000+05    .43434021+05    .81942334-00
CONVER  1 ITRS    PF =    .43290000+05    .43816000+05    .43434019+05    .81943071-00
                  PI =    .43292000+05    .43814000+05    .43434003+05    .80453783-00
CONVER  1 ITRS    PF =    .43292000+05    .43814000+05    .43434000+05    .80454024-00
                  PI =    .43294001+05    .43812001+05    .43433973+05    .78944795-00
CONVER  1 ITRS    PF =    .43294001+05    .43812001+05    .43433996+05    .78944787-00
                  PI =    .43288001+05    .43820000+05    .43434062+05    .84867859-00
CONVER  3 ITRS    PF =    .43288001+05    .43820000+05    .43433987+05    .84867671-00
                  PI =    .43290000+05    .43818001+05    .43433955+05    .83413881-00
CONVER  1 ITRS    PF =    .43290000+05    .43818001+05    .43433963+05    .83414079-00
                  PI =    .43292000+05    .43816000+05    .43433931+05    .81943072-00
CONVER  1 ITRS    PF =    .43292000+05    .43816000+05    .43433938+05    .81943071-00
```

```
                    PI =    .43294001+05    .43814000+05    .43433907+05    .80454017-00
CONVER   1 ITRS     PF =    .43294001+05    .43814000+05    .43433921+05    .80454029-00
                    PI =    .43290000+05    .43820000+05    .43433913+05    .84867483-00
CONVER   1 ITRS     PF =    .43290000+05    .43820000+05    .43433904+05    .84867668-00
                    PI =    .43292000+05    .43818001+05    .43433882+05    .83414078-00
CONVER   1 ITRS     PF =    .43292000+05    .43818001+05    .43433876+05    .83414085-00
                    PI =    .43294001+05    .43816000+05    .43433858+05    .81943071-00
CONVER   1 ITRS     PF =    .43294001+05    .43816000+05    .43433852+05    .81943071-00
                    PI =    .43292000+05    .43820000+05    .43433822+05    .84867665-00
CONVER   1 ITRS     PF =    .43292000+05    .43820000+05    .43433834+05    .84867672-00
                    PI =    .43294001+05    .43818001+05    .43433789+05    .83414091-00
CONVER   1 ITRS     PF =    .43294001+05    .43818001+05    .43433808+05    .83414083-00
                    PI =    .43294001+05    .43820000+05    .43433764+05    .84867676-00
CONVER   1 ITRS     PF =    .43294001+05    .43820000+05    .43433749+05    .84867670-00

*****PERCENTAGE OF POINTS CONVERGING = 100.000*****
```

TRANS ORBIT CHAR

			PLANET	THETA (DEG)	INC (DEG)	HF VEL (EMOS)	FP ANG (DEG)	R.ASCN (DEG)	DECLIN (DEG)	H.E.S. (EMOS)	DATE (J.DATE)
ECCEN =	.46065	DEPART	2.	331.30	3.45	1.4068	98.95	274.83	15.01	.3233	43286.00
S.M.A.=	1.29573	ARRIVE	4.	436.67	-1.61	.7286	115.42	137.44	12.75	.2781	43434.37
ECCEN =	.36326	DEPART	4.	152.87	17.81	.6835	76.25	119.10	69.41	.2780	43434.37
S.M.A.=	1.19683	ARRIVE	0.	330.44	-19.49	1.3930	82.24	.00	.00	1.3030	43725.00
ECCEN =	.41929	DEPART	0.	307.08	.00	1.3039	104.95	.00	.00	1.3039	43725.00
S.M.A.=	1.12994	ARRIVE	3.	450.54	.00	1.0867	112.63	33.44	13.44	.4360	43812.00

DEPARTURE CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
-.00	23.53	23.53	41.67	100.00	23465.8	23465.8	45813.5	45813.5	22347.6	22347.6	44695.3	22347.6	22347.6	44695.3	1.0301

INCLIN (DEG)	C.OMEG. (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA P (DEG)	ALPHA P (DEG)	DELTA P (DEG)	A/RP (N.DIM)	SMA.ELL (NM)	APO-VEL (FPS)
30.08	122.40	107.22	148.89	215.06	156.51	18.00	1.0090	3420.	23465.8

FLYBY CHARACTERISTICS

PLANET	BEND ANG (DEG)	OMEGA (DEG)	INCL (DEG)	OMEGA1 (DEG)	OMEGA2 (DEG)	OMEGAP (DEG)	ALPHAP (DEG)	DELTAP (DEG)	PER RAD (P.RAD)	PER ALT (N.M.)	PER VEL (FT/SEC)	DELTA V (EMOS)	SUB-SOLAR PT. R.ASC (DEG)	DECLIN (DEG)
[illegible]	57.85	139.11	97.33	12.86	70.71	131.78	147.23	-47.70	.1958	-1479.67	46069.10	.0001	-11.91	-5.25

PLANE CHANGE MANEUVER

B.ANGLE (DEG)	ALPHA V (DEG)	DECLIN V (DEG)	DEL.INC. (DEG)	DELTAV (EMOS)
20.40	-110.35	-1.57	19.49	.4617

ARRIVAL CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
-.00	15.32	15.32	39.54	100.00	25579.8	25579.8	55951.[illegible]	55951.	30371.7	30371.7	60743.3	30371.7	30371.7	60743.3	1.0221

INCLIN (DEG)	C.OMEGA (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA P (DEG)	ALPHA P (DEG)	DELTA P (DEG)	A/RP (N.DIM)	SMA.ELL (NM)	APO-VEL (FPS)
20.26	-106.22	137.89	168.49	243.17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

```
TRANS ORBIT CHAR                              PLANET  THETA    INC    HE VEL  DP ANG  R.ASCN  DECLIN  H.E.S.     DATE
                                                      (DEG)   (DEG)   (EMOS)   (DEG)   (DEG)   (DEG)  (EMOS)   (J.DATE)

 ECCEN =    .46907          DEPART               2.   340.07    1.90  1.4159   96.33  275.56    9.26   .2902   43294.00
 S.M.A.=  1.34432           ARRIVE               4.   492.46     .04   .7480  116.86  142.31   15.32   .2950   43433.75

 ECCEN =    .32436          DEPART               4.   153.33   20.43   .6944   78.43  111.87   76.86   .2953   43433.75
 S.M.A.=  1.21783           ARRIVE               0.   331.19  -22.11  1.2393   83.07     .00     .00  1.2393   43725.00

 ECCEN =    .40483          DEPART               0.   296.75     .00  1.2342  107.00     .00     .00  1.2342   43725.00
 S.M.A.=  1.20000           ARRIVE               3.   448.22     .00  1.0886  111.78   42.25   16.26   .4181   43820.00

DEPARTURE CHARACTERISTICS
  DEL.PSI  PSI 1    PSI 2   B.ANGLE   P.ALT   V-CIRC   VP-ELL   VP-HY 1  VP-HY 2  DV-C 1   DV-C 2  DVC-TOT  DV-E 1   DV-E 2  DVE-TOT PER. RAD.
   (DEG)   (DEG)    (DEG)    (DEG)    (NM)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)  (PL.RAD.)
     .00    23.96    23.98    47.96  100.00  23465.8  23465.8  43651.0  43651.0  20185.2  20185.2  40370.4  20185.2  20185.2  40370.4    1.0301

  INCLIN  C.OMEGA OMEGA 1 OMEGA 2 OMEGA P ALPHA P DELTA P   A/RP    SMA.ELL APO-VEL
   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)  (N.DIM)   (NM)    (FPS)
   26.12  114.98  110.60  158.56  224.58  156.48   18.00  1.0000    3420.  23465.8

                                                                                                                  SUB-SOLAR PT.
FLYBY CHARACTERISTICS                                                                                              R.ASC   DECLIN
PLANET BEND ANG  OMEGA    INCL    OMEGA1  OMEGA2  OMEGAP  ALPHAP  DELTAP  PER RAD   PER ALT    PER VEL   DELTA V
          (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)  (P.RAD)    (N.M.)   (FT/SEC)   (EMOS)    (DEG)    (DEG)
  4.      63.49  144.28   97.13   15.44   78.93  137.18  150.84  -42.41   .1468  -1569.87   51730.18   -.0003   -12.20    -5.37

PLANE CHANGE MANEUVER
  B.ANGLE   ALPHA V   DECLIN V   DEL.INC.   DELTAV
   (DEG)     (DEG)     (DEG)      (DEG)     (EMOS)
   23.82   -114.47     -1.95      22.11      .5105

ARRIVAL CHARACTERISTICS
  DEL.PSI  PSI 1    PSI 2   B.ANGLE   P.ALT   V-CIRC   VP-ELL   VP-HY 1  VP-HY 2  DV-C 1   DV-C 2  DVC-TOT  DV-E 1   DV-E 2  DVE-TOT PER. RAD.
   (DEG)   (DEG)    (DEG)    (DEG)    (NM)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)  (PL.RAD.)
    -.00    16.36    16.36    32.72  100.00  25579.8  25579.8  54565.4  54565.4  28985.6  28985.6  57971.2  28985.6  28985.6  57971.2    1.0291

  INCLIN  C.OMEGA OMEGA 1 OMEGA 2 OMEGA P ALPHA P DELTA P   A/RP    SMA.ELL APO-VEL
   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)   (DEG)  (N.DIM)   (NM)    (FPS)
   21.63  -90.42  130.60  163.31  236.95  324.59   18.00  1.0000    3538.  25579.8
IUNIT=14,KON=5,7,NFR=2,
$END
IOS=2,
FTLT='MARS PASSAGE RADIUS CONTOURS                    ',
ALAB='EARTH ARRIVAL J.D.                              ',
OLAB='VENUS DEPARTURE J.D.                            ',
LABC=1,
OMI=43810.,
OMA=43820.,
AMI=43286.,
AMA=43294.,
NFU=1,
IUN=14,
ISO=1,
IVA=220,
P1IN=.05,P2IN=.05,
CMI=.17,
CMA=.19,
CIN=.01,
```

```
IAUS=0,
$END
$TLT='PLANE-CHANGE DELTAV CONTOURS                          ',
IVA=316,
CMI=.46,
CMA=.50,
CIN=.02,
$END

*****CALL PLTCON AT TIME 23:53:45.687*****
```

F8 (Condensed printout)

PLANET	TIME	INBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	OUTBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	PASSAGE RAD. (P.RAD)
2.0	43286.00	.3231	232.96	28.65	.3231	274.82	15.04	
4.0	43434.48	.2777	137.43	12.73	.2777	120.15	67.62	.2076
.0	43725.00	1.3206	.00	.00	1.3229	.00	.00	
3.0	43810.00	.4429	71.21	12.67	.4429	60.09	3.39	
******	******	******	******	******	******	******	******	******
2.0	[illegible]	.3233	[illegible]	28.61	.3233	274.83	15.01	
4.0	43434.37	.2781	137.[illegible]	12.75	.2780	119.10	69.41	.1958
.0	43725.00	1.30[illegible]	[illegible]	.00	1.3039	.00	.00	
3.0	[illegible]	[illegible]	[illegible]	13.44	.4369	62.96	3.96	
******	******	******	******	******	******	******	******	******
2.0	[illegible]	.3234	[illegible]	28.57	.3234	274.83	14.98	
4.0	43434.28	[illegible]	[illegible]	12.77	.2792	117.84	71.10	.1855
.0	43725.00	1.28[illegible]	[illegible]	.00	1.2855	.00	.00	
3.0	[illegible]	.431[illegible]	[illegible]	14.19	.4315	65.79	4.52	
******	******	******	******	******	******	******	******	******
2.0	[illegible]	[illegible]	[illegible]	28.53	.3235	274.84	14.96	
4.0	43434.19	[illegible]	137.[illegible]	12.78	.2797	116.37	72.73	.1759
.0	43725.00	[illegible]	[illegible]	.00	1.2678	.00	.00	
3.0	[illegible]	[illegible]	[illegible]	14.90	.4265	68.57	5.05	
******	******	******	******	******	******	******	******	******
2.0	[illegible]	.3236	[illegible]	[illegible]	.3236	274.84	14.94	
4.0	43434.13	.2790	137.47	12.79	.2796	114.63	74.24	.1676
.0	43725.00	1.2542	.00	.00	1.2507	.00	.00	
3.0	43818.00	.4221	46.06	15.59	.4221	71.31	5.57	
******	******	******	******	******	******	******	******	******
2.0	43286.00	.3237	232.23	28.48	.3237	274.84	14.92	
4.0	43434.06	.2793	137.47	12.80	.2794	112.53	75.75	.1595
.0	43725.00	1.2391	.00	.00	1.2342	.00	.00	
3.0	43820.00	.4161	42.25	16.26	.4181	74.01	6.08	
******	******	******	******	******	******	******	******	******
2.0	[illegible]	.3131	231.18	26.78	.3131	275.23	12.81	
4.0	43434.37	.2815	[illegible]	13.87	.2815	120.12	68.06	.2052
.0	43725.00	1.3208	.00	.00	1.3229	.00	.00	
3.0	43810.00	.4429	31.21	12.67	.4429	60.09	3.39	
******	******	******	******	******	******	******	******	******
2.0	43288.00	.3133	231.22	26.76	.3133	275.23	12.79	
4.0	43434.28	.2819	138.93	13.88	.2816	119.05	69.79	.1940
.0	43725.00	1.3031	.00	.00	1.3039	.00	.00	
3.0	43812.00	.4369	33.44	13.44	.4369	62.96	3.96	
******	******	******	******	******	******	******	******	******
2.0	43288.00	.3134	231.26	26.73	.3134	275.24	12.77	
4.0	43434.18	.2822	138.93	13.88	.2822	117.78	71.49	.1836
.0	43725.00	1.2861	.00	.00	1.2855	.00	.00	
3.0	43814.00	.4315	35.65	14.19	.4315	65.79	4.52	
******	******	******	******	******	******	******	******	******
2.0	43286.00	.3135	231.29	26.70	.3135	275.25	12.76	
4.0	43434.11	.2825	138.94	13.89	.2825	116.28	73.07	.1744
.0	43725.00	1.2699	.00	.00	1.2678	.00	.00	
3.0	43816.00	.4265	37.86	14.90	.4265	68.58	5.05	

PLANET	TIME	INBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	OUTBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	PASSAGE RAD. (P.RAD)
2.0	43288.00	.3136	231.32	26.63	.3136	275.25	12.75	
4.0	43434.04	.2828	13[illegible].34	13.89	.2828	114.52	74.58	.1661
.0	43725.00	1.2542	.00	.00	1.2507	.00	.00	
3.0	43818.00	.4221	40.96	15.59	.4221	71.31	5.57	
******								
2.0	43288.00	.3137	231.35	26.67	.3137	275.26	12.74	
4.0	43433.99	.2830	138.94	13.90	.2830	112.40	76.01	.1585
.0	43725.00	1.2392	.00	.00	1.2342	.00	.00	
3.0	43820.00	.4181	42.25	16.26	.4181	74.01	6.08	
******								
2.0	43290.00	.3045	230.13	25.61	.3045	275.50	11.26	
4.0	43434.27	.2855	140.19	14.57	.2856	120.08	68.51	.2000
.0	43725.00	1.3203	.00	.00	1.3229	.00	.00	
3.0	43810.00	.4429	31.21	12.67	.4429	60.09	3.39	
******								
2.0	43290.00	.3047	230.18	25.59	.3047	275.51	11.25	
4.0	43434.16	.2859	140.20	14.58	.2863	118.98	70.26	.1889
.0	43725.00	1.3031	.00	.00	1.3038	.00	.00	
3.0	43812.00	.4369	33.44	13.44	.4369	62.96	3.96	
******								
2.0	43290.00	.3048	230.22	25.57	.3048	275.52	11.24	
4.0	43434.09	.2862	140.20	14.58	.2861	117.70	71.85	.1795
.0	43725.00	1.2862	.00	.00	1.2855	.00	.00	
3.0	43814.00	.4315	35.65	14.19	.4315	65.79	4.52	
******								
2.0	43290.00	.3049	230.25	25.56	.3049	275.52	11.23	
4.0	43434.02	.2865	140.20	14.58	.2866	116.18	73.42	.1706
.0	43725.00	1.2699	.00	.00	1.2678	.00	.00	
3.0	43816.00	.4265	37.36	14.90	.4265	68.58	5.05	
******								
2.0	43290.00	.3050	230.28	25.54	.3050	275.53	11.22	
4.0	43433.96	.2866	140.20	14.58	.2866	114.39	74.88	.1627
.0	43725.00	1.2542	.00	.00	1.2507	.00	.00	
3.0	43818.00	.4221	40.96	15.59	.4221	71.31	5.57	
******								
2.0	43290.00	.3051	230.31	25.53	.3051	275.53	11.22	
4.0	43433.90	.2870	140.21	14.59	.2871	112.25	76.31	.1553
.0	43725.00	1.2391	.00	.00	1.2342	.00	.00	
3.0	43820.00	.4181	42.25	16.26	.4181	74.01	6.08	
******								
2.0	43292.00	.2967	228.91	24.87	.2967	275.60	10.13	
4.0	43434.17	.2896	141.31	15.02	.2895	120.03	68.93	.1938
.0	43725.00	1.3209	.00	.00	1.3229	.00	.00	
3.0	43810.00	.4429	31.21	12.67	.4429	60.09	3.39	
******								
2.0	43292.00	.2969	228.96	24.85	.2969	275.61	10.12	
4.0	43434.07	.2900	141.31	15.03	.2904	119.90	70.66	.1832
.0	43725.00	1.3032	.00	.00	1.3038	.00	.00	
3.0	43812.00	.4369	33.44	13.44	.4369	62.96	3.96	

PLANET	TIME	INBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	OUTBOUND ASYMPTOTE H.E.S. (EMOS)	R.ASC. (DEG)	DECLIN (DEG)	PASSAGE RAD. (P.RAD)
2.0	43292.00	.2970	229.00	24.84	.2970	275.62	10.12	
4.0	43434.00	.2903	141.32	15.03	.2904	117.61	72.23	.1742
.0	43725.00	1.2861	.00	.00	1.2855	.00	.00	
3.0	43814.00	.4315	35.65	14.19	.4315	65.79	4.52	
2.0	43292.00	.2971	229.03	24.82	.2971	275.63	10.11	
4.0	43433.94	.2905	141.32	15.03	.2904	116.08	73.73	.1659
.0	43725.00	1.2699	.00	.00	1.2678	.00	.00	
3.0	43816.00	.4265	37.86	14.90	.4265	68.58	5.05	
2.0	43292.00	.2972	229.06	24.81	.2972	275.64	10.11	
4.0	43433.83	.2908	141.32	15.03	.2909	114.24	75.20	.1582
.0	43725.00	1.2543	.00	.00	1.2507	.00	.00	
3.0	43818.00	.4221	40.06	15.59	.4221	71.31	5.57	
2.0	43292.00	.2973	229.08	24.80	.2973	275.64	10.11	
4.0	43433.83	.2910	141.32	15.03	.2908	112.09	76.56	.1514
.0	43725.00	1.2392	.00	.00	1.2342	.00	.00	
3.0	43820.00	.4181	42.25	16.26	.4181	74.01	6.08	
2.0	43294.00	.2896	227.51	24.39	.2896	275.51	9.28	
4.0	43434.07	.2937	142.31	15.32	.2937	119.96	69.35	.1870
.0	43725.00	1.3209	.00	.00	1.3229	.00	.00	
3.0	43810.00	.4429	31.21	12.67	.4429	60.09	3.39	
2.0	43294.00	.2898	227.55	24.38	.2898	275.52	9.27	
4.0	43434.00	.2940	142.31	15.32	.2936	118.84	70.97	.1775
.0	43725.00	1.3031	.00	.00	1.3038	.00	.00	
3.0	43812.00	.4369	33.44	13.44	.4369	62.96	3.96	
2.0	43294.00	.2899	227.59	24.37	.2899	275.53	9.27	
4.0	43433.92	.2943	142.31	15.32	.2941	117.51	72.54	.1687
.0	43725.00	1.2862	.00	.00	1.2855	.00	.00	
3.0	43814.00	.4315	35.65	14.19	.4315	65.79	4.52	
2.0	43294.00	.2900	227.63	24.36	.2900	275.54	9.27	
4.0	43433.85	.2946	142.31	15.32	.2947	115.94	74.07	.1607
.0	43725.00	1.2699	.00	.00	1.2678	.00	.00	
3.0	43816.00	.4265	37.86	14.90	.4265	68.58	5.05	
2.0	43294.00	.2901	227.66	24.35	.2901	275.55	9.26	
4.0	43433.81	.2948	142.31	15.32	.2944	114.12	75.46	.1537
.0	43725.00	1.2543	.00	.00	1.2507	.00	.00	
3.0	43818.00	.4221	40.06	15.59	.4221	71.31	5.57	
2.0	43294.00	.2902	227.69	24.34	.2902	275.56	9.26	
4.0	43433.75	.2950	142.31	15.32	.2953	111.87	76.86	.1468
.0	43725.00	1.2393	.00	.00	1.2342	.00	.00	
3.0	43820.00	.4181	42.25	16.26	.4181	74.01	6.08	

APPENDIX G

TRANS ORBIT CHAR

			PLANET	THETA (DEG)	INC (DEG)	HE VEL (EMOS)	DP ANG (DEG)	R.ASCN (DEG)	DECLIN (DEG)	H.E.S. (EMOS)	DATE (J.DATE)
ECCEN =	.65675	DEPART	3.	156.60	8.32	.7489	56.71	275.77	-14.11	.5654	43000.00
S.M.A.=	.70815	ARRIVE	1.	296.73	-10.96	2.2407	65.64	198.09	-20.40	1.1139	43148.46
ECCEN =	.57241	DEPART	1.	51.41	27.10	2.1999	71.75	10.76	58.71	1.1138	43148.46
S.M.A.=	.62749	ARRIVE	2.	222.84	-28.75	1.0758	56.15	351.53	-30.92	.8363	43286.00
ECCEN =	.46036	DEPART	2.	331.31	3.46	1.4067	98.95	274.82	15.04	.3231	43286.00
S.M.A.=	1.29514	ARRIVE	4.	496.74	-1.62	.7283	115.39	137.43	12.73	.2777	43434.48
ECCEN =	.37358	DEPART	4.	152.79	17.42	.6807	75.66	120.15	67.62	.2777	43434.48
S.M.A.=	1.19158	ARRIVE	0.	330.30	-19.10	1.3208	82.05	.00	.00	1.3208	43725.00
ECCEN =	.42403	DEPART	0.	309.80	.00	1.3229	104.37	.00	.00	1.3229	43725.00
S.M.A.=	1.20000	ARRIVE	3.	451.25	.00	1.0862	113.17	31.21	12.67	.4429	43810.00

DEPARTURE CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
.00	10.17	10.17	20.34	100.00	25579.8	25579.8	66033.1	66033.1	40453.2	40453.2	80906.5	40453.2	40453.2	80906.5	1.0291

INCLIN (DEG)	C.OMEGA (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA P (DEG)	ALPHA P (DEG)	DELTA P (DEG)	A/RP (N.DIM)	SMA.ELL (NM)	APO-VEL (FPS)
21.34	55.74	201.71	222.05	301.88	179.47	18.00	1.0000	3538.	25579.8

FLYBY CHARACTERISTICS

PLANET	BEND ANG (DEG)	OMEGA (DEG)	INCL (DEG)	OMEGA1 (DEG)	OMEGA2 (DEG)	OMEGAP (DEG)	ALPHAP (DEG)	DELTAP (DEG)	PER RAD (P.RAD)	PER ALT (N.M.)	PER VEL (FT/SEC)	DELTA V (EMOS)	SUB-SOLAR PT. R.ASC (DEG)	SUB-SOLAR PT. DECLIN (DEG)
1.	141.32	-159.78	84.30	339.50	120.82	140.16	-164.52	-39.61	.0005	-1305.87	638709.20	.0001	-159.41	.00

IMPULSIVE FLYBY CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI2 (DEG)	B.ANGLE (DEG)	ALTOPT (NM)	TR.OPT (PL.RAD)	DVOPT (FPS)	GAMMV (DEG)	THET1 (DEG)	THET2 (DEG)	VT-IN (FPS)	VT-OUT (FPS)
61.51	4.37	20.85	86.73	38207.58	12.5083	63388.1	73.15	38.74	104.12	82265.7	32977.4

INCLIN (DEG)	C.OMEGA (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA T (DEG)	ALPHA T (DEG)	DELTA T (DEG)
143.87	-63.58	299.38	26.11	175.02	120.45	2.94

FLYBY CHARACTERISTICS

PLANET	BEND ANG (DEG)	OMEGA (DEG)	INCL (DEG)	OMEGA1 (DEG)	OMEGA2 (DEG)	OMEGAP (DEG)	ALPHAP (DEG)	DELTAP (DEG)	PER RAD (P.RAD)	PER ALT (N.M.)	PER VEL (FT/SEC)	DELTA V (EMOS)	SUB-SOLAR PT. R.ASC (DEG)	SUB-SOLAR PT. DECLIN (DEG)
4.	56.05	139.17	97.64	12.85	68.90	130.88	147.90	-48.54	.2076	-1458.05	45178.43	.0000	-11.86	-5.22

PLANE CHANGE MANEUVER

B.ANGLE (DEG)	ALPHA V (DEG)	DECLIN V (DEG)	DEL.INC. (DEG)	DELTAV (EMOS)
19.79	-108.70	-1.49	19.10	.4543

ARRIVAL CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
-.00	15.01	15.01	30.02	100.00	25579.8	25579.8	56398.9	56398.9	30819.1	30819.1	61638.1	30819.1	30819.1	61638.1	1.0291

INCLIN C.OMEGA OMEGA 1 OMEGA 2 OMEGA P ALPHA P DELTA P A/RP SMA.ELL APO-VEL

(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(N.DIM)	(NM)	(FPS)
19.94	-110.51	139.98	170.00	244.99	313.10	18.00	1.0000	3538.	25579.8

2442

TRANS ORBIT CHAR

			PLANET	THETA (DEG)	INC (DEG)	HE VEL (EMOS)	DP ANG (DEG)	R.ASCN (DEG)	DECLIN (DEG)	H.E.S. (EMOS)	DATE (J.DATE)
ECCEN =	.64056	DEPART	3.	156.52	10.15	.7502	58.25	277.14	-11.52	.5523	43002.00
S.M.A.=	.70883	ARRIVE	1.	301.37	-12.23	2.2292	67.70	203.15	-23.51	1.0946	43149.55
ECCEN =	.57628	DEPART	1.	52.52	27.03	2.1915	71.30	17.65	59.51	1.0949	43149.55
S.M.A.=	.63392	ARRIVE	2.	223.52	-28.21	1.0827	55.72	357.08	-30.33	.8375	43290.00
ECCEN =	.46406	DEPART	2.	335.62	2.44	1.4112	97.67	275.50	11.26	.3045	43290.00
S.M.A.=	1.31883	ARRIVE	4.	494.61	-.54	.7379	116.11	140.19	14.57	.2855	43434.26
ECCEN =	.37353	DEPART	4.	152.73	18.20	.6809	75.63	120.09	68.52	.2856	43434.26
S.M.A.=	1.19170	ARRIVE	0.	330.35	-19.88	1.3207	82.06	.00	.00	1.3207	43725.00
ECCEN =	.42403	DEPART	0.	309.80	.00	1.3229	104.37	.00	.00	1.3229	43725.00
S.M.A.=	1.20000	ARRIVE	3.	451.25	.00	1.0862	113.17	31.21	12.67	.4429	43810.00

DEPARTURE CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
.00	10.57	10.57	21.15	100.00	25579.8	25579.8	64965.7	64965.7	39385.9	39385.9	78771.7	39385.9	39385.9	78771.7	1.0291

INCLIN (DEG)	C.OMEGA (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA P (DEG)	ALPHA P (DEG)	DELTA P (DEG)	A/RP (N.DIM)	SMA.ELL (NM)	APO-VEL (FPS)
19.97	63.02	194.65	215.79	295.22	179.63	18.00	1.0000	3538.	25579.8

FLYBY CHARACTERISTICS

PLANET	BEND ANG (DEG)	OMEGA (DEG)	INCL (DEG)	OMEGA1 (DEG)	OMEGA2 (DEG)	OMEGAP (DEG)	ALPHAP (DEG)	DELTAP (DEG)	PER RAD (P.RAD)	PER ALT (N.M.)	PER VEL (FT/SEC)	DELTA V (EMOS)	SUB-SOLAR PT. R.ASC (DEG)	SUB-SOLAR PT. DECLIN (DEG)
1.	143.80	-154.97	85.67	336.41	120.22	138.32	-158.81	-41.54	.0004	-1305.93	671452.42	-.0003	-152.67	.00

IMPULSIVE FLYBY CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI2 (DEG)	B.ANGLE (DEG)	ALTOPT (NM)	TR.OPT (PL.RAD)	DVOPT (FPS)	GAMMV (DEG)	THET1 (DEG)	THET2 (DEG)	VT-IN (FPS)	VT-OUT (FPS)
61.63	4.36	22.55	88.55	23801.27	8.1691	63414.2	73.96	36.80	102.06	82671.8	31999.5

INCLIN (DEG)	C.OMEGA (DEG)	OMEGA 1 (DEG)	OMEGA 2 (DEG)	OMEGA T (DEG)	ALPHA T (DEG)	DELTA T (DEG)
146.90	-66.72	292.40	20.95	169.96	121.71	5.46

FLYBY CHARACTERISTICS

PLANET	BEND ANG (DEG)	OMEGA (DEG)	INCL (DEG)	OMEGA1 (DEG)	OMEGA2 (DEG)	OMEGAP (DEG)	ALPHAP (DEG)	DELTAP (DEG)	PER RAD (P.RAD)	PER ALT (N.M.)	PER VEL (FT/SEC)	DELTA V (EMOS)	SUB-SOLAR PT. R.ASC (DEG)	SUB-SOLAR PT. DECLIN (DEG)
4.	55.47	142.42	98.50	14.74	70.20	132.47	151.60	-46.85	.1999	-1472.10	46184.63	-.0001	-11.96	-5.27

PLANE CHANGE MANEUVER

B.ANGLE (DEG)	ALPHA V (DEG)	DECLIN V (DEG)	DEL.INC. (DEG)	DELTAV (EMOS)
20.52	-108.06	-1.82	19.88	.4710

ARRIVAL CHARACTERISTICS

DEL.PSI (DEG)	PSI 1 (DEG)	PSI 2 (DEG)	B.ANGLE (DEG)	P.ALT (NM)	V-CIRC (FPS)	VP-ELL (FPS)	VP-HY 1 (FPS)	VP-HY 2 (FPS)	DV-C 1 (FPS)	DV-C 2 (FPS)	DVC-TOT (FPS)	DV-E 1 (FPS)	DV-E 2 (FPS)	DVE-TOT (FPS)	PER. RAD. (PL.RAD.)
-.00	15.01	15.01	30.02	100.00	25579.8	25579.8	56398.9	56398.9	30819.0	30819.0	61638.1	30819.0	30819.0	61638.1	1.0291

INCLIN C.OMEGA OMEGA 1 OMEGA 2 OMEGA P ALPHA P DELTA P A/RP SMA.ELL APO-VEL

(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(DEG)	(N.DIM)	(NM)	(FPS)
19.94	-110.51	139.98	170.00	244.99	313.10	18.00	1.0000	3538.	25579.8

APPENDIX H

Program Listings

```
   INTAP.USRSPC,USRSPC
BY UNIVAC 1108 PDP ON 30 OCT 69 AT 15:00:28 1102-0007


RUNNL          PROC
C
C       NAMELIST RUNDAT - THESE VARIABLES ARE RELEVANT TO THE RUN BEING
C       MADE AND ARE NOT SAVED ON THE SAVE FILES
C
               NAMELIST/RUNDAT/KON,ISM,ICM,ISAI,ISAF,ISAC,ISAA,IPRI,
     .         IPRF,IPRC,IPRA,IPRP,IPRINT,IUNIT,IREAD,NFR,ICHSA,
     .         NF14,NF15,NF16,NF17,NF18,NAUX
 END
RUNDOC         PROC
C
C              NAMELIST RUNDAT      -
C
C              NAME     TYPE/SIZE   DEFINITION
C
C              KON       INT/3      FUNCTION SEQUENCE INDICATOR
C              ISM       INT/       SEARCH METHOD SELECTOR
C              ICM       INT/       ITERATIVE METHOD SELECTOR
C              ISAI      INT/       FLAG,=1 SAVE INISOL OUTPUT
C              ISAF      INT/       FLAG,=1 SAVE FILGRD OUTPUT
C              ISAC      INT/       FLAG,=1 SAVE COMCHA OUTPUT
C              ISAA      INT/       FLAG,=1 SAVE AUXPRO OUTPUT
C              IPRI      INT/       PRINT LEVEL CONTROL FOR INISOL
C              IPRF      INT/       PRINT LEVEL CONTROL FOR FILGRD
C              IPRC      INT/       PRINT LEVEL CONTROL FOR COMCHA
C              IPRA      INT/       PRINT LEVEL CONTROL FOR AUXPRO
C              IPRP      INT/       PRINT LEVEL CONTROL FOR PLTCON
C              IPRINT   TYPE OUTPUT FROM CHARAC =0 CONDENSED, =1 FULL
C              IUNIT     INT/       FILE FROM WHICH SAVDAT IS READ
C              IREAD    =1 IF SPLANDT DATA FILE IS TO BE READ
C              NFR       INT/       NUMBER OF FRAMES TO BE PLOTTED
C              ICHSA     INT/       FLAG,=1 AFTER READING SAVDAT FROM
C                                   IUNIT,READ(5,SAVDAT)
C              NFI14     INT/       FLAG,=1 READ PAST SAVDAT ON UNIT 14
C              NFI15     INT/       FLAG,=1 READ PAST SAVDAT ON UNIT 15
C              NFI16     INT/       FLAG,=1 READ PAST SAVDAT ON UNIT 16
C              NFI17     INT/       FLAG,=1 READ PAST SAVDAT ON UNIT 17
C              NFI18     INT/       FLAG,=1 READ PAST SAVDAT ON UNIT 18
 END
SAVNL          PROC
C
C       NAMELIST SAVDAT - THESE VARIABLES DEFINE THE SIMULATOR AND THE
C       ANALYSIS PERFORMED. THIS NAMELIST IS SAVED ON ALL SAVE FILES
C
               NAMELIST/SAVDAT/ISGR,ICCH,IOPT,TARV,IP1,IP2,EPS,TPD,IENC,
     .         P1I,P2I,NCS,ICS,PSGCON,PERDEC,ARATS,NSG,NCG,NPUSR,
     .         NCUSR,ICM,ISM,LENGTH,LNENCR,LNLBRT,NPONTS,LEGS
     .         ,DELSTP
 END
SAVDOC         PROC
C
C              NAMELIST SAVDAT
```

```
C
C              NAME     TYPE/SIZE    DEFINITION
C
C              NP        INT/        NUMBER OF INDEPENDENT PARAMETERS
C              NC        INT/        NUMBER OF SYSTEM CHARACTERISTICS
C              A         REA/MP*     LOWER BOUNDS FOR PARAMETERS
C              B         REA/MP*     UPPER BOUNDS FOR PARAMETERS
C              P1I       REA/        INITIAL VALUE OF GRID PARAMETER 1
C              P2I       REA/        INITIAL VALUE OF GRID PARAMETER 2
C              DELP      REA/MP*     STEP SIZE FOR PARAMETERS
C              NCS       INT/        NUMBER OF EQUALITY CONSTRAINTS
C              NTS       INT/        =NCS+1 IF OPTIMIZING,=NCS OTHERWISE
C              EPS       INT/MC*     TOLERANCE ON CONSTRAINTS
C              NSG       INT/        NUMBER OF SOLUTIONS TO BE DEVELOPED
C              ISGR      INT/MG*     ISGR(I)=J,JTH SOLUTION USED TO
C                                    START THE ITH GRID
C              NCG       INT/        NUMBER OF GRIDS FOR WHICH
C                                    CHARACTERISTICS ARE TO BE COMPUTED
C              ICCH      INT/MH*     ICCH(I)=J,USE THE GRID DEVELOPED FROM
C                                    THE JTH SOLUTION TO COMPUTE THE ITH
C                                    CHARACTERISTIC SET
C              IFS       INT/5       INDICATES WHICH FUNCTIONS TO USE
C                                    AS CONSTRAINTS
C              TV        REA/5       TARGET VALUE FOR THE FUNCTIONS
C
C              * SEE PARAMETER STATEMENT FOR NUMERICAL VALUE
C
 END
OPEMPC         PROC
 END
OPEMAP         PROC
 END
KOMUSE         PROC
C
C       COMMON USE PROVIDES AN INTERFACE BETWEEN ALL USER SUBROUTINES
C       AND PAP
C
               COMMON/USER/TPD(13,MP),TPA(5,MP),TARV(MC),PLCVEC(3,MP),
     .         VELCTY(3,MP),RELVEC(3,MP),PSTION(3,MP),NPONTS,LEGS,NCUSR,
     .         NPUSR,IENC(30),IP1,IP1TP,IP1VAR,IP2,IP2TP,IP2VAR,IPRINT,
     .         IOPT,IOEN,IOTY,NCSUSR,ICS(MC),ICPREX(MC),ICEN(MC),
     .         ICTY(MC),ITP(MP),ITY(MP),ELPSD(10,MP),ENCHA(6,MP),
     .         PSGCON(30),PERDEC(30),ARATS(30),LENGTH(30),LNENCR(30),
     .         LNLBRT(30),DELSTP(30),IABORT
 END
USEDOC         PROC
C
C       SPECIFICATION AND DESCRIPTION OF LABELLED COMMON USER
C
C
C       ...NAME          DEFINITION ('*' INDICATES AN INPUT ITEM)
C
C         *TPD(I,MP)     TIME POINT DELIMITERS
C          TPA(I,MP)     TIME POINT ARRAY
C         *TARV(I)       TARGET VALUE OF ITH CONSTRAINT
C          PLCVEC(I,MP)  COMPONENTS OF VECTOR TO PLANET'S CENTER
C          VELCTY(I,MP)  COMPONENTS OF VELOCITY OF PLANET'S CENTER
```

```
C          RELVEC(I,MP) COMPONENTS OF TIME POIN RELATIVE TO PLANET CENTER
C          PSTION(I,MP) TOTAL HELIOCENTRIC TIME POINT VECTOR
C          NPONTS       TOTAL NUMBER OF TIME POINTS
C          LEGS         NUMBER OF LEGS
C          NPUSR        NUMBER OF PARAMETERS
C         *IENC(I)      TYPE OF ENCOUNTER AT ITH TIME POINT
C         *IABC=IJ      TWO DIGIT NUMBER BROKEN INTO IABP,IABV
C          IABP=I       ABSCISSA VARIABLE IS FROM TIME POINT I
C          IABV=J       ABSCISSA VARIABLE IS TPA(J,I)
C         *IORD=KL      TWO DIGIT NUMBER BROKEN INTO IORP,IORV
C          IORP=K       ORDINATE VARIABLE IS FROM TIME POINT K
C          IORV=L       ORDINATE VARIABLE IS TPA(L,K)
C         *IPRINT       =0 FOR SUMMARY PRINT, .GT.0 FOR EXPANDED PRINT
C         *IOPT=0       IF NO OPTIMIZATION IS TO OCCUR
C         *IOPT=MN      TWO DIGIT NUMBER BROKEN INTO IOEN,IOTY
C          IOEN=M       FUNCTION TO BE OPTIMIZED IS FROM TIME POINT 'M'
C          IOTY=N       FUNCTION TO BE OPTIMIZED IS TYPE 'N'
C          NCSUSR       NUMBER OF EQUALITY CONSTRAINTS
C         *ICS(I)=JKL   THREE DIGIT NUMBER BROKEN INTO ICPRFX(I),
C                       ICEN(I),ICTY(I)
C          ICPRFX(I)=0  ITH CONSTRAINT IS AT AN ENCOUNTER
C          ICPRFX(I)=1  ITH CONSTRAINT IS ON A LAMBERT LEG
C          ICEN(I)=K    ITH EQUALITY CONSTRAINT IS EITHER AT TIME POINT
C                       'K' OR ON LEG CONNECTING POINTS 'K' AND 'K+1'
C          ICTY(I)=L    ITH CONSTRAINT IS TYPE 'L' WHERE;
C                            L=1 REFERS TO DELTAV
C                            L=5 REFERS TO SEMI-MAJOR AXIS
C                            L=6 REFERS TO ECCENTRICITY
C          ITP(I)=K     ITH PARAMETER IS FROM TIME-POINT 'K'
C          ITY(I)=J     ITH PARAMETER IS TYPE 'J' WHERE;
C                            J=2 REFERS TO TIME
C                            J=3 REFERS TO RADIUS
C                            J=4 REFERS TO RIGHT ASCENSION
C                            J=5 REFERS TO DECLINATION
C          ELPSD(I,J)   VALUE OF ITH LAMBERT VARIABLE CONNECTING TIME
C                       POINTS 'J' AND 'J+1'
C          ENCHA(I,J)   VALUE OF ITH ENCOUNTER VARIABLE FROM TIME POINT J
C         *PSGCON(I)    REQUIRED PASSAGE ALTITUDE OR RADIUS AT POINT 'I'
C         *PERDEC(I)    REQUIRED PERIAPSE DECLINATION AT TIME POINT 'I'
C         *ARATS(I)     RATIO OF SEMI-MAJOR AXIS OF ELLIPTIC PARKING
C                       ORBIT TO CIRCULAR PARKING ORBIT AT TIME POINT 'I'
C          LENGTH(MP)   NUMBER OF CHARACTERISTICS COMPUTED PER POINT.
C          LNENCR(MP)   NUMBER OF ENCOUNTER CHARACTERISTICS PER POINT
C          LNLBRT(MP)   NUMBER OF LAMBERT CHARACTERISTICS PER SEGMENT
C          DELSTP(I)    STOPOVER DURATION AT ITH PLANET
C          IABORT       =1 IF ABORT TRAJECTORY IS TO BE COMPUTED
C
C       ...THIS COMMON IS INCLUDED IN SUBROUTINES CHARAC,CONSEG,CONSTR,
C          ENCTR,ENCONT,FIXPLC,FIXREL,PRTCHA,READUI,SETUP
C
 END
KOMPEN        PROC
              COMMON/PENCTR/NPLNO,TIMEIN,VIN,RASIN,DECLIN,TIMOUT,VOUT,
     .        RASOUT,DECOUT,BA,RASCSP,DECLSP,PLINC,PLINC1,PLINC2,
     .        POMEGA,OMEGA1,OMEGA2,OMEGT1,OMEGT2,OMEGAP,ALPH1,ALPH2,
     .        ALPHP,DECL1,DECL2,DECLP,DELPSI,PSI1,PSI2,DUMMY1(10),
     .        ALTTR1,RADTR1,ALTTR2,RADTR2,DELVBL,VELPER,DELVT1,DELVT2,
```

```
     .          GAMT1,GAMT2,THE1T1,THE2T1,THE1T2,THE2T2,VTINT1,VTOUT1,
     .          VTINT2,VTOUT2,PALT,PRAD,ARATIO,AENM,VAPO,VELC,VELPE,
     .          VELPH1,VELPH2,DELVC1,DELVC2,SUMVC,DELVE1,DELVE2,SUMVE,
     .          DINC
 END
 PENDOC         PROC
C
C     SPECIFICATION AND DESCRIPTION OF LABELLED COMMON PENCTR
C
C
C    ...NAME         DEFINITION
C       NPLNO        PLANET NUMBER
C       TIMEIN       TIME OF ARRIVAL AT ENCOUNTER
C       VIN          MAGNITUDE OF HYPERBOLIC EXCESS VELOCITY
C                    ON INBOUND LEG
C       RASIN        RIGHT ASCENSION OF HYPERBOLIC EXCESS VELOCITY
C                    ON INBOUND LEG
C       DECLIN       DECLINATION OF HYPERBOLIC EXCESS VELOCITY VECTOR
C                    ON INBOUND LEG
C       TIMOUT       TIME OF DEPARTURE FROM ENCOUNTER
C       VOUT         SAME AS ABOVE OUTBOUND LEG
C       RASOUT       SAME AS ABOVE OUTBOUND LEG
C       DECOUT       SAME AS ABOVE OUTBOUND LEG
C       BA           ANGLE BETWEEN INBOUND AND OUTBOUND ASYMPTOTES
C       RASCSP       RIGHT ASCENSION OF SUB-SOLAR POINT
C       DECLSP       DECLINATION OF SUB-SOLAR POINT
C       PLINC        INCLINATION OF PASSAGE ORBIT TO PLANET'S EQUATOR
C       PLINC1       INCLINATION OF FIRST SEGMENT OF TRANSFER ORBIT
C                    TO PLANET'S EQUATOR
C       PLINC2       INCLINATION OF SECOND SEGMENT OF TRANSFER ORBIT
C                    TO PLANET'S EQUATOR
C       POMEGA       LONGITUDE OF ASCENCING NODE THROUGH PLANET EQATR.
C       OMEGA1       ANGLE FROM ASCENDING NODE TO INBOUND ASYMPTOTE
C       OMEGA2       ANGLE FROM ASCENDING NODE TO OUTBOUND ASYMPTOTE
C       OMEGT1       ANGLE FROM ASCENDING NODE TO FIRST TRANSFER
C       OMEGT2       ANGLE FROM ASCENDING NODE TO SECOND TRANSFER
C       OMEGAP       ANGLE FROM ASCENDING NODE TO PERIAPSE
C       ALPH1        LONGITUDE OF FIRST TRANSFER POINT
C       ALPH2        LONGITUDE OF SECOND TRANSFER POINT
C       ALPHP        LONGITUDE OF PERIAPSE
C       DECL1        LATITUDE OF FIRST TRANSFER POINT
C       DECL2        LATITUDE OF SECOND TRANSFER POINT
C       DECLP        LATITUDE OF PERIAPSE
C       DELPSI       EXCESS IN NATURAL BEND ANGLE
C       PSI1         NATURAL BEND ON INBOUND HYPERBOLA
C       PSI2         NATURAL BEND ON OUTBOUND HYPERBOLA
C       DUMMY1(10)   UNASSIGNED AS YET
C       ALTTR1       ALTITUDE OF FIRST TRANSFER
C       RADTR1       RADIUS OF FIRST TRANSFER
C       ALTTR2       ALTITUDE OF SECOND TRANSFER
C       RADTR2       RADIUS OF SECOND TRANSFER
C       DELVBL       VIN-VOUT MAGNITUDE
C       VELPER       VELOCITY AT PERIAPSE
C       DELVT1       IMPULSE REQUIRED FOR FIRST TRANSFER
C       DELVT2       IMPULSE REQUIRED FOR SECOND TRANSFER
C       GAMT1        ANGLE BETWEEN HORIZONTAL AND DELTAV VECTOR AT
C                    FIRST TRANSFER
```

```
C         GAMT2          ANGLE BETWEEN HORIZONTAL AND DELTAV VECTOR AT
C                        SECOND TRANSFER
C         THE1T1         TRUE ANOMALY ON INBD. HYPERBOLA OF FIRST TRANS.
C         THE2T1         TRUE ANOMALY ON OTBD. HYPERBOLA OF FIRST TRANS.
C         THE1T2         TRUE ANOMALY ON INBD. HYPERBOLA OF SECOND TRANS.
C         THE2T2         TRUE ANOMALY ON OTBD. HYPERBOLA OF SECOND TRANS.
C         VTINT1         VELOCITY ON INBD. HYPERBOLA AT FIRST TRANSFER
C         VTOUT1         VELOCITY ON OTBD. HYPERBOLA AT FIRST TRANSFER
C         VTINT2         VELOCITY ON INBD. HYPERBOLA AT SECOND TRANSFER
C         VTOUT2         VELOCITY ON OTBD. HYPERBOLA AT SECOND TRANSFER
C         PALT           PASSAGE ALTITUDE (MINIMUM)
C         PRAD           PASSAGE RADIUS (MINIMUM)
C         ARATIO         RATIO OF ELLIPTICAL PARKING ORBIT SEMI-MAJOR AXIS
C                        TO CIRCULAR PARKING ORBIT
C         AENM           SEMI-MAJOR AXIS OF ELLIPTICAL PARKING ORBIT
C         VAPO           VELOCITY AT APOAPSE ON PARKING ORBIT
C         VELC           CIRCULAR VELOCITY AT MINIMUM RADIUS
C         VELPE          VELOCITY AT PERIAPSE ON ELLIPTIC PARKING ORBIT
C         VELPH1         VELOCITY AT PERIAPSE ON INBD. HYPERBOLA
C         VELPH2         VELOCITY AT PERIAPSE ON OTBD. HYPERBOLA
C         DELVC1         VELOCITY REQUIRED TO DEBOOST FROM INBD. TO CIRC.
C         DELVC2         VELOCITY REQUIRED TO BOOST FROM CIRC. TO OTBND.
C         SUMVC          TOTAL DELTAV REQUIRED FOR CIRCULAR PARKING ORBIT
C         DELVE1         VELOCITY REQUIRED TO DEBOOST FROM INBD. TO ELLIP.
C         DELVE2         VELOCITY REQUIRED TO BOOST FROM ELLIP. TO OUTBD.
C         SUMVE          TOTAL DELTAV REQUITRED FOR ELLIP. PARKING ORBIT
C         DINC           INCLINATION BETWEEN INBOUND AND OUTBOUND PLANES
C                        FOR PLANE CHANGE MANEUVER.
C
C      ...THIS COMMON IS INCLUDED IN SUBROUTINES ENCONT,LODCHA,
C         OPTRAN,SUBSOL,VASTOT
C
 END
KOMPLN          PROC
                COMMON/PLANET/AMU(20),RHO(20),SMA(20),ECC(20),TPHPAS(20),
     .          FINC(20),OMEGA(20),PERMIN(20),VEFRPH(20),EQATOR(20)
 END
PLNDOC          PROC
C
C       SPECIFICATION AND DESCRIPTION OF LABELLED COMMON PLANET
C
C      ...NAME           DEFINITION
C
C         AMU(I)         GRAVITATIONAL CONSTANT (FT**3/SEC**2)
C         RHO(I)         PLANETARY RADIUS (NM)
C         SMA(I)         SEMI-MAJOR AXIS OF PLANET'S ORBIT (AU)
C         ECC(I)         ECCENTRICITY OF PLANET'S ORBIT
C         TPHPAS(I)      TIME OF PERIHELION PASSAGE (JULIAN DATE-2400000.)
C         FINC(I)        INCLINATION OF PLANET'S ORBIT TO ECLIPTIC (DEG)
C         OMEGA(I)       ANGLE FROM 'FIRST POINT OF ARIES' TO PLANET'S
C                        ASCENDING NODE THROUGH ECLIPTIC (DEG)
C         PERMIN(I)      ANGLE FROM ASCENDING NODE THROUGH ECLIPTIC TO
C                        PLANET'S PERIHELION (DEG)
C         VEFRPH(I)      ANGLE FROM PLANET'S PERIHELION TO PLANET'S
C                        VERNAL EQUINOX (DEG)
C         EQATOR(I)      INCLINATION OF PLANET'S EQUATOR TO ORBITAL
C                        PLANE (DEG)
```

```
C
C       ...THIS COMMON IS INCLUDED IN SUBROUTINES CONSEG,CONST,ENCONT,
C          EQRTR,FIXPLC,FIXREL,OPTRAN,READUI,SUBSOL,VASTOT
C
 END
INITIL            PROC
C
                  COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                  COMMON/AUXPLT/IPAUX
                  DIMENSION KM(5)
C
 1                IUNIT=5
                  ICHSA=0
                  NFR=0
                  IPAUX=0
                  IREAD=0
                  IPRINT=0
                  ISM=1
                  ICM=1
                  IPRI=1
                  IPRF=1
                  IPRC=1
                  IPRA=1
                  IPRP=1
                  P1I=0.
                  P2I=0.
                  IOPT=0
                  NSG=1
                  NCG=1
                  IABORT=0
                  DO 11 IJ=1,MP
                  DO 11 IK=1,13
 11               TPD(IK,IJ)=0.
                  DO 12 IJ=1,MC
 12               EPS(IJ)=.0001
                  DO 13 IJ=1,30
 13               DELSTP(IJ)=0.
 END
RPDOS             PROC
                  DIMENSION ISGR(MG),ICCH(MH),TARV(MC),EPS(MC),
         .        TPD(13,MP),IENC(30),ICS(MC),PSGCON(MP),PERDEC(MP),
         .        ARATS(MP),LENGTH(30),LNENCR(30),LNLBRT(30),DELSTP(30)
 END
SIMDFN            PROC
C
C       DEFINE SIMULATOR
C
                  CALL CONST(IREAD)
C
C       CONVERT INPUTS FROM DEGREES TO RADIANS
C
                  DO 31 IJ=1,MP
                  PERDEC(IJ)=PERDEC(IJ)*DTR
                  DO 31 IK=8,13
                  TPD(IK,IJ)=TPD(IK,IJ)*DTR
 31               CONTINUE
C
```

```
C        DETERMINE NUMBER OF TIME-POINTS AND NUMBER OF LEGS.
C
               DO 32 IJ=1,11
               IF(TPD(1,IJ).LT.(10.)) GO TO 33
 32            CONTINUE
 33            NPONTS=IJ-1
               LEGS=NPONTS-1
C
C        IS THIS AN ABORT RUN?
C
               IF(NAUX.EQ.2) IABORT=1
C
C        DETERMINE TOTAL NUMBER OF CHARACTERISTICS FOR THIS SEGMENT
C
               NC=0
               DO 36 I=1,NPONTS
               IENTYP=IENC(I)
               N1=TPD(1,I)+.01
               IF(I.LT.NPONTS) N2=TPD(1,I+1)+.01
               LNLBRT(I)=10
               IF(((N1.EQ.N2).AND.(N1.NE.0)).OR.(I.EQ.NPONTS))
     .                 LNLBRT(I)=0
               IF(IENTYP.NE.0) GO TO 34
               LNENCR(I)=0
               GO TO 35
 34            IF((IENTYP.EQ.1).OR.(IENTYP.EQ.8).OR.(IENTYP.EQ.6))
     .                                         LNENCR(I)=37
               IF((IENTYP.EQ.4).OR.(IENTYP.EQ.5)) LNENCR(I)=0
               IF(IENTYP.EQ.2) LNENCR(I)=23
               IF(IENTYP.EQ.3) LNENCR(I)=30
               IF(IENTYP.EQ.7) LNENCR(I)=16
               NC=NC+LNENCR(I)+LNLBRT(I)
 35            LENGTH(I)=LNLBRT(I)+LNENCR(I)
 36            CONTINUE
               IF(IOPT.GT.0) NC=NC+1
               NCUSR=NC
C
C        DETERMINE WHICH VARIABLES AND AT WHAT TIME POINTS ARE TO
C        BE GRID PARAMETERS
C
               IP1TP=IP1/10
               IP1VAR=IP1-10*IP1TP
               IP2TP=IP2/10
               IP2VAR=IP2-10*IP2TP
C
C        SCAN POINTS TO SET UP PARAMETER TYPES AND NUMBERS AND SET
C        FIXED TIME POINTS INTO COMMON TPA. ALSO DETERMINE NUMBER
C        OF PARAMETERS.
C
               NP=0
               IL=3
               DO 40 IJ=1,NPONTS
               NOVARY=0
               IFIXT=1
               DO 39 IK=4,13,3
               IF(ABS(TPD(IK,IJ)).LT.(.001)) GO TO 37
               ITP(NP+1)=IJ
```

```
      ITY(NP+1)=IK/3+1
      NF=NP+1
      NFUSR=NF
C
C     ...SET UPPER AND LOWER BOUNDS AND INCREMENTS FOR NON-GRID VARS.
C
      IF((IJ.EQ.IP1TP).OR.(IJ.EQ.IP2TP)) GO TO 39
      A(IL)=TPD(IK-2,IJ)
      B(IL)=TPD(IK-1,IJ)
      DELP(IL)=TPD(IK,IJ)
      IL=IL+1
      GO TO 39
   37 NOVARY=NOVARY+1
      IF(IK.GT. 4) GO TO 38
      TPA(2,IJ)=TPD(2,IJ)
      T=TPA(2,IJ)
      CALL FIXPLC(IJ,T)
      IFIXT=0
   38 IF(IK.EQ. 7) TPA(3,IJ)=TPD(5,IJ)
      IF(IK.EQ.10) TPA(4,IJ)=TPD( 8,IJ)
      IF(IK.EQ.13) TPA(5,IJ)=TPD(11,IJ)
      IF((NOVARY.NE.4).AND.((NOVARY+IFIXT).NE.4)) GO TO 39
      CALL FIXREL(IJ)
   39 CONTINUE
      TPA(1,IJ)=TPD(1,IJ)
   40 CONTINUE
C
C     SET UP LOWER AND UPPER BOUNDS AND INCREMENTS ON GRID PARAMETERS
C     AND INITIAL GUESS POINT.
C
      B(1)=TPD(3*IP1VAR-3,IP1TP)
      A(1)=TPD(3*IP1VAR-4,IP1TP)
      DELP(1)=TPD(3*IP1VAR-2,IP1TP)
      A(2)=TPD(3*IP2VAR-4,IP2TP)
      B(2)=TPD(3*IP2VAR-3,IP2TP)
      DELP(2)=TPD(3*IP2VAR-2,IP2TP)
      IF(P1I.LT.(.01)) P1I=A(1)
      IF(P2I.LT.(.01)) P2I=A(2)
C
C     SET UP INFORMATION REGARDING EQUALITY CONSTRAINTS
C
      NCSUSR=NCS
      DO 41 IJ=1,NCS
      ICPRFX(IJ)=ICS(IJ)/100
      ICEN(IJ)=(ICS(IJ)-100*ICPRFX(IJ))/10
      ICTY(IJ)=ICS(IJ)-10*ICEN(IJ)-100*ICPRFX(IJ)
   41 CONTINUE
C
C     SET UP INFORMATION REGARDING OPTIMIZATION
C
      NTC=NCS
      IF(IOPT.EQ.0) GO TO 42
      IOEN=IOPT/10
      IOTY=IOPT-10*IOEN
      NTC=NCS+1
C
C     CHECK TO SEE THAT PROBLEM IS NOT OVERCONSTRAINED
```

```
C
 42               IF(NTC.LT.(NP-2)) GO TO 43
                  WRITE(6,1004)
 1004             FORMAT(1H ,9X,'TOO MANY CONSTRAINTS, PROGRAM STOP')
                  CALL EXIT
C
C        COMPUTE LOCATION WITHIN RECORD OF PLOT VARIABLES.
C
 43               DO 49 I=1,NF
                  JMAX=NP(I)
                  DO 48 J=1,JMAX
                  IF(IVAR(I,J).GT.0) GO TO 430
                  IVAR(I,J)=-IVAR(I,J)
                  GO TO 48
 430              DO 44 K=1,5
 44               KW(K)=IVAR(I,J)/10**(K-1)-10*(IVAR(I,J)/10**K)
                  IF(KW(5).EQ.0) GO TO 45
                  IFAUX=1
                  GO TO 48
 45               LNGTH=0
                  LHVAR=IVAR(I,J)/100-1
                  IRHVAR=IVAR(I,J)-100*(LHVAR+1)
                  IF(LHVAR.EQ.0) GO TO 47
                  DO 46 K=1,LHVAR
 46               LNGTH=LNGTH+LENGTH(K)
 47               IVAR(I,J)=LNGTH+IRHVAR+NPUSR-2
                  IF(IOPT.GT.0.AND.IVAR(I,J).GT.0) IVAR(I,J)=IVAR(I,J)+1
 48               CONTINUE
 49               CONTINUE
 END
```

```
INTAP.AUXPRO
LEVEL      3
          SUBROUTINE AUXPRO(NAUX)
C
          GO TO (1,2), NAUX
1         CALL AUX1
          RETURN
2         CALL AUX2
          RETURN
          END
```

```
INTAP.AUX1
LEVEL      3
   C
   C  TITLE            AUXILIARY PROGRAM
   C
   C  AUTHOR           [illegible]
   C
   C  SPONSOR          A.A.VANDERVEEN
   C
   C  DATE             5-8-68
   C
   C  PURPOSE          CONNECT SEGMENTS AND PRINT SAME. GENERATE PLOT FILES.
   C
   C        SUBROUTINE STATEMENT
   C
                       SUBROUTINE AUX1
   C
   C
   C        SPECIFICATION STATEMENTS
   C
                       INCLUDE PARSTA,LIST
                       INCLUDE KOMMCL,LIST
                       INCLUDE KOMUSE,LIST
                       INCLUDE SAVNL,LIST
                       COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                       COMMON/AUXPLT/IPAUX
                       NAMELIST/CONECT/IGV1,IGV2,KONECT,TFIXED,LIA1,LIA2,LIA3
        .              ,IPRINT
                       DIMENSION TFIXED(2),IENCA(MP,3),PSGA(MP,3),T(6),ICHK(2),
        .              ARATSA(MP,3),LENA(MP,3),LNA(MP,3),LNLBRA(MP,3),ISLN(3),
        .              ISEG(3),KN(6),NPA(3),NCA(3),NAA(3),NORA(3),IP1A(3),
        .              IP2A(3),LEGSA(3),P(MP,3),C(500,3),NPNTSA(3),NRECS(3),
        .              CHAR(50),CSTOP(50,2),NCST(3),CTOT(MK),NSTOP(2,3),
        .              IANPUT(3),PTOT(MP),ISTYPE(2),TMIN(3),TMAX(3),
        .              NLEP(3),NARIV(4),PERDA(MP,3)
   C
                       ICHSA=0
                       LIA1=0
                       LIA2=0
                       LIA3=0
                       IPLOT=1
                       READ(5,CONECT)
                       REWIND LIA1
                       IF(LIA2.EQ.0) GO TO 1
                       REWIND LIA2
                       IF(LIA3.GT.0) REWIND LIA3
   C
    1                  IF(IGV1.EQ.0) IPLOT=0
                       IF(IGV1.EQ.0) GO TO 4
                       IGV1L=IGV1/10
                       IGV1R=IGV1-10*IGV1L
                       IGV2L=IGV2/10
                       IGV2R=IGV2-10*IGV2L
                       IT1=MAX0((2*IGV1L+IGV1R-2),(2*IGV2L+IGV2R-2))
                       IT2=MIN0((2*IGV1L+IGV1R-2),(2*IGV2L+IGV2R-2))
   C
   C        DETERMINE WHICH POINTS ARE TO BE HELD CONSTANT IN GRID GENERATION
```

```
C
            I1=1
            DO 3 I=1,6
            IF(I.EQ.IT1.OR.I.EQ.IT2) GO TO 3
            IF(I.EQ.1) GO TO 2
            IF((I+1)/2.NE.(I/2)) GO TO 3
            IF((I+1).EQ.IT1.OR.(I+1).EQ.IT2) GO TO 3
            ICHK(I1)=1
            I1=I1+1
            GO TO 3
  2         ICHK(1)=1
            I1=I1+1
  3         CONTINUE
            IF(LTA2.EQ.0) GO TO 300
C
C      DECOMPOSE 'KONECT' AND STORE INTO ISLN AND ISEG
C
  4         JMAX=4
            IF(LTA3.NE.0) JMAX=6
            DO 5 J=1,JMAX
  5         KN(J)=KONECT/10**(J-1)-10*(KONECT/10**J)
            NGRIDS=3
            DO 6 J=1,5,2
  6         IF(KN(J).EQ.0) NGRIDS=2
            JM2=JMAX/2
            DO 7 J=1,JM2
            ICOUNT=JMAX-2*J+1
            ISEG(J)=KN(ICOUNT)
  7         ISLN(J)=KN(ICOUNT+1)
            IF(NGRIDS.EQ.2) ICNV3=1
C
C      READ USRDAT FROM COMCHA TAPES AND POSITION SAME AT PROPER SOLUTN.
C
            DO 14 J=1,NGRIDS
            I=ISEG(J)
            IFILE=13+I
C
C      ...READ DATA DESCRIBING HOW SEGMENT WAS GENERATED AND STORE.
C
            READ(IFILE,SAVDAT)
            DO 10 IMP=1,MP
            IENCA(IMP,I)=IENC(IMP)
            PSGA(IMP,I)=PSGCON(IMP)
            ARATSA(IMP,I)=ARATS(IMP)
            PERDA(IMP,I)=PERDEC(IMP)
            LENA(IMP,I)=LENGTH(IMP)
            LNA(IMP,I)=LNFNCR(IMP)
            LNLBRA(IMP,I)=LNLBRT(IMP)
 10         CONTINUE
            NPA(I)=NPUSR
            NCA(I)=NCUSR
            NPNTSA(I)=NPONTS
            LEGSA(I)=LEGS
            IP1A(I)=IP1
            IP2A(I)=IP2
C
C      ...READ SOLUTION IDENTIFICATION RECORD.
```

```
C
            DO 12 IJ=1,MG
            READ(IFILE) ISOLUT,NA,NO,NP,NC
C
C      ...IF THIS IS NOT THE DESIRED SOLUTION READ TO NEXT I.D. RECORD.
C
            IF(ISOLUT.EQ.ISLN(I)) GO TO 13
            NGRPTS=NA*NO
            DO 11 IL=1,NGRPTS
            READ(IFILE)ICONV,(P(II,1),II=1,NPUSR),
                                       (C(II,1),II=1,NCUSR)
 11         CONTINUE
 12         CONTINUE
 13         NAA(I)=NA
            NORA(I)=NO
 14         CONTINUE
C
C      ...IF USRDAT IS TO BE RESPECIFIED READ FILE 5
C
            IF(ICHSA.GT.0) READ(5,SAVDAT)
C
C      ...DEFINE OUTPUT FILE
C
            LOA1=17
            LOA2=18
C
C      DETERMINE WHICH VARIABLE WAS ABSCISSA AND ORDINATE AT TIME
C      FILE WAS ORIGINALLY GENERATED.
C
            DO 20 II=1,NGPIDS
            IABPNT(II)=IP1A(II)/10
            NRECS(II)=NORA(II)
 20         IF(IABPNT(II).NE.1) NRECS(II)=1
C
C      SET UP 'DO LOOP' INDICES.
C
            NDEP1=NAA(1)
            IF(IABPNT(1).NE.1) NDEP1=NORA(1)
            NARIV2=NORA(1)
            IF(IABPNT(1).NE.1) NARIV2=NAA(1)
            NDEP2=NARIV2
            NARIV3=NORA(2)
            IF(IABPNT(2).NE.1) NARIV3=NAA(2)
            NDEP3=NARIV3
            IF(IABPNT(3).NE.1) NARIV4=NAA(3)
            NARIV(2)=NARIV2
            NARIV(3)=NARIV3
            NARIV(4)=NARIV4
            NDEP(1)=NDEP1
            NDEP(2)=NDEP2
            NDEP(3)=NDEP3
            NP1=NPA(1)
            NP2=NPA(2)
            NP3=NPA(3)
            NC1=NCA(1)
            NC2=NCA(2)
            NC3=NCA(3)
```

```
      NARIV4=NORA(3)
      NPNT1=NAA(1)*NORA(1)
      NPNT2=NAA(2)*NORA(2)
      NPNT3=NAA(3)*NORA(3)
C
C     ...SET UP THE END POINT LOGIC
C
      NSTOP(1,1)=0
      NSTOP(2,1)=NPNTSA(1)
      NSTOP(1,2)=1
      NSTOP(2,2)=0
      IF(NGRIDS.EQ.2) GO TO 29
      NSTOP(2,2)=NPNTSA(2)
      NSTOP(1,3)=1
      NSTOP(2,3)=0
C
C     COMPUTE TOTAL NUMBER OF 'PARAMETER ONE' AND 'PARAMETER TWO'
C
   29 IF(IGV1.EQ.0) GO TO 30
      ISUB1=IT1/2+1
      NPAR1=NARIV(ISUB1)
      ISUB2=IT2/2+1
      NPAR2=NDEP(ISUB2)
      NRT=NDEP1*NARIV2*NARIV3
      IF(NGRIDS.EQ.3) NRT=NRT*NARIV4
C
C     COMPUTE TOTAL NUMBER OF PARAMETERS AND CHARACTERISTICS, SET
C     RECORD LENGTHS AND REDEFINE ENCOUNTERS TO MAKE TRIP CONTINUOUS
C
   30 DO 31 IK=1,NGRIDS
      ITAPE=13+IK
      READ(ITAPE) ICNV,P(1,IK),P(2,IK)
      BACKSPACE ITAPE
      TMAX(IK)=AMAX1(P(1,IK),P(2,IK))
   31 TMIN(IK)=AMIN1(P(1,IK),P(2,IK))
      NGM1=NGRIDS-1
      DO 32 IK=1,NGM1
      ISTYPE(IK)=6
      NCST(IK)=37
      IF(ABS(TMAX(IK)-TMIN(IK+1)).GT.(.01)) GO TO 32
      ISTYPE(IK)=3
      NCST(IK)=30
   32 CONTINUE
C
      LEGS=0
      NPONTS=0
      JSUM=0
      NCTOT=0
      NCST(3)=0
      NPT1=NPNTSA(1)
      NPT2=NPNTSA(2)
      DO 34 I1=1,NGRIDS
      I2MAX=NPNTSA(I1)
      I2MIN=1
      IF(I1.GT.1) I2MIN=2
      DO 33 I2=I2MIN,I2MAX
      IENC(JSUM+I2)=IENCA(I2,I1)
```

```
      LENGTH(JSUM+I2)=LENA(I2,I1)
      LNENCR(JSUM+I2)=LNA(I2,I1)
 33   LNLBRT(JSUM+I2)=LNLBRA(I2,I1)
      JSUM=JSUM+I2MAX-1
      NCTOT=NCTOT+NCA(I1)+NCST(I1)
      LEGS=LEGS+LEGSA(I1)
      NPONTS=NPONTS+NPNTSA(I1)
 34   CONTINUE
      NPONTS=NPONTS+1-NGRIDS
      NCTOT=NCTOT-LNA(NPT1,1)-LNA(1,2)
      IF(NGRIDS.GT.2) NCTOT=NCTOT-LNA(NPT2,2)-LNA(1,3)
      IENC(NPT1)=ISTYPE(1)
      IF(NGRIDS.GT.2) IENC(NPT1+NPT2-1)=ISTYPE(2)
      NGM1=NGRIDS-1
      JSUM=0
      DO 35 I1=1,NGM1
      I2=NPNTSA(I1)
      LNENCR(JSUM+I2)=NCST(I1)
      LNLBRT(JSUM+I2)=LNLBRA(1,I1+1)
      LENGTH(JSUM+I2)=LNENCR(JSUM+I2)+LNLBRT(JSUM+I2)
      JSUM=JSUM+NPNTSA(I1)-1
 35   CONTINUE
      NPTOT=2
C
C     SET IVAR FOR ANY VARIABLES TO BE PLOTTED FROM MULTI-GRID SEGMNTS.
C
      IF(IPAUX.EQ.0) GO TO 40
      DO 39 I=1,NFR
      JMAX=NPT(I)
      DO 38 J=1,JMAX
      IF(IVAR(I,J).LT.10000) GO TO 38
      IVAR(I,J)=IVAR(I,J)-10000
      LNGTH=0
      LHVAR=IVAR(I,J)/100-1
      IRHVAR=IVAR(I,J)-100*(LHVAR+1)
      IF(LHVAR.EQ.0) GO TO 37
      DO 36 K=1,LHVAR
 36   LNGTH=LNGTH+LENGTH(K)
 37   IVAR(J,I)=LNGTH+IRHVAR
      IF(IOPT.GT.0) IVAR(J,I)=IVAR(J,I)+1
 38   CONTINUE
 39   CONTINUE
C
C     WRITE I.D. RECORD ON PLOT FILE
C
 40   IF(IPLOT.NE.0) WRITE(LOA1)ISOLUT,NPAR1,NPAR2,NPTOT,NCTOT
C
C     WRITE I.D. RECORD ON OUTPUT FILE.
C
      NGP1=NGRIDS+1
      IF(ISAA.EQ.1) WRITE(LOA2)NGRIDS,(NDEP(I),I=1,NGRIDS),
     •                          (NARIV(I),I=2,NGP1),NRT,NCTOT
C
C     THIRD SEGMENT DATA
C
      IFRST3=1
      DO 200 II=1,NARIV4
```

```
              DO 100 IJ=1,NARIV3
              IF(LIA3.EQ.0) GO TO 50
              N1=NRECS(3)
              IF(IFRST3.EQ.1) N1=1
              DO 41 I2=1,N1
 41           READ(LIA3)ICNV3,(P(I1,3),I1=1,NP3),(C(I1,3),I1=1,NC3)
              T(5)=AMIN1(P(1,3),P(2,3))
              T(6)=AMAX1(P(1,3),P(2,3))
              IFRST3=0
C
C      SECOND SEGMENT DATA
C
 50           IFRST2=1
              DO 100 IK=1,NARIV2
              N1=NRECS(2)
              IF(IFRST2.EQ.1) N1=1
              DO 51 I2=1,N1
 51           READ(LIA2) ICNV2,(P(I1,2),I1=1,NP2),(C(I1,2),I1=1,NC2)
              T(5)=AMIN1(P(1,2),P(2,2))
              T(4)=AMAX1(P(1,2),P(2,2))
              IFRST2=0
C
C      FIRST SEGMENT DATA
C
              IFRST1=1
              DO 170 IL=1,NDEP1
              N1=NRECS(1)
              IF(IFRST1.EQ.1) N1=1
              DO 60 I1=1,N1
 60           READ(LIA1)ICNV1,(P(I2,1),I2=1,NP1),(C(I2,1),I2=1,NC1)
              T(1)=AMIN1(P(1,1),P(2,1))
              T(2)=AMAX1(P(1,1),P(2,1))
              IFRST1=0
C
C      IF ANY GRID POINT DID NOT CONVERGE DON'T COMPUTE STOPOVERS.
C
              ICVTOT=ICNV1*ICNV2*ICNV3
              IF(ICVTOT.EQ.0) GO TO 100
C
C      SET INPUTS AND CALL ENCONT TO COMPUTE STOPOVERS
C
              IG=1
 70           N=NPNTSA(IG)-1
              L=0
              DO 71 I1=1,N
 71           L=L+LPNA(I1,IG)
C
C      ...STORE INBOUND VECTOR
C
              ENCHA(1,1)=C(L+3,IG)
              ENCHA(2,1)=C(L+4,IG)*DTR
              ENCHA(3,1)=C(L+5,IG)*DTR
C
C      ...STORE OUTPUT VELOCITY VECTOR
C
              ENCHA(4,1)=C(7,IG+1)
              ENCHA(5,1)=C(8,IG+1)*DTR
```

```
      LNCHA(6,1)=C(9,IG+1)*DTR
      NPLNET=C(1,IG+1)+.1
      TPA(2,1)=C(L+2,IG)
      NPNT=NPNTSA(IG)
      IF(ABS(PSGA(NPNT,IG)-PSGA(1,IG+1)).LT.(.1)) GO TO 72
      WRITE(6,1000)
   72 PSGCON(IG)=AMAX1(PSGA(NPNT,IG),PSGA(1,IG+1))
      ARATS(IG)=AMAX1(ARATSA(NPNT,IG),ARATSA(1,IG+1))
      PERDEC(IG)=PERDA(1,IG+1)
      IF(ISTYPE(IG).EQ.6) GO TO 74
C
C     COMPUTE IMPULSIVE FLYBY CHARACTERISTICS
C
      CALL ENCTR(3,NPLNET,1,CHAR)
      DO 73 I1=1,30
   73 CSTOP(I1,IG)=CHAR(I1)
      IG=IG+1
      IF(IG.LT.NGRIDS.AND.IL.EQ.1) GO TO 70
      GO TO 76
C
C     COMPUTE DEBOOST STOPOVER CHARACTERISTICS
C
   74 CALL ENCTR(6,NPLNET,1,CHAR)
      DO 75 I1=1,37
   75 CSTOP(I1,IG)=CHAR(I1)
      IG=IG+1
      IF(IG.LT.NGRIDS.AND.IL.EQ.1) GO TO 70
C
C     REPLACE SEGMENT CHARACTERISTICS WITH TOTAL TRAJECTORY VECTOR.
C
   76 LN=0
      JSUM=0
      DO 82 I1=1,NGRIDS
      NPT=NPNTSA(I1)
      DO 81 I2=1,NPT
      IF(I2.EQ.NSTOP(1,I1).OR.I2.EQ.NSTOP(2,I1)) GO TO 78
      I3MAX=LENA(I2,I1)
      DO 77 I3=1,I3MAX
   77 CTOT(JSUM+I3)=C(LN+I3,I1)
      LN=LN+LENA(I2,I1)
      JSUM=JSUM+I3MAX
      GO TO 81
   78 IF(I2.NE.1) GO TO 82
      I3MAX=NCST(I1-1)
      DO 79 I3=1,I3MAX
   79 CTOT(JSUM+I3)=CSTOP(I3,I1-1)
      JSUM=JSUM+I3MAX
      LN=LNA(1,I1)
      I3MAX=LNLBRA(1,I1)
      DO 80 I3=1,I3MAX
   80 CTOT(JSUM+I3)=C(LN+I3,I1)
      JSUM=JSUM+I3MAX
      LN=LN+LNLBRA(1,I1)
   81 CONTINUE
   82 CONTINUE
C
C                SET THE GRID PARAMETERS.
```

```
C
                PTOT(1)=T(IT1)
                PTOT(2)=T(IT2)
C
C     ...WRITE OUTPUT FILE IF REQUIRED
C
 100            IF(ISAV.EQ.0) GO TO 101
                WRITE (LSAV)ICVTOT,(PTOT(I1),I1=1,NPTOT),
     .                           (CTOT(I1),I1=1,NCTOT)
C
C     ...WRITE PLOT FILE IF REQUIRED
C
 101            IF(IPLOT.EQ.0) GO TO 103
                NOM1=NOPIES-1
                DO 102 I1=1,NOM1
                I2=ICON(I1)
 102            IF(ABS(T(I2)-TFIXED(I1)).GT.(.01)) GO TO 103
                WRITE(LPA1)ICVTOT,(PTOT(I1),I1=1,NPTOT),
     .                           (CTOT(I1),I1=1,NCTOT)
C
C     CALL PRTCHA TO WRITE OUT TRAJECTORIES
C
 103            IF(ICVTOT.EQ.1) CALL PRICHA(PTOT,CTOT)
C
 170            CONTINUE
                IF(IK.NE.NARIV2) GO TO 172
C
                DO 171 I1=1,NPNT1
 171            BACKSPACE LIA1
                GO TO 180
 172            IF(NRECS(1).EQ.1) GO TO 180
                NBACK=NAA(1)*(NORA(1)-1)
                DO 173 I1=1,NBACK
 173            BACKSPACE LIA1
C
 180            CONTINUE
C
                IF(IJ.NE.NARIV3) GO TO 182
                DO 181 I1=1,NPNT2
 181            BACKSPACE LIA2
                GO TO 190
 182            IF(NRECS(2).EQ.1) GO TO 190
                NBACK=NAA(2)*(NORA(2)-1)
                DO 183 I1=1,NBACK
 183            BACKSPACE LIA2
C
 190            CONTINUE
C
                IF(NRECS(3).EQ.1) GO TO 200
                NBACK=NAA(3)*(NORA(3)-1)
                DO 191 I1=1,NBACK
 191            BACKSPACE LIA3
                IFST3=1
C
 200            CONTINUE
C
                REWIND LOA1
```

```
      REWIND LOA2
      RETURN
C
C     CREATE A NEW PLOT FILE FROM A PREVIOUSLY GENERATED
C     CHARACTERISTIC FILE.
C
300   READ(LIA1) NGRIDS,(NDEP(I),I=1,NGRIDS),
     .           (NARIV(I),I=1,NGRIDS),NRT,NCTOT
      DO 301 I=1,NGRIDS
301   NARIV(NGRIDS+2-I)=NARIV(NGRIDS+1-I)
      NGM1=NGRIDS-1
      NTIMES=2*NGRIDS
      ISOLUT=1
      ISUB1=IT1/2+1
      NPAR1=NARIV(ISUB1)
      ISUB2=IT2/2+1
      NPAR2=NDEP(ISUB2)
      NP=2
      WRITE(LOA1)ISOLUT,NPAR1,NPAR2,NP,NCTOT
C
      DO 303 I=1,NRT
      READ(LIA1)ICVTOT,(T(K),K=1,NTIMES),(CTOT(K),K=1,NCTOT)
      DO 302 J=1,NGM1
      I2=ICHK(J)
302   IF(ABS(T(I2)-TFIXED(J)).GT.(.01)) GO TO 303
      WRITE(LOA1)ICVTOT,T(IT1),T(IT2),(CTOT(K),K=1,NCTOT)
303   CONTINUE
      REWIND LOA1
      RETURN
C
1000  FORMAT(1H ,'INCONSISTENT SPECIFICATION OF MINIMUM PASSAG'
     .'E RADIUS. MAXIMUM HAS BEEN SELECTED.')
C
      END
```

LST   ACCOUNT: AAV   PROJECT: VNDRVEEN

:02.176   IN: 5   OUT: 0   PAGES: 20

TIME: 15:00:27-OCT 30,1969

TIME: 15:00:46-OCT 30,1969

S: 19

119

0.511

```
I.TAP.AUX2
EVEL      3
   C TITLE          ABORT BACK TO EARTH
   C
   C AUTHOR         R.W.SPITZNER
   C
   C SPONSOR        A.A.VANDERVEEN
   C
   C DATE           6-19-69
   C
   C PURPOSE        GENERATE ABORT TRAJECTORIES FROM A GIVEN HELIOCENTRIC
   C                ORBIT.
   C
   C INPUT
   C                TOFUST - TIME TO START ABORTS WRT LAUNCH
   C                TOLAST - TIME TO END ABORTS WRT LAUNCH
   C                DTABRT - INCREMENT IN TIME OF ABORT
   C                DTARIV - INCREMENT IN EARTH ARRIVAL TIMES
   C                NARIV  - NUMBER OF EARTH ARRIVALS CORRESPONDING TO
   C                         EARLIEST ABORT TIME
   C
                    SUBROUTINE AUX2
                    INCLUDE PARSTA,LIST
                    INCLUDE KOMUSE,LIST
                    DIMENSION P(MP),C(MK),CHA(50)
   C
                    NAMELIST/ABORT/TOFUST,TOLAST,DTABRT,DTARIV,NARIV
   C
                    READ(5,ABORT)
                    T1NOM=TPD(2,1)
                    T2NOM=TPD(2,2)
                    TPD(1,1)=20.
                    TPD(1,2)=3.
                    IABORT=0
   C
                    NDEP=(TOLAST+DTABRT/2.-TOFUST)/DTABRT+1
                    DO 5 I=1,NDEP
                    NARV=NARIV+I-1
                    TD=TOFUST+FLOAT(I-1)*DTABRT
                    TDEP=T1NOM+TD
   C
                    DO 5 J=1,NARV
                    TARIV=T1NOM+2.*TD+FLOAT(J-1)*DTARIV
                    P(1)=TDEP
                    P(2)=TARIV
                    IF(IP1TP.EQ.1) GO TO 1
                    P(1)=TARIV
                    P(2)=TDEP
    1               CALL SETUP(P)
                    CALL CONSLG(1,2,0)
                    L9=0
                    DO 4 IJ=1,2
                    IF(IJ.EQ.1) CALL ENCTR(1,20,IJ,CHA)
                    IF(IJ.EQ.2) CALL ENCTR(8, 3,IJ,CHA)
                    L=LENGTH(IJ)
                    DO 3 II=1,L
    3               C(L9+II)=CHA(II)
```

```
      LN=LN+LENGTH(IJ)
4     CONTINUE
      CALL PRTCHA(P,C)
5     CONTINUE
      RETURN
      END
```

```
INTAP.CHARAC
EVEL      3
   C
   C TITLE          CONTROL PROGRAM FOR COMPUTING ENCOUNTERS
   C
   C AUTHOR         R.W.GRUTZNER
   C
   C SPONSOR        A.A.VANDERVEE
   C
   C DATE           5-24-68
   C
   C PURPOSE        MAKE LOGIC DECISIONS CONCERNING SEQUENTIAL CALLS
   C                TO CONIC SEGMENT AND PLANETARY ENCOUNTER ROUTINES.
   C
   C CALL           CALL CHARAC(P,NP,C,NC)
   C
   C INPUT          THROUGH LIST
   C                P                   VECTOR OF PARAMETERS
   C                NP                  NUMBER OF PARAMETERS
   C                NC                  NUMBER OF CHARACTERISTICS TO COMPUTE
   C                THROUGH COMMON
   C                TPA                 TIME-POINT ARRAY
   C                NPONTS              NUMBER OF TIME-POINTS
   C                IENC                SEQUENTIAL ENCOUNTER TYPES
   C                IOPT                OPTIMIZATION FLAG
   C
   C OUTPUT         THROUGH LIST
   C                C                   VECTOR OF CHARACTERISTICS
   C
   C SUBROUTINES    SETUP,CONSEG,ENCTR,OPTFNC
   C
   C      SUBROUTINE STATEMENT
   C
                    SUBROUTINE CHARAC(P,C)
   C
   C      SPECIFICATION STATEMENTS
   C
                    INCLUDE PARSTA,LIST
                    INCLUDE KOMUSE,LIST
                    DIMENSION C(MK),CHA(50),P(MP)
   C
                    CALL SETUP(P)
                    LN=0
                    DO 5 IJ=1,NPONTS
                    N1=TPA(1,IJ)+.01
                    IF(IJ.EQ.NPONTS) GO TO 3
                    N2=TPA(1,IJ+1)
                    IF((N1.EQ.N2).AND.(N1.NE.0)) GO TO 3
   C
   C      ...IF THIS POINT REQUIRES A DEBOOST MANEUVER RESET THE
   C         PLANET AT THE DEPARTURE TIME.
   C
                    IF(IENC(IJ).NE.6) GO TO 2
                    T=TPA(2,IJ)+DELSTP(IJ)
                    TPA(2,IJ)=T
                    CALL FIXPLC(IJ,T)
                    DO 1 I2=1,3
```

```
1       PSTION(I2,IJ)=PLCVEC(I2,IJ)
2       CALL CONSEG(IJ,IJ+1,0)
3       IENTYP=IENC(IJ)
        CALL ENCTR(IENTYP,N1,IJ,CHA)
        L=LENGTH(IJ)
        DO 4 II=1,L
4       C(LN+II)=CHA(II)
        LN=LN+LENGTH(IJ)
5       CONTINUE
        IF(IOPT.EQ.0) RETURN
        CALL OPTENC(C,IOPN,IOTY,VALUE)
        DO 6 II=1,LN
6       C(NCUSR+1-II)=C(NCUSR-II)
        C(1)=VALUE
        RETURN
        END
```

```
INTAP.CONSEG
LEVEL      3
  C
  C TITLE         CONIC SEGMENT
  C
  C AUTHOR        R.W.GRUTZNER
  C
  C DATE          4/17/68
  C
  C PURPOSE       GENERATE SEQUENTIAL TRAJECTORIES CONNECTING TIME-POINTS
  C
  C INPUT         THROUGH LIST
  C               I           DEPARTURE POINT
  C               J           ARRIVAL POINT
  C               IFORCN      =0 FOR SUN CENTER, =PLANET NO. OTHERWISE
  C OUTPUT        THROUGH COMMON
  C               ELPSD       TRAJECTORY DATA
  C
  C      SUBROUTINE STATEMENT
  C
                SUBROUTINE CONSEG(I,J,IFORCN)
  C
                INCLUDE PARSTA,LIST
                INCLUDE KOMUSE,LIST
                INCLUDE KOMPLN,LIST
                COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                COMMON/VHELIO/VHCIN(3,MP),VHCOUT(3,MP)
  C
                DIMENSION R1(3),R2(3),V1(3),V2(3),CROSS(3),VNTRSF(3),
       .        VNODE(3),VNORM(3),TEMP1(3),TEMP2(3),DV(3),VPLNET(3),
       .        ELEM(3,3)
  C
  C      SET UP ARRIVAL AND DEPARTURE VECTORS
  C
                DO 1 IJ=1,3
                R1(IJ)=PSTION(IJ,I)
                R2(IJ)=PSTION(IJ,J)
   1            CONTINUE
                GM=GMSUN
                IF(IFORCN.NE.0) GM=AMU(I)
  C
  C      COMPUTE INCLUDED ANGLE
  C
   2            THETA=VANGLE(R1,R2)
                CROSS(1)=VCROSS(R1,R2)
                IF(CROSS(3).LT.0.) THETA=TWOPI-THETA
                M=3
                IF(THETA.GT.PI) M=1
  C
  C      COMPUTE TIME OF FLIGHT AND CALL LAMBRE TO GENERATE CONIC SEGMENT
  C
                T1=TPA(2,I)
                T=TPA(2,J)-TPA(2,I)
                CALL LAMBRE(GM,M,1,R1,R2,T1,T,THETA,A,E,V1,V2,0,IABORT)
  C
  C      ...STORE INTO COMMON ARRAY ELPSD
  C
```

```
      ELPSD( 1,I)=A
      ELPSD( 2,I)=E
      ELPSD( 3,I)=VMAG(V1)
      ELPSD( 4,I)=VANGLE(R1,V1)*RTD
      COSF1=(A*(1.-E*E)-VMAG(R1))/(E*VMAG(R1))
      IF((COSF1.GT.1.).AND.((COSF1-1.).LT.(10E-3))) COSF1=1.
      IF((COSF1.LT.-1.).AND.((COSF1+1.).GT.(-10E-3))) COSF1=-1.
      ELPSD( 5,I)=ACOS(COSF1)*RTD
      IF(ELPSD(4,I).GT.90.) ELPSD(5,I)=360.-ELPSD(5,I)
      ELPSD( 7,I)=VMAG(V2)
      ELPSD( 8,I)=(PI-VANGLE(R2,V2))*RTD
      ELPSD( 9,I)=ELPSD( 5,I)+THETA*RTD
C
C     COMPUTE INCLINATION OF TRANSFER WITH RESPECT TO ORBIT PLANES OF
C     BOTH ARRIVAL AND DEPARTURE PLANETS. IF EITHER OF THESE IS A
C     NON-PLANETARY TIME POINT THE INCLINATION WITH RESPECT TO THE
C     ECLIPTIC WILL BE COMPUTED.
C
C     ...COMPUTE VECTOR NORMAL TO TRANSFER PLANE
C
      VNTRSF(1)=VCROSS(R1,R2)
      VNTRSF(1)=VUNIT(VNTRSF)
      SGN=SIGN(1.,VNTRSF(3))
      VNTRSF(1)=VSCALR(VNTRSF,SGN)
      DO 4 K=I,J
      N=TPA(1,K)+.01
      IF(N.GT.0) GO TO 3
      IF(K.EQ.I) GO TO 22
      ARVINC=ACOS(VNTRSF(3))
      DO 21 II=1,3
21    VNCONT(II,J)=V2(II)
      GO TO 4
22    DEPINC=ACOS(VNTRSF(3))
      DO 23 II=1,3
23    VNCONT(II,I)=V1(II)
      GO TO 4
3     VNODE(1)=COS(OMEGA(N))
      VNODE(2)=SIN(OMEGA(N))
      VNODE(3)=0.
      IF(K.EQ.I) VNORM(1)=VCROSS(R1,VNODE)
      IF(K.EQ.J) VNORM(1)=VCROSS(R2,VNODE)
      SGN=SIGN(1.,VNORM(3))
      VNORM(1)=VSCALR(VNORM,SGN)
      IF(K.EQ.I) DEPINC=VANGLE(VNORM,VNTRSF)
      IF(K.EQ.J) ARVINC=VANGLE(VNORM,VNTRSF)
4     CONTINUE
      ELPSD( 6,I)=DEPINC*RTD
      ELPSD(10,I)=ARVINC*RTD
      IF(V1(3).LT.VELCTY(3,I)) ELPSD( 6,I)=-DEPINC*RTD
      IF(V2(3).LT.VELCTY(3,J)) ELPSD(10,I)=-ARVINC*RTD
C
C     COMPUTE POLAR COORDINATES OF RELATIVE VELOCITY VECTORS AT
C     DEPARTURE AND ARRIVAL AND STORE IN COMMON ARRAY ENCHA
C
      DO 11 IJ=I,J
      N=TPA(1,IJ)+.01
      IF(N.GT.0) GO TO 5
```

```
      DECLIN=0.
      RASC=0.
      IF(IJ.EQ.I) VRELMG=ELPSD(3,I)
      IF(IJ.EQ.J) VRELMG=ELPSD(7,I)
      GO TO 9
5     DO 6 IK=1,3
6     VPLNET(IK)=VELCTY(IK,IJ)
      IF(IJ.EQ.J) GO TO 7
      DV(1)=VSUB(V1,VPLNET)
      GO TO 8
7     DV(1)=VSUB(V2,VPLNET)
8     VRELMG=VMAG(DV)
      ALPHA=-EQATOR(N)
      BETA=VEFRPH(N)
      FINCL=FINC(N)
      OMEG=OMEGA(N)
      PERM=PERMIN(N)
      CALL MATELM(FINCL,OMEG,PERM,ELEM)
      TEMP1(1)=AMTRTM(ELEM,DV)
      CALL MATELM(ALPHA,BETA,0.,ELEM)
      TEMP2(1)=AMTRTM(ELEM,TEMP1)
      RASC=ATAN2(TEMP2(2),TEMP2(1))
      IF(RASC.LT.0.) RASC=TWOPI+RASC
      DECLIN=ASIN(TEMP2(3)/VRELMG)
9     IF(IJ.EQ.J) GO TO 10
      ENCHA(4,I)=VRELMG
      ENCHA(5,I)=RASC
      ENCHA(6,I)=DECLIN
      GO TO 11
10    ENCHA(1,J)=VRELMG
      ENCHA(2,J)=RASC
      ENCHA(3,J)=DECLIN
11    CONTINUE
      RETURN
      END
```

```
INTAP.CONST
LEVEL      3
      C
      C TITLE         PLANETARY CONSTANTS
      C
      C AUTHOR        P.F. LONG
      C
      C SPONSOR       A.A. VANDERVEEN
      C
      C DATE          11-2-66
      C
      C REVISIONS     7-6-67 R.W.GRUTZNER  SPECIFY SIX STANDARD ORBITAL
      C               ELEMENTS AND REDIMENSION TO ACCOMMODATE ARTIFICIAL PLANET
      C
      C PURPOSE       TO LOAD COMMON AREA PLANET.
      C
      C INPUT         NONE
      C
      C OUTPUT        THROUGH COMMON
      C               AMAJ      SEMI-MAJOR AXIS OF PLANETARY ORBIT
      C               ECC       ECCENTRICITY OF PLANETARY ORBIT
      C               TPHPAS    TIME OF PERIHELION PASSAGE REFERED TO
      C                         JULIAN DATE 2400000
      C               FINC      INCLINATION OF PLANETARY ORBIT TO ECLIPTIC
      C               OMEGA     LONGITUDE OF ASCENDING NODE OF PLANETARY ORBIT
      C               PERMIN    ANGLE FROM ASCENDING NODE TO PERIHELION
      C               VLFRPH    ANGLE BETWEEN ASCENDING NODE AND LOCAL VERNAL
      C                         EQUINOX
      C               EQATOR    INCLINATION OF PLANETARY EQUATOR TO PLANE OF
      C                         MOTION
      C               RHO       PLANETARY RADIAL CONSTANTS (FT)
      C               AMU       GRAVITATIONAL CONSTANTS (FT**3/SEC**2)
      C
      C       SUBROUTINE STATEMENT
      C
              SUBROUTINE CONST(IREAD)
      C
      C       SPECIFICATION STATEMENTS
      C
              INCLUDE KOMPLN,LIST
      C
              NAMELIST/PLANDT/AMU,RHO,SMA,ECC,TPHPAS,FINC,OMEGA,PERMIN,
             . VEFRPH,EQATOR
              COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
      C
      C       SET CONSTANTS
      C
              PI=3.1415921
              TWOPI=2.*PI
              HAFPI=PI/2.
              RTD=57.295780
              DTR=1./RTD
              GMSUN=.017202124**2
      C
      C       ORBITAL ELEMENTS FOR THE PLANETS
      C       ALL TIMES REFERED TO JULIAN DATE 2400000.
      C
```

```
C       ORBITAL PARAMETERS FOR MERCURY
C
                    AMU(1)    = 7.6453542E14
                    RHO(1)    = 1306.5
                    SMA(1)    = 0.387099
                    ECC(1)    = 0.205627
                    TPHPAS(1)= 36899.375
                    FINC(1)   = 7.00399
                    OMEGA(1) = 47.85714
                    PERMIN(1)= 28.97595
                    VEFRPH(1)= 0.0
                    EQATOR(1)= 0.0
C
C       ORBITAL ELEMENTS FOR VENUS
C
                    AMU(2)    = 1.1449900E16
                    RHO(2)    = 3320.
                    SMA(2)    = 0.723332
                    ECC(2)    = .0067916025
                    TPHPAS(2)= 36907.982
                    FINC(2)   = 3.39423
                    OMEGA (2)= 76.31972
                    PERMIN(2)= 54.68859
                    VEFRPH(2)= 0.0
                    EQATOR(2)= 0.0
C
C       ORBITAL ELEMENTS FOR EARTH
C
                    AMU(3)    = 1.4075253E16
                    RHO(3)    = 3438.
                    SMA(3)    = 1.0
                    ECC(3)    = 0.016725495
                    TPHPAS(3)= 36571.869
                    FINC(3)   = 0.0
                    OMEGA(3)  = 0.0
                    PERMIN(3)= 102.25253
                    VEFRPH(3)= -102.25253
                    EQATOR(3)= 23.444356
C
C       ORBITAL ELEMENTS FOR MARS
C
                    AMU(4)    = 1.5152000E15
                    RHO(4)    = 1840.
                    SMA(4)    = 1.523691
                    ECC(4)    = 0.09336903
                    TPHPAS(4)= 36394.110
                    FINC(4)   = 1.84991
                    OMEGA(4) = 49.24903
                    PERMIN(4)= 286.07366
                    VEFRPH(4)= -67.01463
                    EQATOR(4)= 23.98609
C
C       ORBITAL ELEMENTS FOR JUPITER
C
                    AMU(5)    = 4.4672300E18
                    RHO(5)    = 37735.
                    SMA(5)    = 5.202803
```

```
            ECC(5)    = 0.048435
            TPHPAS(5)= 33971.150
            FINC(5)   = 1.30536
            OMEGA(5)  = 100.04444
            PERMIN(5)= 273.62379
            VEFRPH(5)= 0.0
            EQATOR(5)= 0.0
C
C      ORBITAL ELEMENTS FOR SATURN
C
            AMU(6)    = 1.3377600E18
            RHO(6)    = 31075.
            SMA(6)    = 9.538843
            ECC(6)    = 0.055682
            TPHPAS(6)= 31303.352
            FINC(6)   = 2.48991
            OMEGA(6)  = 113.30747
            PERMIN(6)= 338.95700
            VEFRPH(6)= 0.0
            EQATOR(6)= 0.0
C
C      ORBITAL ELEMENTS FOR URANUS
C
            AMU(7)    = 2.0459900E17
            RHO(7)    = 12800.
            SMA(7)    = 19.181951
            ECC(7)    = 0.047209
            TPHPAS(7)= 8696.078
            FINC(7)   = 0.77306
            OMEGA(7)  = 73.79630
            PERMIN(7)= 96.21453
            VEFRPH(7)= 0.0
            EQATOR(7)= 0.0
C
C      ORBITAL ELEMENTS FOR NEPTUNE
C
            AMU(8)    = 2.4225900E17
            RHO(8)    = 11620.
            SMA(8)    = 30.057779
            ECC(8)    = 0.008575
            TPHPAS(8)= 8065.467
            FINC(8)   = 1.77375
            OMEGA(8)  = 131.33980
            PERMIN(8)= 272.93415
            VEFRPH(8)= 0.0
            EQATOR(8)= 0.0
C
C      ORBITAL ELEMENTS FOR PLUTO
C
            AMU(9)    = 1.1697500E16
            RHO(9)    = 3515.
            SMA(9)    = 39.43871
            ECC(9)    = .250236
            TPHPAS(9)= 47618.423
            FINC(9)   = 17.1699
            OMEGA(9)  = 109.88562
            PERMIN(9)= 114.27462
```

```
      VEFRPH(9)= 0.0
      EQATOR(9)= 0.0
C
C     ...IF IREAD .GT. 0 READ /PLANDT/
C
      IF(IREAD.GT.0) READ(5,PLANDT)
C
C     CONVERT TO RADIANS
C
      DO 1 IJ=1,20
      FINC(IJ)=FINC(IJ)*DTR
      OMEGA (IJ)=OMEGA (IJ)*DTR
      PERMIN(IJ)=PERMIN(IJ)*DTR
      EQATOR(IJ)=EQATOR(IJ)*DTR
      VEFRPH(IJ)=VEFRPH(IJ)*DTR
    1 CONTINUE
      RETURN
      END
```

```
C
C TITLE         COMPUTE CONSTRAINTS
C
C AUTHOR        R.W.GRUTZNER
C
C DATE          4/16/69
C
C PURPOSE       COMPUTE THE PRESENT VALUES OF THE DEPENDENT VARIABLES
C               GIVEN A VECTOR OF PARAMETERS
C
C METHOD        SUCCESSIVE CALLS TO CONIC SECTION ROUTINE
C
CINPUTS         THROUGH LIST AND COMMON USER
C               P - VECTOR OF PARAMETERS
C               TPD - TIME POINT DELIMITERS FOR INVARIANT PARAMETERS
C
C OUTPUT        THROUGH LIST AND COMMON USER
C               Y - ERROR VECTOR
C               ENCHA - MATRIX OF ENCOUNTER CHARACTERISTICS
C               ELPSD - MATRIX OF CONIC SECTION CHARACTERISTICS
C
C SUBROUTINES   CONSEG,SETPLT
C
C       SUBROUTINE STATEMENT
C
              SUBROUTINE CONSTR(P,Y)
C
              INCLUDE PARSTA,LIST
              INCLUDE KOMUSE,LIST
              DIMENSION P(MP),Y(MC)
C
              CALL SETUP(P)
C
C       COMPUTE CONIC SEGMENTS AS REQUIRED
C       IF OPTIMIZATION IS TO TAKE PLACE COMPUTE ALL LEGS, SINCE
C       MOST LIKELY WILL HAVE TO ANYWAY AND ALSO THE LOGIC FOR SAID CASE
C       IS NOT DEFINED.
C
              DO 9 I=1,LEGS
              J=1
 5            IF(((ICPREX(J).EQ.0).AND.((ICEN(J).EQ.I).OR.
     .        (ICEN(J).EQ.(I+1)))).OR.((ICPREX(J).EQ.1).AND.
     .        (ICEN(J).EQ.I)).OR.(IOPT.GT.0)) GO TO 6
              J=J+1
              IF(J.LE.NCSUSR) GO TO 5
              GO TO 9
 6            IF(IENC(I).NE.6) GO TO 8
              TPA(2,I)=TPA(2,I)+DELSTP(I)
              T=TPA(2,I)
              CALL FIXPLC(I,T)
              DO 7 II=1,3
 7            PSTION(II,I)=PLCVEC(II,I)
 8            CALL CONSEG(I,I+1,0)
 9            CONTINUE
C
```

```
C       EVALUATE CONSTRAINTS
C
 23             K=0
                IF(IOPT.GT.0) K=1
                DO 22 J=1,NCSUSR
                ICONEN=ICEN(J)
                ICTYPE=ICTY(J)
                IF(ICPREX(J).EQ.0) GO TO 17
C
C       ...EQUALITY CONSTRAINT IS ON LAMBERT SEGMENT
C
                ICTMN4=ICTYPE-4
                GO TO (13,14,15,16), ICTMN4
C
C           SEMI-MAJOR AXIS CONSTRAINT
C
 13             Y(J+K)=ELPSD(1,ICONEN)-TARV(J)
                GO TO 22
C
C           ECCENTRICITY CONSTRAINT
C
 14             Y(J+K)=ELPSD(2,ICONEN)-TARV(J)
                GO TO 22
C
C           DUMMY CONSTRAINT
C
 15             GO TO 22
C
C           DUMMY CONSTRAINT
C
 16             GO TO 22
C
C       ...EQUALITY CONSTRAINT IS AT ENCOUNTER
C
 17             IF(ICTYPE.EQ.1) GO TO 18
                CALL DMCSTR(ICONEN,ICTYPE,0)
                GO TO (19,20,21), ICTYPE
C
C           DELTAV CONSTRAINT
C
 18             Y(J+K)=ENCHA(3,ICONEN)-ENCHA(4,ICONEN)-TARV(J)
                GO TO 22
C
C           DUMMY CONSTRAINT
C
 19             GO TO 22
C
C           DUMMY CONSTRAINT
C
 20             GO TO 22
C
C           DUMMY CONSTRAINT
C
 21             GO TO 22
 22             CONTINUE
                IF(IOPT.EQ.0) RETURN
C
```

```
...EVALUATE PRESENT VALUE OF FUNCTION TO BE OPTIMIZED

      GO TO (24,25,26,27), IOTY

...THE FOLLOWING ARE ALL DUMMIES SINCE OPT. FNCTS. ARENT DEFINED.
      Y(1)=0.
      RETURN
      Y(1)=1.
      RETURN
      Y(1)=2.
      RETURN
      Y(1)=3.
      RETURN
      END
```

```
  FAP.ECANOM
  L       3
C
C TITLE           ECCENTRIC ANOMALY
C
C AUTHOR          M.MINKOFF
C
C SPONSOR         A.A.VANDERVEEN
C
C DATE            4-30-67
C
C REVISION        R.W.GRUTZNER 4-30-67 LIFTED EQUATIONS FROM MAIN1 AND
C                 PUT INTO FORM OF SUBROUTINE
C
C PURPOSE         COMPUTE ECCENTRIC ANOMALY GIVEN MEAN ANOMALY
C                 AND ECCENTRICITY
C
C METHOD          ITERATIVE SOLUTION OF KEPLERS EQUATION
C
C INPUT           THROUGH LIST
C                       E              ECCENTRICITY
C                       M              MEAN ANOMALY
C
C OUTPUT          THROUGH LIST
C                       ECAN           ECCENTRIC ANOMALY
C
      SUBROUTINE ECANOM(E,M,ECAN)
C
                  REAL M
C
                  ECAN=M
                  I=1
 10               GUESS=ECAN-E*SIN(ECAN)
                  IF((ABS(M-GUESS)).LT.(.000001)) GO TO 20
                  ECAN=ECAN+(M-GUESS)/(1.0-E*COS(ECAN))
                  I=I+1
                  IF(I.LE.100) GO TO 10
                  WRITE(6,100)
 100              FORMAT(43H0CONVERGENCE NOT ACHIEVED IN 100 ITERATIONS)
 20               RETURN
                  END
```

```
 TAP.ENCTR
 VEL       3
    C
    C TITLE          ENCOUNTER CHARACTERISTICS
    C
    C AUTHOR         R.W.GRUTZNER
    C
    C SPONSOR        A.A.VANDERVEEN
    C
    C DATE           5-24-68
    C
    C PURPOSE        CALL PROPER SEQUENCE OF ENCOUNTER SUBROUTINES
    C
    C CALL           CALL ENCTR(IENTYP,LAMBR,N1,IJ,CHA,LENGTH)
    C
    C INPUT          THROUGH LIST
    C                IENTYP              ENCOUNTER TYPE
    C                LAMBR               CONIC SEGMENT FLAG
    C                N1                  DEPART PLANET NUMBER
    C                IJ                  TIME-POINT NUMBER OF DEPARTURE
    C                THROUGH COMMON
    C                ELPSD               ELLIPSE CHARACTERISTICS FOR NEXT SEG.
    C
    C OUTPUT         THROUGH LIST
    C                LENGTH              NUMBER OF CHARACTERISTICS COMPUTED
    C                CHA                 CHARACTERISTIC VECTOR
    C
    C SUBROUTINES    LODCHA WITH ENTRIES LOAD,LOAD1,LOAD2,LOAD3,LOAD6,LOAD7
    C
    C       SUBROUTINE STATEMENT
    C
                     SUBROUTINE ENCTR(IENTYP,N1,IJ,CHA)
    C
    C       SPECIFICATION STATEMENTS
    C
                     INCLUDE PARSTA,LIST
                     INCLUDE KOMUSE,LIST
                     DIMENSION CHA(50)
    C
                     IF(IENTYP.EQ.0) GO TO 80
                     CALL ENCTR1(IENTYP,N1,IJ,$1)
     1               GO TO (10,20,30,40,50,60,70,10), IENTYP
    C
    C       ...COMPUTE ARRIVAL OR DEPARTURE CHARACTERISTICS
    C
     10              CALL ENCTR2(IENTYP,$11)
     11              CALL ENCTR3($12)
     12              CALL ENCTR5($13)
     13              CALL LOAD (CHA,$14)
     14              CALL LOAD1(CHA,$80)
    C
    C       ...COMPUTE BALLISTIC FLYBY CHARACTERISTICS
    C
     20              CALL ENCTR2(IENTYP,$21)
     21              CALL LOAD(CHA,$22)
     22              CALL LOAD2(CHA,$80)
    C
```

```
C       ...COMPUTE SINGLE IMPULSE FLYBY
C
 30           CALL [illegible](IFNTYP,$31)
 31           CALL [illegible](I[illegible])
 32           CALL [illegible]($33)
 33           CALL [illegible](CHA,$34)
 34           CALL [illegible](CHA,$59)
C
C       ...COMPUTE TWO IMPULSE FLYBY
C
 40           RETURN
C
C       ...COMPUTE DOUBLE TO AND FROM ELLIPSE(A.L.S. ROUTINE)
C
 50           RETURN
C
C       COMPUTE DEBOOST TO CIRCULAR AND ELLIPTIC
C
 60           CALL [illegible]CTR2(IFNTYP,$61)
 61           CALL [illegible]CTR3($62)
 62           CALL [illegible]($63)
 63           CALL LOAD (CHA,$64)
 64           CALL [illegible]1(CHA,$80)
C
C       ...COMPUTE A PLANE CHANGE MANEUVER
C
 70           CALL ENCTR6(IJ,$71)
 71           CALL LOAD (CHA,$72)
 72           CALL LOAD7(CHA,$80)
C
C       STORE ELLIPSE DATA AFTER ENCOUNTER DATA
C
 80           IF(L[illegible](IJ).EQ.0) RETURN
              L=LNENC[illegible](IJ)
              DO 90 IK=1,10
 90           CHA(L+IK)=ELPS[illegible](IK,IJ)
              RETURN
              END
```

```
INTAP.ENCONT
_VLL      3
      C
      C TITLE         COMPUTE ENCOUNTER CHARACTERISTICS
      C
      C AUTHOR        R.W.GRUTZNER
      C
      C SPONSOR       A.A.VANDERVEEN
      C
      C DATE          5-24-63
      C
      C PURPOSE       COMPUTE ENCOUNTER CHARACTEPISTICS AS DICTATED BY
      C               FLAG IENTYP.
      C
      C METHOD        PROGRAM IS ENTERED THROUGH ANY OF SEVERAL ENTRY POINTS
      C               AND COMPUTES CHARACTERISTICS PECULIAR TO THAT SECTION.
      C               AN ENCOUNTER MAY CONSIST OF SEVERAL SETS OF COMPUTATIONS
      C
      C INPUT         THROUGH LIST
      C               IENTYP                 ENCOUNTER TYPE
      C               N1                     PLANET NUMBER
      C               IJ                     TIME-POINT NUMBER
      C               $X                     X=RETURN STATEMENT IN CALLING PROGRAM
      C               THROUGH COMMON
      C               ELEMENTS OF COMMON BLOCK /PENCTR/ DEPENDING UPON
      C               WHICH ENTRY POINT IS USED. SEE DOCUMENTATION OF
      C               /PENCTR/ FOR DEFINITIONS.
      C
      C OUTPUT        THROUGH COMMON
      C               ELEMENTS OF COMMON BLOCK /PENCTR/ DEPENDING UPON
      C               WHICH ENTRY POINT IS USED. SEE DOCUMENTATION OF
      C               /PENCTR/ FOR DEFINITIONS.
      C
      C SUBROUTINES   EQRTR,OPTRAN,PERXFR
      C
      C       SUBROUTINE STATEMENT
      C
                      SUBROUTINE ENCONT
      C
      C       SPECIFICATION STATEMENTS
      C
                      INCLUDE PARSTA,LIST
                      INCLUDE KOMUSE,LIST
                      INCLUDE KOMPEN,LIST
                      INCLUDE KOMPLN,LIST
                      COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                      COMMON/VHELIO/VHCIN(3,MP),VHCOUT(3,MP)
      C
                      DIMENSION V(2),VHIN(3),VHOUT(3),VLOCAL(3),VNMIN(3),
           .          VNMOUT(3),VNODE(3),VNORM(3),VPER(3),VPERP(3),
           .          CROS(3),POSIT(3),RTRAN(3),REQATR(3),A2(2),E(2),
           .          DV(3),V1(3),V2(3),ASMTOT(2),ELEM(3,3),UNITN(3)
      C
      C       ***************************************************************
      C       ***************************************************************
      C
                      ENTRY ENCTR1(IENTYP,N1,IJ,$)
```

```
C
C       THIS ENTRY COMPUTES CHARACTERISTICS COMMON TO ALL TYPES OF
C       ENCOUNTERS
C
C       ...DETERMINE WHETHER A PASSAGE RADIUS OR ALTITUDE IS SPECIFIED
C          FOR THIS TYPE OF ENCOUNTER AND SET INPUTS
C
              NPLNO=N1
              TIMOUT=TPA(2,IJ)
              TIMEIN=TIMOUT-DELSTP(IJ)
              GO TO (1,3,1,1,1,1,3,1),IENTYP
 1            ARATIO=ARATS(IJ)
              DECLP=PERDEC(IJ)
              IF((PSGCON(IJ).GT.(.01)).AND.(PSGCON(IJ).LT.(10.))) GOTO2
              PRAD=(RHO(NPLNO)+PSGCON(IJ))*6080.
              PALT=PSGCON(IJ)
              GO TO 3
 2            PRAD=PSGCON(IJ)*RHO(NPLNO)*6080.
              PALT=(PRAD-1.)*RHO(NPLNO)
 3            VIN   =ENCHA(1,IJ)
              RASIN =ENCHA(2,IJ)
              DECLIN=ENCHA(3,IJ)
              VOUT  =ENCHA(4,IJ)
              RASOUT=ENCHA(5,IJ)
              DECOUT=ENCHA(6,IJ)
              IF(IENTYP.EQ.1) CALL VASTOT(1)
              IF(IENTYP.EQ.8) CALL VASTOT(8)
C
C       ...CONVERT ASYMPTOTIC VELOCITIES TO FEET/SEC. AND ALSO STORE
C          INTO V1 AND V2 VECTORS FOR SUBSEQUENT USE
C
              V(1)= VIN*97702.1
              V(2)=VOUT*97702.1
              V1(1)=-COS(DECLIN)*COS(RASIN)
              V1(2)=-COS(DECLIN)*SIN(RASIN)
              V1(3)=-SIN(DECLIN)
              V2(1)=COS(DECOUT)*COS(RASOUT)
              V2(2)=COS(DECOUT)*SIN(RASOUT)
              V2(3)=SIN(DECOUT)
              BA=PI-VANGLE(V1,V2)
              IF(NPLNO.EQ.0) GO TO 7
              CALL SUBSOL
              RETURN 4
 7            RASCSP=0.
              DECLSP=0.
              RETURN 4
C
C                   ***********************************
C                   ***********************************
C
              ENTRY ENCTR2(IENTYP,$)
C
C       THIS ENTRY COMPUTES SPATIAL PARAMETERS FOR THE PLANETARY ORBITS
C
              BEND=BA+PI
C
C       ...DETERMINE ASCENDING NODE VECTOR
```

```
C
               VNORM(1)=VCROSS(V1,V2)
               UNITN(1)=0.
               UNITN(2)=0.
               UNITN(3)=1.
               VNODE(1)=VCROSS(VNORM,UNITN)
C
C      ...DETERMINE THE NUMBER OF NODES. IF THE V-INFINITY COMPONENTS
C         NORMAL TO THE EQUATOR ARE OF DIFFERENT SIGNS THERE IS ONLY
C         ONE REAL NODE. FIRST SET UP SOME TEST ANGLES
C
               ANGLE1=VANGLE(V1,VNODE)
               ANGLE2=VANGLE(V2,VNODE)
               ANG=ANGLE1+ANGLE2
C
C      CHECK FOR NUMBER OF REAL NODES
C
               IF(V1(3)*V2(3).GT.(0.)) GO TO 11
C
C      THIS CASE ONE REAL NODE. DETERMINE WHETHER ASCENDING NODE LIES
C      BETWEEN OR WITHOUT ASYMPTOTES
C
               IF(V1(3).GT.(0.)) GO TO 9
C
C      ...ASCENDING NODE IS REAL
C
               IF(ABS(ANG-BEND).LT.(.0001)) GO TO 8
               VNODE(1)=VSCALR(VNODE,-1.)
 8             CROS(1)=VCROSS(V1,VNODE)
               CROS(1)=VUNIT(CROS)
               PLINC=ACOS(CROS(3))
C
C      .....COMPUTE OMEGA1 AND OMEGA2
C
               OMEGA1=PI-VANGLE(V1,VNODE)
               OMEGA2=VANGLE(V2,VNODE)
               GO TO 15
C
C      ...ASCENDING NODE IS VIRTUAL
C
 9             IF(ABS(ANG-BEND).GT.(.0001)) GO TO 10
               VNODE(1)=VSCALR(VNODE,-1.)
 10            CROS(1)=VCROSS(VNODE,V1)
               CROS(1)=VUNIT(CROS)
               PLINC=ACOS(CROS(3))
C
C      .....COMPUTE OMEGA1 AND OMEGA2
C
               OMEGA1=VANGLE(V1,VNODE)+PI
               OMEGA2=TWOPI-VANGLE(V2,VNODE)
               GO TO 15
C
C      THIS CASE FOR TWO REAL NODES. FIND NODE CLOSEST TO THE
C      ASYMPTOTE WHICH IS ASCENDING
C
 11            IF(V1(3).LT.(0.)) GO TO 13
C
```

```
C       ...FIND THE NODE CLOSEST TO THE OUTGOING ASYMPTOTE SINCE
C          OUTBOUND VELOCITY VECTOR IS ASCENDING
C
              IF(ANGLE1.GT.ANGLE2) GO TO 12
              VNODE(1)=VSCALR(VNODE,-1.)
 12           CROS(1)=VCROSS(VNODE,V2)
              CROS(1)=VUNIT(CROS)
              PLINC=ACOS(CROS(3))
C
C       .....COMPUTE OMEGA1 AND OMEGA2
C
              OMEGA1=VANGLE(V1,VNODE)+PI
              OMEGA2=VANGLE(V2,VNODE)
              GO TO 15
C
C       ...FIND THE NODE CLOSEST TO THE INCOMING ASYMPTOTE SINCE
C          INBOUND VELOCITY VECTOR IS ASCENDING
C
 13           IF(ANGLE1.LT.ANGLE2) GO TO 14
              VNODE(1)=VSCALR(VNODE,-1.)
 14           CROS(1)=VCROSS(V1,VNODE)
              CROS(1)=VUNIT(CROS)
              PLINC=ACOS(CROS(3))
C
C       .....COMPUTE OMEGA1 AND OMEGA2
C
              OMEGA1=PI-VANGLE(V1,VNODE)
              OMEGA2=FACPI-VANGLE(V2,VNODE)
C
C       COMPUTE LONGITUDE OF THE ASCENDING NODE
C
 15           VNODE(1)=VUNIT(VNODE)
              POMEGA=ATAN2(VNODE(2),VNODE(1))
C
C       IF NEITHER ARRIVE,DEPART NOR BALLISTIC FLYBY, RETURN. IF
C       BALLISTIC FLYBY COMPUTE PERIAPSE ALTITUDE, VELOCITY,RADIUS
C       DECLINATION, RIGHT ASCENSION AND OMEGAP, ANGLE BETWEEN ASCENDING
C       NODE AND PERIAPSE. IF ARRIVE OR DEPART COMPUTE ONLY OMEGAP.
C
              GO TO (17,16,18,18,18,18,18,17), IENTYP
 16           E(1)=-1./COS(BEND/2.)
              A2(1)=AMU(NPLNO)/V(1)**2
              PERRAD=A2(1)*(E(1)-1.)
              VELPER=SQRT(2.*AMU(NPLNO)/PERRAD+V(1)**2)
              PRAD=PERRAD/6080./RHO(NPLNO)
              PALT=(PRAD-1.)*RHO(NPLNO)
C
C       ...COMPUTE UNIT PERIAPSE VECTOR AS NORMALIZED SUM OF
C                         VIN-VOUT
C
              DELVEL=VIN-VOUT
              VPER(1)=VADD(V1,V2)
              VPER(1)=VSCALR(VPER,-1.)
              VPER(1)=VUNIT(VPER)
              ALPHP=ATAN2(VPER(2),VPER(1))
              DECLP=ASIN(VPER(3))
 17           OMEGAP=OMEGA1+BEND/2.+TWOPI
```

```
          CLEGAP=AMOD(CLGAP,TWOPI)
 18       RETURN 2
C
C                   ********************************
C                   ********************************
C
          ENTRY FNCTN3(*)
C
C     THIS ENTRY COMPUTES DEBOOST AND BOOST VELOCITIES TO AND FROM
C     CIRCULAR AND ELLIPTIC ORBITS FOR ARRIVAL, DEPARTURE AND STOPOVER
C     CASES
C
C     ...COMPUTE CIRCULAR VELOCITY AT PERIAPSE
C
          VELC=SQRT(AMU(NPLNO)/PRAD)
C
C     ...COMPUTE ELLIPTIC PERIAPSE VELOCITY
C
          A3=PRAD*ARATIO
          AENM=A3/6080.
          VELPE=SQRT(AMU(NPLNO)*((2./PRAD)-(1./A3)))
C
C     ...COMPUTE ELLIPTIC APOAPSE VELOCITY
C
          VAPO=VELPE*PRAD/(2.*A3-PRAD)
C
C     ...COMPUTE PERIAPSE VELOCITY ON INBOUND AND OUTBOUND LEGS
C
          VELPH1=SQRT(2.*(VELC**2)+V(1)**2)
          VELPH2=SQRT(2.*(VELC**2)+V(2)**2)
C
C     ...COMPUTE DEBOOST AND BOOST TO AND FROM CIRCULAR
C
          DELVC1=VELPH1-VELC
          DELVC2=VELPH2-VELC
C
C     ...COMPUTE DEBOOST AND BOOST TO AND FROM ELLIPTIC
C
          DELVE1=VELPH1-VELPE
          DELVE2=VELPH2-VELPE
C
C     ...SUM BOOST AND DEBOOST
C
          SUMVC=DELVC1+DELVC2
          SUMVE=DELVE1+DELVE2
C
          RETURN 1
C
C                   ********************************
C                   ********************************
C
          ENTRY FNCTN4(*)
C
C     THIS ENTRY COMPUTES SINGLE IMPULSE OPTIMUM HYPERBOLIC TRANSFERS.
C
C     DETERMINE WHICH OPTION SHALL BE USED TO COMPUTE MINIMUM IMPULSE.
C     IF A PERIAPSE TRANSFER AT RMIN YIELDS AN EXCESS IN THE BEND ANGLE
```

```
C        USE MARC[illegible]S WHICH PROVIDE A KICK NEAR THE NATURAL
C        PERIAPSIS [illegible]. IF THE BEND IS INSUFFICIENT USE A SEARCH
C        METHOD [illegible] A MINIMUM SUBJECT TO THE CONSTRAINT THAT THE
C        TRANSFER [illegible] NO CLOSER THAN RMIN TO THE PLANET.
C
               [illegible]
               [illegible]=([illegible]LT[illegible](NPLNO))*6080.
               DO 19 IK=1,2
               A2(IK)=AMU(NPLNO)/V(IK)**2
               E(IK)=1.+[illegible]IN/A2(IK)
 19            ASMTOT(IK)=ACOS(-1./E(IK))
C
C        ...TEST FOR TYPE OF TRANSFER
C
               IF(([illegible]T(1)+ASMTOT(2)).LT.BEND) GO TO 26
C
C        ...USE [illegible] FORMULATION FOR TOO MUCH BEND
C
               S[illegible]
               VE[illegible]T=(V(2)-V(1))/(V(2)+V(1))
               D[illegible](VE[illegible]ACT*TAN(S))
               G[illegible]
               DEL[illegible](V(1)+V(2))*[illegible]S(SIN(D))
               VE[illegible](1)*COS(D)*SQRT(2.*COS(S)**2-COS(D)**2)/SIN(G)
               R[illegible](NPLNO)/VESCAP**2
C
C        ...IF MARC[illegible] SOLUTION RESULTS IN A TRANSFER ALTITUDE LESS THAN
C           ALLOWABLE, COMPUTE A PERIAPSE TRANSFER, WHICH WILL BE VERY
C           CLOSE TO [illegible]MAL.
C
               IPE[illegible]
               IF(R[illegible]) GO TO 20
               IPE[illegible]1
               CALL [illegible](A[illegible],E,R)
 20            VTINT1=SQRT(2.*AMU(NPLNO)/R+V(1)**2)
               VTOUT1=SQRT(2.*AMU(NPLNO)/R+V(2)**2)
               RAL[illegible]1=R/(6080.*RHO(NPLNO))
               ALT[illegible]1=(RA[illegible]TR[illegible]-1.)*RHO(NPLNO)
C
C        ...SET VAL[illegible] PERIAPSE TRANSFER. IF THIS IS NOT THE CASE
C           THE NEXT [illegible] WILL OVERWRITE THE VALUES.
C
               GA[illegible]
               TH1T1=0.
               TH2T1=0.
               IF(I[illegible].1) GO TO 27
C
C        ...HAVING FOUND THE RADIUS OF INTERSECTION FOR THE OPTIMUM
C           IMPULSE TRANSFER USING MARCHAL'S EQUATIONS, IT IS REQUIRED
C           TO ITERATE IN ORDER TO COMPUTE THE ECCENTRICITIES OF THE
C           TWO HYPERBOLII AND THE TRUE ANOMALIES OF THE INTERSECTION
C
               [illegible]=([illegible])**2-VTINT1**2-VTOUT1**2)/(2.*VTINT1
     .                                                  *VTOUT1**2)
               GA[illegible]ACOS(COSD[illegible])
               [illegible]
               [illegible]
```

```
      DELE=0.
      EMAX=1.+R/A2(1)
      EIT=EMAX-.001
      DO 23 IK=1,20
      EIT=EIT+DELE
      SEA1SQ=(A2(1)*(EIT**2-1.)/R)/(2.+R/A2(1))
      A1=ASIN(SQRT(SEA1SQ))
      ANGLE1=A1-A1
      ANTEST=ANGLE1-GAMMAV
      SNTEST=SIN(ANTEST)
      RONA1=R/A2(1)
      RONA2=R/A2(2)
      E22=SQRT((2.*RONA2+RONA2**2)*SNTEST+1.)
      F1=ACOS(((EIT**2-1.)/RONA1-1.)/EIT)
      F2=ACOS(((E22**2-1.)/RONA2-1.)/E22)
      ASMTOT(1)=ACOS(-1./EIT)
      ASMTOT(2)=ACOS(-1./E22)
      ERROR=BEND-(ASMTOT(1)-F1)-(ASMTOT(2)-F2)
      IF(ABS(ERROR).LT.(.0001)) GO TO 24
      IF(IFIRST.EQ.0) GO TO 21
      IFIRST=0
      ERBACK=ERROR
      DELE=-.01
      GO TO 23
21    SLOPE=(ERROR-ERBACK)/DELE
      DELE=-ERROR/SLOPE
      ERBACK=ERROR
      IF(((EIT+DELE).LT.EMAX).AND.((EIT+DELE).GT.(1.)))GO TO 23
      IF((EIT+DELE).LT.EMAX) GO TO 22
      DELE=EMAX-EIT
      GO TO 23
22    DELE=1.01-EIT
23    CONTINUE
24    THE1T1=F1
      THE2T1=-F2
      E(1)=EIT
      E(2)=E22
      ALF1=ASIN(SQRT((A2(2)*(E22**2-1.)/R)/(2.+R/A2(2))))
      COSAL2=(VTINT1**2+DELVT1**2-VTOUT1**2)/(2.*VTINT1*DELVT1)
      ALF2=ARCOS(COSAL2)
      IF(VTOUT1.GT.VTIN(1)) GO TO 25
      GAMT1=GAMMAV+ALF1+ALF2+PI
      GO TO 27
25    GAMT1=ALF2-(PI-GAMMAV-ALF1)
      GO TO 27
C
C     THERE IS INSUFFICIENT NATURAL BEND. COMPUTE OPTIMUM IMPULSE BY
C     PASSING PERIAPSE ON THE SLOW HYPERBOLA AND TRANSFERRING
C     THEREAFTER TO THE OUTBOUND (INBOUND) LEG. AS AN INITIAL POINT
C     COMPUTE THOSE HYPERBOLII WHICH HAVE EQUAL PERIAPSE RADII.
C
26    CALL EQRTR(E,A2,V,BEND,NPLNO,VEQUAL,THZERO,GAM)
      CALL OPTRAN(VEQUAL,THZERO,GAM)
C
C     COMPUTE OMEGT1,ALPH1,DECL1, THE ANGLE BETWEEN THE ASCENDING
C     NODE, THE RIGHT ASCENSION AND THE DECLINATION RESPECTIVELY OF
C     THE TRANSFER POINT.
```

```
C
 27           OMEGT1=OMEGA1-THETT1+ACOS(-1./E(1))+PI
              IF(OMEGT1.GT.TWOPI) OMEGT1=AMOD(OMEGT1,TWOPI)
              IF(OMEGT1.LT.0.) OMEGT1=AMOD(OMEGT1,TWOPI)+TWOPI
C
C       ...SET UP MATRIX ELEMENTS TO TRANSFORM TRANSFER POINT TO
C          EQUATORIAL SYSTEM IN ORDER TO COMPUTE RIGHT ASCENSION AND
C          DECLINATION
C
              CALL MATELM(PLINC,POMEGA,OMEGT1,ELEM)
              RTRAN(1)=1.
              RTRAN(2)=0.
              RTRAN(3)=0.
              REGATR(1)=AMTV(ELEM,RTRAN)
C
C       ...COMPUTE RIGHT ASCENSION AND DECLINATION
C
              DECL1=ASIN(REGATR(3))
              ALPH1=ATAN2(REGATR(2),REGATR(1))
              RETURN 1
C
C                  ************************************
C                  ************************************
C
              ENTRY ENCTR5($)
C
C       THIS ENTRY COMPUTES BEND ANGLES
C
              PSI1=ASIN(1./(1.+PRAD*V(1)**2/AMU(NPLNO)))
              PSI2=ASIN(1./(1.+PRAD*V(2)**2/AMU(NPLNO)))
              DELPSI=BA-(PSI1+PSI2)
              RETURN 1
C
C                  ************************************
C                  ************************************
C
              ENTRY ENCTR6(IJ,$)
C
C       THIS ENTRY COMPUTES CHARACTERISTICS UNIQUE TO A PLANE CHANGE
C
C       ...SET THE INBOUND AND OUTBOUND HELIOCENTRIC VECTORS.
C
              DO 28 I=1,3
              VHIN(I)=VHCIN(I,IJ)
              VHOUT(I)=VHCOUT(I,IJ)
 28           POSIT(I)=PSTION(I,IJ)
              BA=VANGLE(VHIN,VHOUT)
C
C       ...COMPUTE VECTORS NORMAL TO INBOUND AND OUTBOUND PLANES AND THEN
C          VELOCITY VECTOR DIFFERENCE AT INTERSECTION.
C
              VNMIN(1)=VCROSS(POSIT,VHIN)
              VNMOUT(1)=VCROSS(POSIT,VHOUT)
              DV(1)=VSUB(VHOUT,VHIN)
C
C       ...COMPUTE INCLINATION BETWEEN INBOUND AND OUTBOUND PLANES.
C
```

```
      DINC=VANGLE(VNMIN,VNMOUT)
      IF(DV(3).LT.0.) DINC=-DINC
C
C     ...SET UP A VECTOR NORMAL TO THE ECLIPTIC.
C
      VPERP(1)=0.
      VPERP(2)=0.
      VPERP(3)=1.
C
C     ...THE ASCENDING NODE VECTOR IS (VPERP X (R X V)).
C
      VNODE(1)=VCROSS(VPERP,VNMIN)
      VNODE(1)=VUNIT(VNODE)
C
C     ...COMPUTE EULER ANGLES FOR COORDINATE TRANSFORMATION.
C
      OMEG=ATAN2(VNODE(2),VNODE(1))
      PERM=VANGLE(POSIT,VNODE)
      IF(POSIT(3).LT.0.) PERM=TWOPI-PERM
      CALL MATELM(PLINC,OMEG,PERM,ELEM)
      VLOCAL(1)=AMTRTM(ELEM,DV)
C
C     ...COMPUTE DV MAGNITUDE AND ANGLES.
C
      DELVT1=VMAG(DV)
      ALPH1=ATAN2(VLOCAL(2),VLOCAL(1))
      DECL1=ASIN(VLOCAL(3)/DELVT1)
      RETURN 2
      END
```

```
INIAP.EQRTR
EVEL      5
   C
   C TITLE          MINIMUM PASSAGE TRANSFER
   C
   C AUTHOR         R.W.
   C
   C SPONSOR        A.A.VAN DER VEEN
   C
   C DATE           1-25-68
   C
   C PURPOSE        FIND THE ELEMENTS OF HYPERBOLA THAT HAVE A MINIMUM
   C                PERIAPSE DISTANCE, TO USE AS A STATING GUESS FOR OPTRAN
   C
   C METHOD         STANDARD CONIC EQUATIONS
   C
   C INPUT          THROUGH LIST
   C                      BEND              BEND ANGLE PLUS PI
   C                      A2(2)             SEMI-TRANSVERSE AXES
   C                      E(2)              ECCENTRICITIES OF HYPERBOLII
   C
   C OUTPUT         THROUGH COMMON
   C                      DELVEL            VELOCITY REQUIRED FOR TRANSFER
   C                      DELTH1            TRUE ANOMALY ON 'SLOWER' HYPERBOLA
   C                                        OF TRANSFER POINT
   C
                    SUBROUTINE EQRTR(E,A2,VINF,BEND,J,VEQUAL,THZERO,GAM)
   C
                    INCLUDE KOMPLN,LIST
                    COMMON/CONFAC/PI,DUM1(5)
                    DIMENSION E(2),A2(2),V(2),PARAM(2),SINGAM(2),GAMMA(2),
         .          VINF(2)
   C
                    DELTHT=BEND-ARCOS(-1./E(1))-ARCOS(-1./E(2))
                    A=E(1)*A2(2)*(E(2)**2-1.)-A2(1)*E(2)*COS(DELTHT)*
         .                     (E(1)**2-1.)
                    B=A2(1)*E(2)*(E(1)**2-1.)*SIN(DELTHT)
                    C=A2(1)*(E(1)**2-1.)-A2(2)*(E(2)**2-1.)
                    DISCR=SQRT(A**2+B**2-C**2)
                    ROOT1=(A*C+B*DISCR)/(A**2+B**2)
                    ROOT2=(A*C-B*DISCR)/(A**2+B**2)
                    DELTH1=ARCOS(ROOT1)
                    I1=1
    1               IF((((C-A*COS(DELTH1))/B)-SIN(DELTH1)).LT.(.0001))
         .                   GO TO 2
                    I1=I1+1
                    IF(I1.EQ.3) GO TO 4
                    DELTH1=ARCOS(ROOT2)
                    GO TO 1
    2               R=A2(1)*(E(1)**2-1.)/(1.+E(1)*COS(DELTH1))
                    DO 3 IJ=1,2
                    V(IJ)=SQRT(2.*AMU(J)/R+VINF(IJ)**2)
                    PARAM(IJ)=A2(IJ)*(E(IJ)**2-1.)
                    SINGAM(IJ)=SQRT((PARAM(IJ)/R)/(2.-(R/A2(IJ))))
    3               GAMMA(IJ)=ARSIN(SINGAM)
                    GAMMA(2)=PI-GAMMA(2)
   C
```

```
C         COMPUTE VELOCITY DIFFERENCE AT INTERSECTION
C
                 VEQUAL=SQRT(V(1)**2+V(2)**2-2.*V(1)*V(2)*COS(GAMMA(1)
                                   -GAMMA(2)))
                 COSPHI=(V(2)**2+VEQUAL**2-V(1)**2)/(2.*V(2)*VEQUAL)
                 PHI=ACOS(COSPHI)
                 GAM=GAMMA(2)+PHI
                 GO TO 5
  4              WRITE(6,111)
  111            FORMAT(1H0,20X,'**********ERROR IN FORTR***********')
                 CALL EXIT
  5              THZERO=DELTH1
                 IF(VINF(1).GT.VINF(2)) THZERO=DELTHT-THZERO
                 RETURN
                 END
```

```
INTAP.FIXPLC
LEVEL     3
   C TITLE          COORDINATES OF PLANET'S CENTER IN ECLIPTIC
   C
   C AUTHOR         R.W.GROTZNER
   C
   C DATE           4/16/68
   C
   C PURPOSE        CONVERT THE TIME POINT ARRAY TO A VECTOR IN THE
   C                ECLIPTIC COORDINATE SYSTEM
   C
   C METHOD         KEPLER'S EQUATION AND MATRIX MULTIPLICATION
   C
   C INPUT          THROUGH LIST
   C                IJ         TIME POINT SEQUENCE NUMBER
   C                T          TIME (JD-2400000.)
   C
   C OUTPUT         THROUGH COMMON
   C                PLCVEC     VECTOR COMPONENTS
   C
   C SUBROUTINES    MANOM,ECANOM,TRUANM,RADMAG,MATELM,MXM,CONST
   C
   C      SUBROUTINE STATEMENT
   C
                    SUBROUTINE FIXPLC(IJ,T)
   C
                    INCLUDE PARSTA,LIST
                    INCLUDE KOMUSE,LIST
                    INCLUDE KOMPLN,LIST
                    DIMENSION RCOMP(3),ELEM(3,3),RECLIP(3),VCOMP(3),VECLIP(3)
   C
                    N=TPD(1,IJ)+.01
                    IF(N.GT.0) GO TO 2
                    DO 1 IK=1,3
                    VELCTY(IK,IJ)=0.
    1               PLCVEC(IK,IJ)=0.
                    GO TO 4
   C
   C      SET PLANETARY CONSTANTS IF N.GT.0
   C
    2               E=ECC(N)
                    A=SMA(N)
                    TP=TPHPAS(N)
                    FINCL=FINC(N)
                    OMEG=OMEGA(N)
                    PERM=PERMIN(N)
   C
   C      ...COMPUTE MEAN ANOMALY
   C
                    CALL MANOM(A,T,TP,ANOMM)
   C
   C      ...COMPUTE ECCENTRIC ANOMALY
   C
                    CALL ECANOM(E,ANOMM,ECAN)
   C
   C      ...COMPUTE TRUE ANOMALY
   C
```

```
      CALL TRUANM(ECAN,E,TRNM)
C
C     ...COMPUTE RADIUS AND VELOCITY COMPONENTS IN ORBIT PLANE
C
      CALL RADMAG(A,E,ECAN,TRNM,RAD,RCOMP)
      CALL SLOPE(A,E,RAD,TRNM,GAMMA)
      CALL VELMAG(A,RAD,TRNM,GAMMA,VCOMP)
C
C     ...COMPUTE ROTATION MATRIX ELEMENTS
C
      CALL MATELM(FINCL,OMEG,PERM,ELEM)
C
C     ...CONVERT TO ECLIPTIC SYSTEM
C
      RECLIP(1)=AMTV(ELEM,RCOMP)
      VECLIP(1)=AMTV(ELEM,VCOMP)
C
C     ...STORE VECTOR
C
      DO 3 IK=1,3
      VELCTY(IK,IJ)=VECLIP(IK)
    3 PLCVEC(IK,IJ)=RECLIP(IK)
    4 RETURN
      END
```

```
INTAP.FIXREL
:VEL    3
  C TITLE           COMPUTE RELATIVE VECTOR
  C
  C AUTHOR          R.W.GRUIZNER
  C
  C DATE            4/16/68
  C
  C PURPOSE         IF A POINT OTHER THAN THE PLANET CENTER IS SPECIFIED
  C                 AS A TARGET THIS ROUTINE CONVERTS THE PLANET CENTERED
  C                 SPECIFICATION TO THE ECLIPTIC SO THAT THE VECTOR MAY BE
  C                 ADDED TO THAT REPRESENTING THE PLANET CENTER. ALSO SETS
  C                 UP NON-PLANETARY TIME POINT COORDINATES IN THE ECLIPTIC
  C
  C METHOD          MATRIX MULTIPLICATION
  C
  C INPUT           THROUGH LIST
  C                 IJ          TIME POINT INDICATOR
  C
  C OUTPUT          THROUGH COMMON
  C                 RELVEC      TRANSFORMED VECTOR
  C
  C       SUBROUTINE STATEMENT
  C
                    SUBROUTINE FIXREL(IJ)
  C
                    INCLUDE PARSTA,LIST
                    INCLUDE KOMUSE,LIST
                    INCLUDE KOMPLN,LIST
                    DIMENSION R(3),ELEM(3,3),RPRIME(3),RVEC(3)
  C
                    N=TPD(1,IJ)+.01
                    RELVEC(1,IJ)=TPA(3,IJ)*COS(TPA(5,IJ))*COS(TPA(4,IJ))
                    RELVEC(2,IJ)=TPA(3,IJ)*COS(TPA(5,IJ))*SIN(TPA(4,IJ))
                    RELVEC(3,IJ)=TPA(3,IJ)*SIN(TPA(5,IJ))
  C
  C      IF N=0 TIME POINT IS NOT A PLANET. HENCE COORDINATES ARE GIVEN
  C      HELIOCENTRICALLY. IF N.GT.0 POIN IS A PLANET AND COORDINATES ARE
  C      GIVEN IN A PLANET CENTERED SYSTEM. MUST CONVERT TO ECLIPTIC
  C
                    IF(N.EQ.0) RETURN
                    DO 1 I=1,3
   1                R(I)=RELVEC(I,IJ)
                    ALPHA=-EQATOR(N)
                    BETA=VEFRPH(N)
                    FINCL=FINC(N)
                    OMEG=OMEGA(N)
                    PERM=PERMIN(N)
                    CALL MATELM(ALPHA,BETA,0.,ELEM)
                    RPRIME(1)=AMTV(ELEM,R)
                    CALL MATELM(FINCL,OMEG,PERM,ELEM)
                    RVEC(1)=AMTV(ELEM,RPRIME)
                    DO 2 I=1,3
   2                RELVEC(I,IJ)=RVEC(I)
                    RETURN
                    END
```

```
INTAP.LAMBRE
LEVEL      3
      C
      C TITLE          SOLVE LAMBERT'S PROBLEM
      C
      C AUTHOR         P.A.WHITLOCK (MODIFIED BY R.W.GRUTZNER)
      C
      C DATE           4/17/68
      C
      C PURPOSE        GIVEN TWO POSITION VECTORS AND THE FLIGHT TIME, FIND THE
      C                CONIC SECTION ELEMENTS
      C
      C METHOD         THIS SUBROUTINE EMPLOYS A NEW SOLUTION TO LAMBERT'S
      C                PROBLEM BY W. LIM DESCRIBED IN JAN, 2967 MIT REVIEW
      C
      C INPUT          THROUGH LIST
      C                GM         CENTRAL FORCE GRAVITATIONAL CONSTANT
      C                M          =3 IF THETA.LT.PI, =1 IF THETA.GT.PI
      C                N          1+NUMBER OF COMPLETE REVOLUTIONS, =1 HYPERBOLA
      C                R1         INITIAL POSITION VECTOR
      C                R2         FINAL POSITION VECTOR
      C                THETA      INCLUDED ANGLE
      C                T          REQUIRED FLIGHT TIME
      C
      C OUTPUT         THROUGH LIST
      C                A          SEMI-TRANSVERSE AXIS OF SOLUTION CONIC
      C                E          ECCENTRICITY OF SOLUTION CONIC
      C                V1         DEPARTURE VELOCITY VECTOR
      C                V2         ARRIVAL VELOCITY VECTOR
      C
      C     SUBROUTINE STATEMENT
      C
                SUBROUTINE LAMBRE(GM,M,N,R1,R2,T1,T,THETA,A,E,V1,V2,
           .                      IDBUG,IABORT)
      C
                COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
      C
                DIMENSION R1(3),R2(3),V1(3),TV1(3),TV2(3),V2(3),ANGMOM(3)
                NAMELIST/ERROR/R1,R2,T1,T,THETA,A,E,C,S,X,Y,W,BEM,TM,
           .    SGNT,COEFT,TP,QM,Q,SQ,CQ,SGNO,FACT,GRP,XQ
      C
                MINA=0
                R1M=VMAG(R1)
                R2M=VMAG(R2)
                SGNM=M-2
                Q=0.
      C
      C     DEFINE C AS CHORD FROM R1 TO R2 AND S AS HALF THE PERIMETER
      C
       5        C=SQRT(R1M**2+R2M**2-2.*R1M*R2M*COS(THETA))
                S=(R1M+R2M+C)/2.
                X=1.+R2M/R1M
                Y=C/R1M
                W=SGNM*SQRT(1.-(C/(R1M+R2M))**2)
      C
      C     DEFINE TIME OF FLIGHT ON MINIMUM ENERGY ELLIPSE
      C
```

```
      BEM=ACOS((2.*C-S)/S)
      TM=SQRT((S**3)/(8.*GM))*(PI-SGNM*(BEM-SIN(BEM)))
      SGNT=SIGN(1.,TM-T)
      IF(ABS(T-TM).GT..005) GO TO 2
      A=.25*(R1M+R2M+C)
      MINA=1
      GO TO 12
    2 COEFT=SQRT(.5*(R1M+R2M)**3/GM)
C
C     COMPUTE TIME OF FLIGHT ON PARABOLIC ORBIT
C
      TP=COEFT*SQRT(1.-W)*(2.+W)/3.
      IONCE=1
      TTOL=.001
      IF(ABS(T-TP).LT.(.1)) TTOL=.1
C
C     COMPUTE MAXIMUM Q FOR HYPERBOLIC ORBIT
C
      QM=ALOG((1./ABS(W))+SQRT((1./W**2)-1.))
      FN=N-1
C
C     TEST IF FIRST TIME THROUGH
C
      IF(ABS(Q).GT.(.001)) GO TO 6
C
C     INITIALIZE Q
C
      Q=.5
      IF(N.LE.1)GO TO 6
      Q=.75+FN*1.5
C
C     BEGIN ITERATION ON Q
C
    6 DO 10 J=1,50
      IF(T-TP) 8,7,7
C
C     ELLIPTICAL ORBIT
C
    7 IF(Q.LT.PI)GO TO 71
      Q=(PI+QOLD)/2.
   71 SQ=SIN(Q)
      CQ=COS(Q)
      SGNO=1.
      GO TO 9
C
C     HYPERBOLIC ORBIT
    8 IF(Q.LT.QM)GO TO 81
      Q=QM/2.
C
C     ...COMPUTE SINH AND COSH
C
   81 EQ=EXP(Q)
      EMQ=1./EQ
      SQ=(EQ-EMQ)/2.
      CQ=(EQ+EMQ)/2.
      SGNO=-1.
    9 FACT=1.-W*CQ
```

```
      IF(FACT.LT.0.) WRITE(6,ERROR)
      FROOT=SQRT(FACT)
C
C     COMPUTE SEMI-MAJOR AXIS
C
      A=SGNO*.5*X*R1M*FACT/SQ**2
 91   GRP=FACT*(FM*PI+Q)-SQ*(CQ-W)
      XQ=SGNO*COEFT*FROOT*GRP/SQ**3
      PXQ=XQ*(-3.*CQ/SQ+SGNO*.5*W*SQ/FACT+SGNO*SQ/GRP*
     1(2.*SQ+ *Q))
C
C     CHECK ON CONVERGENCE
C
 92   DX=XQ-T
      IF(ABS(DX).LT.TTOL) GO TO 12
C
C     COMPUTE NEW Q
C
      QOLD=Q
      Q=Q-DX/PXQ
      IF(Q.GT.0.)GO TO 93
      Q=QOLD/2.
 93   IF(IDBUG.EQ.0)GO TO 10
      WRITE(6,94)QOLD,PXQ,CQ,DX,XQ
 10   CONTINUE
      IF(IONCE.GT.1) GO TO 11
      IONCE=2
      TTOL=10.*TTOL
      GO TO 6
 11   WRITE(6,112) XQ,T,DX
      CALL EXIT
C
C     COMPUTE DEPARTURE VELOCITY VECTOR
C
 12   D=.25*R1M/A
      U=SQRT(1./(X-Y)-D)
      IF(THETA.GT.PI) U=-U
      IF(MINA.NE.1) GO TO 13
      V=0.
      GO TO 14
 13   V=SGNT*SQRT(1./(X+Y)-D)
 14   SCALR=SQRT(GM/R1M)
      COEF=SCALR*(U+V)/C
      SCALR=SCALR*(U-V)/R1M
      V1(1)=VSUB(R2,R1)
      V1(1)=VSCALR(V1,COEF)
      TV1(1)=VSCALR(R1,SCALR)
      V1(1)=VADD(V1,TV1)
      IF(IABORT.EQ.1) CALL ORBIT(R1,V1,T1)
C
C     COMPUTE ANGULAR MOMENTUM VECTOR AND THEN MAGNITUDE
C
      ANGMOM(1)=VCROSS(R1,V1)
      AMOMSC=VMAG(ANGMOM)
C
C     COMPUTE SEMI-LATUS RECTUM AND ECCENTRICITY
C
```

```
               P=AMOMSC**2/GM
               E=SQRT(1.-P/A)
               COEF1=VDOT(R1,V1)*(1.-COS(THETA))/(P*R1M)-
     .                  SQRT(GM/P)*SIN(THETA)/R1M
               COEF2=1.-R1M*(1.-COS(THETA))/P
C
C     COMPUTE ARRIVAL VELOCITY VECTOR
C
               V2(1)=VSCALR(R1,COEF1)
               TV2(1)=VSCALR(V1,COEF2)
               V2(1)=VADD(V2,TV2)
C
C     ...CONVERT VELOCITIES TO EMOS
C
               EMO=365.25/TWOPI
               V1(1)=VSCALR(V1,EMO)
               V2(1)=VSCALR(V2,EMO)
C
C     FORMAT STATEMENTS
C
 94            FORMAT(5H QOLD1PE14.7,4H PXQE14.7,3H CQE14.7,3H DXE14.7,
     .         3H XQE14.7)
 112           FORMAT(1H ,2X,'*****NO LAMBERT CONVERGENCE - COMPUTED TI'
     .         'ME='F10.6,5X,'REQUIRED TIME='F10.6,5X,'DELTA T='E15.8)
C
               RETURN
               END
```

```
INTAP.LODCHA
EVEL      3
   C
   C TITLE          LOAD CHARACTERISTIC VECTOR
   C
   C AUTHOR         R.W.SPRITZER
   C
   C SPONSOR        A.A.VANDERVEEN
   C
   C DATE           5-24-6
   C
   C PURPOSE        LOAD CHARACTERISTIC VECTOR (CHA) WHICH IS OF VARIAPLE
   C                LENGTH DEPENDING ON TYPE OF ENCOUNTER.
   C
   C METHOD         USE OF SEVERAL ENTRY POINTS INTO SUBROUTINE.
   C
   C INPUT          THROUGH LIST
   C                $X, WHERE X IS RETURN STATEMENT IN CALLING PROGRAM
   C                THROUGH COMMON
   C                /PENCTR/ - SEE LIST FOR DEFINITIONS
   C
   C OUTPUT         THROUGH LIST
   C                CHA - VECTOR OF CHARACTERISTICS
   C
   C ENTRY POINTS LOAD,LOAD1,LOAD2,LOAD3,LOAD6,LOAD7
   C
   C       SUBROUTINE STATEMENT
   C
                  SUBROUTINE LODCHA
   C
   C       SPECIFICATION STATEMENTS
                  INCLUDE KOMPEN,LIST
                  INCLUDE KOMPLN,LIST
                  COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                  DIMENSION CHA(50)
   C
   C       ******************************************************************
   C
                  ENTRY LOAD (CHA,$)
   C
   C       THIS ENTRY LOADS THE ENCOUNTER VECTORS CHA. THE CHARACTERISTICS
   C       LOADED HERE ARE COMMON TO ALL ENCOUNTER TYPES.
   C
                  CHA( 1)=NPLNO
                  CHA( 2)=TIMEIN
                  CHA( 3)=VIN
                  CHA( 4)=RASIN*RTD
                  CHA( 5)=DECLIN*RTD
                  CHA( 6)=TIMOUT
                  CHA( 7)=VOUT
                  CHA( 8)=RASOUT*RTD
                  CHA( 9)=DECOUT*RTD
                  CHA(10)=BA*RTD
                  CHA(11)=RASCSP*RTD
                  CHA(12)=DECLSP*RTD
                  RETURN 2
   C
```

```
C                 ************************************
C
               ENTRY LOAD1(CHA,$)
C
C      THIS ENTRY LOADS CHARACTERISTICS UNIQUE TO ARRIVAL, DEPARTURE
C      OR DEBOOST AND BOOST TO AND FROM ELLIPTIC OR CIRCULAR ORBITS.
C
               CHA(13)=DELPSI*RTD
               CHA(14)=PSI1*RTD
               CHA(15)=PSI2*RTD
               CHA(16)=PLINC*RTD
               CHA(17)=POMEGA*RTD
               CHA(18)=OMEGA1*RTD
               CHA(19)=OMEGA2*RTD
               CHA(20)=OMEGAP*RTD
               CHA(21)=ALPHP*RTD
               CHA(22)=DECLP*RTD
               CHA(23)=PALT
               CHA(24)=PRAD/6080./RHO(NPLNO)
               CHA(25)=ARATIO
               CHA(26)=AENM
               CHA(27)=VAPO
               CHA(28)=VELC
               CHA(29)=VELPE
               CHA(30)=VELPH1
               CHA(31)=VELPH2
               CHA(32)=DELVC1
               CHA(33)=DELVC2
               CHA(34)=SUMVC
               CHA(35)=DELVE1
               CHA(36)=DELVE2
               CHA(37)=SUMVE
               RETURN 2
C
C                 ************************************
C
               ENTRY LOAD2(CHA,$)
C
C      THIS ENTRY LOADS CHARACTERISTICS UNIQUE TO BALLISTIC FLYBYS.
C
               CHA(13)=POMEGA*RTD
               CHA(14)=PLINC*RTD
               CHA(15)=OMEGA1*RTD
               CHA(16)=OMEGA2*RTD
               CHA(17)=OMEGAP*RTD
               CHA(18)=ALPHP*RTD
               CHA(19)=DECLP*RTD
               CHA(20)=PRAD
               CHA(21)=PALT
               CHA(22)=VELPER
               CHA(23)=DELVBL
               RETURN 2
C
C                 ************************************
C
               ENTRY LOAD3(CHA,$)
C
```

```
C       THIS ENTRY LOADS CHARACTERISTICS UNIQUE TO IMPULSIVE FLYBYS.
C
                CHA(13)=DELPSI*RTD
                CHA(14)=PSI1*RTD
                CHA(15)=PSI2*RTD
                CHA(16)=ALTTR1
                CHA(17)=RADTR1
                CHA(18)=DELVT1
                CHA(19)=GAMT1*RTD
                CHA(20)=THE1T1*RTD
                CHA(21)=THE2T1*RTD
                CHA(22)=VTINT1
                CHA(23)=VTOUT1
                CHA(24)=PLINC*RTD
                CHA(25)=POMEGA*RTD
                CHA(26)=OMEGA1*RTD
                CHA(27)=OMEGA2*RTD
                CHA(28)=OMEGT1*RTD
                CHA(29)=ALPH1*RTD
                CHA(30)=DECL1*RTD
                RETURN 2
C
C                   ************************************
C
                ENTRY LOAD6(CHA,$)
                RETURN 2
C
C                   ************************************
C
                ENTRY LOAD7(CHA,$)
C
C       THIS ENTRY LOADS CHARACTERISTICS UNIQUE TO A PLANE CHANGE.
C
                CHA(13)=ALPH1*RTD
                CHA(14)=DECL1*RTD
                CHA(15)=DINC*RTD
                CHA(16)=DELVT1
                RETURN 2
C
                END
```

```
INTAP.MANOM
LEVEL      3
      C
      C TITLE        MEAN ANOMALY
      C
      C AUTHOR       R.A. SWITZER
      C
      C SPONSOR      A.A.VAN DORVEEN
      C
      C DATE         4-20-67
      C
      C PURPOSE      COMPUTE MEAN ANOMALY BETWEEN ZERO AND 2*PI GIVEN TIME
      C              AND PERIHELION PASSAGE TIME
      C
      C METHOD       SOLUTION OF M=2*PI(T-TP)/PERIOD
      C
      C INPUT        THROUGH COMMON
      C                    PI
      C                    TWOPI
      C                    K                 EARTH MEAN MOTION
      C              THROUGH LIST
      C                    A                 SEMI MAJOR AXIS
      C                    T                 TIME AT WHICH MEAN ANOMALY IS DESIRED
      C                    TP                TIME OF PERIHELION PASSAGE
      C
      C OUTPUT       THROUGH LIST
      C                    ANOMM             MEAN ANOMALY AT TIME T
      C
            SUBROUTINE MANOM(A,T,TP,ANOMM)
      C
                     COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
      C
       30            ANOMM=(T-TP)*SQRT(GMSUN/A**3)
                     IF((ANOMM.GT.0.).AND.(ANOMM.LT.TWOPI)) GO TO 10
                     IF(ANOMM.LT.0.0) GO TO 20
                     ANOMM=AMOD(ANOMM,TWOPI)
                     GO TO 10
       20            TAU=TWOPI*SQRT(A**3/GMSUN)
                     TP=TP-TAU
                     GO TO 30
       10            RETURN
                     END
```

```
INTAP.MATELM
EVEL      3
  C
  C TITLE         ORTHOGONAL MATRIX ELEMENTS
  C
  C AUTHOR        R.W.GRUTZNER
  C
  C SPONSOR       A.A.VANDERVEEN
  C
  C DATE          4-30-67
  C
  C PURPOSE       COMPUTE ELEMENTS FOR ROTATION MATRIX WHICH TRANSFORMS
  C               FROM PLANETARY COORDINATES TO ECLIPTIC
  C
  C INPUT         THROUGH LIST
  C                    INCL            INCLINATION OF PLANETS ORBIT
  C                                    TO ECLIPTIC
  C                    OMEG            LONGITUDE OF ASCENDING NODE RELATIVE
  C                                    TO ARIES
  C                    PERM            ANGLE BETWEEN ASCENDING NODE AND
  C                                    PLANETARY PERIHELION MEASURED IN
  C                                    PLANETARY ORBIT
  C
  C OUTPUT        THROUGH LIST
  C                    ELEM(3,3)       MATRIX ELEMENTS
  C
        SUBROUTINE MATELM(INCL,OMEG,PERM,ELEM)
  C
              DIMENSION ELEM(3,3)
              REAL INCL
  C
              ELEM(1,1)=COS(PERM)*COS(OMEG)
     .                       -SIN(PERM)*COS(INCL)*SIN(OMEG)
              ELEM(1,2)=-SIN(PERM)*COS(OMEG)
     .                       -COS(PERM)*COS(INCL)*SIN(OMEG)
              ELEM(1,3)=SIN(INCL)*SIN(OMEG)
              ELEM(2,1)=COS(PERM)*SIN(OMEG)
     .                       +SIN(PERM)*COS(INCL)*COS(OMEG)
              ELEM(2,2)=-SIN(PERM)*SIN(OMEG)
     .                       +COS(PERM)*COS(INCL)*COS(OMEG)
              ELEM(2,3)=-SIN(INCL)*COS(OMEG)
              ELEM(3,1)=SIN(PERM)*SIN(INCL)
              ELEM(3,2)=COS(PERM)*SIN(INCL)
              ELEM(3,3)=COS(INCL)
              RETURN
              END
```

```
INTAP.OPTRAN
EVEL      3
  C
  C TITLE          OPTIMUM HYPERBOLIC TRANSFER
  C
  C AUTHOR         R.W.CL[illegible]IZ[illegible]R
  C
  C SPONSOR        A.A.VANDERVEEN
  C
  C DATE           6-10-67
  C
  C PURPOSE        TO FIND THE MINIMUM VELOCITY REQUIREMENT TO TRANSFER
  C                FROM AN INCOMING TO AN OUTGOING HYPERBOLA SUCH THAT A
  C                SPECIFIED MINIMUM PASSAGE ALTITUDE IS ACHIEVED
  C
  C METHOD         FIX ONE OF THE HYPERBOLII BY SPECIFYING A MINIMUM
  C                RADIUS AND THEN SOLVE FOR THE INTERSECTION OF THAT
  C                HYPERBOLA AND A SECOND HYPERBOLA SUCH THAT THE ASYMPTOTIC
  C                CONSTRAINT IS MET AND COMPUTE DELTA VELOCITY.
  C                THIS PROCEEDURE IS CARRIED OUT OVER THE POSSIBLE RANGE
  C                OF TRANSFER POINTS AND THE MINIMUM IS SELECTED
  C
  C CALL           CALL OPTRAN(VEQUAL,THZERO)
  C
  C INPUT          THROUGH LIST
  C                     J                 PLANET NUMBER
  C                THROUGH COMMON
  C                     VELAR             MAGNITUDE OF ARRIVAL VELOCITY VECTOR
  C                     VELDP             MAGNITUDE OF DEPARTURE VELOCITY VECTOR
  C                     BEND              BEND ANGLE BETWEEN VELOCITY VECTORS
  C                     ALTT              ALTITUDE OF PASSAGE
  C
  C                     RTOPT             RADIUS CORRESPONDING TO OPTIMUM
  C                                       TRANSFER POINT
  C                     ALTOPT            ALTITUDE OF OPTIMUM POINT
  C OUTPUT         THROUGH COMMON
  C                     VOPT              MINIMUM TRANSFER VELOCIY
  C                     RTOPT             RADIUS CORRESPONDING TO OPTIMUN
  C
  C
                 SUBROUTINE OPTRAN(VEQUAL,THZERO,GAM)
  C
  C      SPECIFICATION STATEMENTS
  C
                 INCLUDE KOMPEN,LIST
                 INCLUDE KOMPLN,LIST
                 COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
  C
                 DIMENSION A2(2),ECC2(2),V(2)
  C
  C      INITIALIZE
  C
                 BEND=BA+PI
                 V(1)=VIN*97702.1
                 V(2)=VOUT*97702.1
                 IF(V(1).LT.V(2)) GO TO 1
                 VSAVE=V(1)
```

```
      V(1)=V(2)
      V(2)=VSAVE
    1 A2(1)=AMU(NPLNO)/V(1)**2
      A2(2)=AMU(NPLNO)/V(2)**2
      RP=PRAD
      E1=1.+RP/A2(1)
      P1=A2(1)*(E1**2-1.)
C
C     INITIALIZE
C
      AST1=ARCOS(-1./E1)
      ALPHA=BEND-AST1
      DELTH=.1*DTR
      THETA=THZERO
      DTEST=.001*DTR
      IFIRST=1
C
C     COMPUTE VELOCITY DIFFERENCE AT INFINITY
C
      DEL=2.*AST1-BEND
      DELINF=SQRT(V(1)**2+V(2)**2-2.*V(1)*V(2)*COS(DEL))
C
      DO 5 IK=1,999
C
C     ...THETA IS TRIAL POINT AT WHICH DELTAV IS TO BE COMPUTED. RTRAN IS
C        CORRESPONDING RADIUS
C
      THETA=THETA+DELTH
      IF(THETA.GT.(AST1-DTEST)) GO TO 7
      RTRAN=A2(1)*(E1**2-1.)/(1.+E1*COS(THETA))
C
C     ...COMPUTE BETA, ANGLE BETWEEN TRIAL POINT AND OUTBOUND ASYMPTOTE
C
      BETA=ALPHA-THETA
C
C     ...SET UP COEFFICIENTS TO DETERMINE ECCENTRICITY OF OUTBOUND LEG
C
      COEF1=A2(2)/(RTRAN*SIN(BETA))
      COEF2=(RTRAN*(COS(BETA)-1.)-A2(2))/(RTRAN*SIN(BETA))
      C1=(2.*COEF1*COEF2-1.)/COEF1**2
      C2=(COEF2**2+1.)/COEF1**2
      ASIGN=1.
      DISCSQ=C1**2-4.*C2
      IF(DISCSQ.LT.(0.)) GO TO 5
      DO 3 IJ=1,2
      ECC2(IJ)=(-C1+ASIGN*SQRT(DISCSQ))/2.
      IF(ECC2(IJ).LT.(1.)) GO TO 2
      ECTEST=SQRT(ECC2(IJ))
      ANTEST=THETA-(ALPHA-ARCOS(-1./ECTEST))
      RTEST=A2(2)*(ECTEST**2-1.)/(1.+ECTEST*COS(ANTEST))
      IF(ABS(RTRAN-RTEST).GT.(5000.)) GO TO 2
      ECCSQ=ECC2(IJ)
      E=SQRT(ECCSQ)
    2 ASIGN=-1.
    3 CONTINUE
C
C     ...COMPUTE FLIGHT PATH ANGLE AND VELOCITIES ON BOTH HYPERBOLII
```

```
C          AT INTERSECTION AND THEN VECTOR DIFFERENCE
C
                VELIN=SQRT(2.*AMU(NPLNO)/RTRAN+V(1)**2)
                VELOUT=SQRT(2.*AMU(NPLNO)/RTRAN+V(2)**2)
                SNGIN =SQRT((P1/RTRAN)/(2.-RTRAN/A2(1)))
                P2=A2(2)*(ECCSQ-1.)
                SNGOUT=SQRT((P2/RTRAN)/(2.-RTRAN/A2(2)))
                GAMIN=ARSIN(SNGIN)
                GAMOUT=ARSIN(SNGOUT)
                IF(ACOS(-1./E).LT.BETA) GAMOUT=PI-GAMOUT
                DELTAV=SQRT(VELIN**2+VELOUT**2-2.*VELIN*VELOUT*
     .                  COS(GAMIN-GAMOUT))
                COSALF=(VELOUT**2+DELTAV**2-VELIN**2)/(2.*VELOUT*DELTAV)
C
C       ...INHIBIT OVERFLOW ON ACOS
C
                IF((COSALF.GT.(0.)).AND.(ABS(COSALF-1.).LT.(.0001)))
     .               COSALF=1.
                IF((COSALF.LT.(0.)).AND.(ABS(COSALF+1.).LT.(.0001)))
     .               COSALF=-1.
                ALF=ARCOS(COSALF)
                GAMT1=ALF+GAMOUT
C
C       TEST FOR FIRST TIME THROUGH LOOP
C
                IF(IFIRST.EQ.(0)) GO TO 10
                SLOPE=(DELTAV-VEQUAL)/DELTH
                IF(SLOPE.LT.(0.)) GO TO 9
                IF(DELINF.GT.VEQUAL) GO TO 8
C
C       TRANSFER AT INFINITY
C
 7              DELVT1=DELINF
                RADTR1=99.9999
                ALTTR1=9999.99
                COSALF=(V(2)**2+DELINF**2-V(1)**2)/(2.*V(2)*DELINF)
                ALF=ARCOS(COSALF)
                GAMT1=DEL+ALF
                IF(VIN.GT.VOUT) GO TO 13
                THE1T1=-AST1
                THE2T1=-HAFPI
                VTINT1=V(1)
                VTOUT1=V(2)
                RETURN
 13             THE2T1=AST1
                THE1T1=HAFPI
                VTINT1=V(2)
                VTOUT1=V(1)
                RETURN
C
C       TRANSFER AT 'EQUAL PERIAPSE'
C
 8              DELVT1=VEQUAL
                RADTR1=RTRAN/6080./RHO(NPLNO)
                ALTTR1=RTRAN/6080.-RHO(NPLNO)
                GAMT1=GAM
                IF(VIN.GT.VOUT) GO TO 14
```

```
      VTINT1=VELIN
      VTOUT1=VELOUT
      THE1T1=-THETA
      THE2T1=BEND-AST1+THE1T1-ACOS(-1./E)
      RETURN
14    VTINT1=VELOUT
      VTOUT1=VELIN
      THE2T1=THETA
      THE1T1=AST1+THE2T1+ACOS(-1./E)-BEND
      RETURN
9     IFIRST=0
      GO TO 4
C
C  ...IF THIS VALUE OF DELTAV IS LESS THAN PREVIOUS VALUE CONTINUE
C     OTHERWISE REDUCE STEP SIZE AND POLARITY AND CONTINUE
C
10    IF(DELTAV.LT.DVM1) GO TO 4
      DELTH=-.1*DELTH
      IF(ABS(DELTH).GT.DTEST) GO TO 4
      DELVT1=DVM1
      RADTR1=RM1/6080./RHO(NPLNO)
      ALTTR1=(RADTR1-1.)/RHO(NPLNO)
      VTINT1=V(1)
      VTOUT1=V(2)
      TH2=ACOS(-1./E)+THETA-BEND+AST1
      IF(VIN.LT.VOUT) GO TO 11
      VTINT1=VELOUT
      VTOUT1=VELIN
      THE1T1=TH2
      THE2T1=THETA
      GO TO 12
11    THE1T1=-THETA
      THE2T1=-TH2
12    RETURN
4     DVM1=DELTAV
      RM1=RTRAN
5     CONTINUE
      RETURN
      END
```

```
INTAP.ORBIT
EVEL     3
   C TITLE           ORBITAL PARAMETERS
   C
   C AUTHOR          R.W.GRUTZNER
   C
   C SPONSOR         A.A.VANDERVEEN
   C
   C DATE            6-19-69
   C
   C PURPOSE         COMPUTE THE ORBITAL PARAMETERS OF A SPACECRAFT GIVEN ITS
   C                 POSITION AND VELOCITY VECTORS AND JULIAN DATE.
   C
   C RESTRICTION     PROGRADE ORBITS ONLY I.E. Z COMPONENT OF R X V GT 0.
   C
                     SUBROUTINE ORBIT(R,V,T)
   C
                     DIMENSION R(3),V(3),ANODE(3),UNITN(3),H(3),TEST(3),
        •            UNITH(3)
                     COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                     INCLUDE KOMPLN,LIST
                     DATA/UNITN/0.,0.,1./
   C
                     H(1)=VCROSS(R,V)
                     P=(VMAG(H))**2/GMSUN
                     UNITH(1)=VUNIT(H)
                     FINC(20)=ACOS(UNITH(3))
                     ANODE(1)=VCROSS(UNITN,H)
                     OMEGA(20)=ATAN2(ANODE(2),ANODE(1))
                     ENGY=VDOT(V,V)/2.-GMSUN/VMAG(R)
                     SMA(20)=-GMSUN/(2.*ENGY)
                     ECC(20)=SQRT(1.-P/SMA(20))
                     COSF0=(P/VMAG(R)-1.)/ECC(20)
                     FZERO=ACOS(COSF0)
                     TEST=VDOT(R,V)
                     IF(TEST.LT.0.) FZERO=TWOPI-FZERO
                     SMALOM=VANGLE(ANODE,R)
                     TEST(1)=VCROSS(ANODE,R)
                     IF(TEST(3).LT.0.) SMALOM=TWOPI-SMALOM
                     TANODE=FZERO-SMALOM
                     PERMIN(20)=TWOPI-TANODE
                     TANE2=SQRT((1.-ECC(20))/(1.+ECC(20)))*TAN(FZERO/2.)
                     EANOM=2.*ATAN(TANE2)
                     IF(FZERO.GT.PI) EANOM=TWOPI+EANOM
                     TPHPAS(20)=T-(EANOM-ECC(20)*SIN(EANOM))*
        •                                    SQRT(SMA(20)**3/GMSUN)
   C
   C        SET UP PROBE VALUES SAME AS EARTH FOR FOLLOWING:
   C
                     AMU(20)=AMU(3)
                     RHO(20)=RHO(3)
                     VEFRPH(20)=VEFRPH(3)
                     EQATOR(20)=EQATOR(3)
                     RETURN
                     END
```

```
INTAP.PAP
EVEL      3
  C TITLE          PAP - PARAMETRIC ANALYSIS PROGRAM
  C
  C AUTHOR         P.F. LONG
  C
  C DATE           4-3-68
  C
  C PURPOSE        CONSIDER A SYSTEM WHOSE CHARACTERISTICS ARE CONTINUOUS
  C                FUNCTIONS OF THE PARAMETERS P(1),P(2),....,P(NP). PAP
  C                PERFORMS PARAMETRIC ANALYSIS WITH RESPECT TO THESE
  C                PARAMETERS AS FOLLOWS:
  C
  C                   1. THE PARAMETER SPACE FORMED BY HOLDING P(1) AND P(2)
  C                   FIXED IS SEARCHED FOR POINTS SATISFYING SPECIFIED
  C                   CONSTRAINTS.
  C
  C                   2. AN ATTEMPT IS MADE TO MAP THE SOLUTIONS FOUND TO
  C                   ALL POINTS OF A GRID FORMED FROM THE DOMAINS OF P(1)
  C                   AND P(2).
  C
  C                   3. FOR THE SOLUTIONS FOUND, THE SYSTEM CHARACTERISTICS
  C                   ARE COMPUTED AND STORED.
  C
  C                   4. AN INTERFACE IS PROVIDED FOR PROGRAMS DOING
  C                   ADDITIONAL COMPUTATION BASED ON THE GRIDS DEVELOPED.
  C
  C                   5. CONTOUR PLOTS OF THE CHARACTERISTICS AND THE
  C                   NON-GRID PARAMETERS ARE PLOTTED AS SPECIFIED.
  C
  C METHOD          THE PARAMETER SPACE IS SEARCHED BY STARTING AN ITERATIVE
  C                 PROCEDURE AT EACH POINT OF A GRID FORMED FROM THE DOMAINS
  C                 OF P(3),P(4),...,P(NP). TO AVOID REPEATED CONVERGENCE
  C                 TO THE SAME SOLUTION OR NEEDLESS WANDERING, THE ITERATIVE
  C                 PROCEDURE IS STOPPED IF IT STRAYS OUTSIDE A BOUNDARY
  C                 CENTERED ON THE INITIAL POINT. PROVISIONS ARE MADE FOR
  C                 OTHER SEARCH TECHNIQUES AND/OR ITERATIVE PROCEDURES.
  C                 SOLUTION MAPPING IS USED TO MAP EACH INITIAL SOLUTION
  C                 TO AS MANY POINTS OF THE GRID AS POSSIBLE. THE CHARACTER-
  C                 ISTICS ARE STORED AS FUNCTIONS OF THE GRID PARAMETERS.
  C                 CONTOURS OF THESE FUNCTIONS ARE FOUND USING SECOND
  C                 ORDER LAGRANGE INTERPOLATION.
  C
  C INPUT
                    INCLUDE RUNDOC.LIST
                    INCLUDE SAVDOC.LIST
  C
  C                 NAME     TYPE/SIZE   DEFINITION
  C
  C                 IOS       INT/       PLOTTING OPTION SELECTOR
  C                                      =0, OFF-LINE PLOTS ONLY
  C                                      =1, PRINTER PLOTS ONLY
  C                                      =2, BOTH TYPES
  C                 FTLT      HOL/8      48 CHARACTER FRAME TITLE
  C                 ALAB      HOL/8      48 CHARACTER ABSCISSA LABEL
  C                 OLAB      HOL/8      48 CHARACTER ORDINATE LABEL
  C                 LABC      INT/       =1, LABEL CONTOURS OTHERWISE DO NOT
```

```
C              [illegible]     REA/        MINIMUM VALUE OF ABSCISSA
C              [illegible]     REA/        MAXIMUM VALUE OF ABSCISSA
C              [illegible]     REA/        MINIMUM VALUE OF ORDINATE
C              [illegible]     REA/        MAXIMUM VALUE OF ORDINATE
C              [illegible]     INT/        NUMBER OF FUNCTIONS ON THIS FRAME
C              [illegible]     INT/MU*     UNITS WHERE FUNCTION ARE TO BE FOUND
C              ISO             INT/MU*     SOLUTION NUMBERS OF FUNCTIONS
C              IVA             INT/MU*     VARIABLE NUMBERS OF FUNCTIONS
C              ICD             INT/MU*     CONTOUR DIRECTION INDICATOR
C                                          =0, FUNCTION CONSTANT CONTOURS
C                                          =1, P1 CONSTANT CONTOURS
C                                          =2, P2 CONSTANT CONTOURS
C              IAOS            INT/MU*     ABSCISSA-ORDINATE INDICATOR
C                                          =0, P1/P2,P2/F, OR P1/F IS THE
C                                          ABSCISSA/ORDINATE
C                                          =1, THE REVERSE IS TRUE
C              P1IN            INT/MU*     INTERPOLATION INCREMENTS FOR P1
C              P2IN            INT/MU*     INTERPOLATION INCREMENTS FOR P2
C              CMI             REA/MU*     MINIMUM CONTOUR IF IDC=0
C              CMA             REA/MU*     MAXIMUM CONTOUR IF IDC=0
C              CIN             REA/MU*     STEP SIZE BETWEEN CONTOURS IF IDC=0
C              INS             INT/MU*     IF IDC=J ,PJ(INS) IS THE FIRST
C                                          CONTOUR VALUE, J=1 OR 2.
C              LAS             INT/MU*     IF IDC=J ,PJ(LAS) IS THE LAST
C                                          CONTOUR VALUE, J=1 OR 2
C              ISI             INT/MU*     IF IDC=J ,PJ(INS+K*ISI),K=0,L WHERE
C                                          L*ISI.LE.LAS, ARE THE CONTOUR VALUES
C                                          PLOTTED, J=1 OR 2
C
C
C OUTPUT
               INCLUDE OPFMPC,LIST
               INCLUDE OPFMAP,LIST
C
C SUBROUTINES  NAME        PURPOSE
C
C              INISOL      TO FIND INITIAL SOLUTIONS
C              FILGRD      TO USE INITIAL SOLUTIONS FOUND TO FORM A GRID
C              COMCHA      TO COMPUTE THE SYSTEM CHARACTERISTICS AT EACH
C                          GRID POINT
C              AUXPRO      TO PROVIDE AN INTERFACE WITH OTHER PROGRAMS
C              PLTCON      TO PLOT CONTOURS OF THE SYSTEM CHARACTERISTICS
C
C SIMULATOR    THE USER MUST SUPPLY THE FOLLOWING SUBROUTINE ENTRY
C              POINTS:
C
C                  CONSTR(P,CS) WHICH COMPUTES THE VALUE OF THE
C                  CONSTRAINT FUNCTIONS(CS) AT P.
C
C                  CHARAC(P,NP,C,NC) WHICH COMPUTES THE SYSTEM
C                  CHARACTERISTICS(C) FOR THE GIVEN VALUE OF THE
C                  PARAMETERS(P). NP AND NC ARE THE NUMBER OF PARAMETERS
C                  AND CHARACTERISTICS RESPECTIVELY.
C
C                  PRTCHA(P,NP,C,NC) WHICH PRINTS P AND C WITH
C                  DESCRIPTIVE DETAILS MEANINGFUL TO THE USER
C
```

```
C               AUXPRO WHICH PERFORMS OPERATIONS USEFUL TO THE
C               CURRENT STUDY BUT NOT PROVIDED FOR BY PAP.
C
C            IN ADDITION TO THESE SUBROUTINE ENTRY POINTS, THE USER
C            MUST SUPPLY A PROCEDURE ELEMENT WITH THE FOLLOWING ENTRY
C            POINTS AND CONTENTS:
C
C               RUNDOC CONTAINS THE NAMES AND DEFINITIONS OF THE
C               VARIABLES IN NAMELIST RUNDAT.
C
C               SAVDOC CONTAINS THE NAMES AND DEFINITIONS OF THE
C               VARIABLES IN NAMELIST SAVDAT.
C
C               OPFMPC CONTAINS THE DESCRIPTION OF THE OUTPUT
C               RESULTING FROM A CALL TO PRTCHA.
C
C               OPFMAP CONTAINS THE DESCRIPTION OF THE OUTPUT
C               RESULTING FROM A CALL TO AUXPRO.
C
C               KOMUSE CONTAINS COMMON USE WHICH IS USED FOR COMMUNI-
C               CATION BETWEEN USER SUBROUTINES.
C
C               RUNNL CONTAINS THE NAMELIST RUNDAT, THIS NAMELIST
C               MUST CONTAIN THOSE INPUTS REVELENT TO THE CURRENT
C               RUN ONLY. INPUTS KON, IUNIT, ICHSA, AND NFR MUST BE
C               IN THIS NAMELIST. THIS NAMELIST IS NOT SAVED ON THE
C               SAVE FILES.
C
C               SAVNL CONTAINS THE NAMELIST SAVDAT WHICH MUST
C               CONTAIN THE INPUTS WHICH DEFINE THE CONSTRAINTS
C               AND THE SIMULATOR. THESE INPUTS WILL BE NEEDED IF
C               FURTHER PROCESSING IS DONE AND ARE WRITTEN ON ALL
C               SAVE FILES. NAMELISTS RUNDAT AND SAVDAT MUST CONTAIN
C               ALL INPUTS REQUIRED TO DEFINE THE SYSTEM, THE
C               CONSTRAINTS TO BE SATISFIED, AND THE INPUTS FOR PAP.
C
C               INITIL CONTAINS THE FORTRAN CODE THAT SETS THE
C               VALUES OF ANY VARIABLES THAT MUST BE INITILIZED
C               AT THE BEGINNING OF EACH PASS. VARIOUS INPUTS WHICH
C               ARE NOT OFTEN CHANGED CAN BE SET HERE. THEIR VALUE
C               CAN BE CHANGED BY THE SUBSEQUENT READ NAMELISTS
C               RUNDAT AND SAVEDAT.
C
C               SIMDFN CONTAINS THE FORTRAN CODE WHICH DERIVES FROM
C               THE INPUTS IN RUNDAT AND SAVDAT THE INFORMATION
C               NECESSARY TO DEFINE THE SYSTEM CONFIGURATION, THE
C               CONSTRAINTS TO BE SATISFIED, AND THE INPUTS TO PAP.
C
C               RPDOS CONTAINS A DIMENSION STATEMENT FOR ALL
C               VARIABLES IN NAMELIST SAVDAT. THE DIMENSION AND SIZE
C               OF THESE VARIABLES MUST CORRESPOND TO THOSE GIVEN
C               TO THE VARIABLES IN PAP.
C
C
C       SPECIFICATION STATEMENTS
C
C
```

```
      INCLUDE PARSTA,LIST
      INCLUDE PARDOC,LIST
      INCLUDE KOMMCL,LIST
      INCLUDE MCLDOC,LIST
      INCLUDE KOMUSE,LIST
      INCLUDE RUNNL,LIST
      INCLUDE SAVNL,LIST
      DIMENSION IUN(MU),ISO(MU),IVA(MU),CMI(MU),CMA(MU),
     .CIN(MU),ICD(MU),INS(MU),LAS(MU),ISI(MU),FTLT(8),ALAB(8),
     .OLAB(8),IAOS(MU),P1IN(MU),P2IN(MU)
      DIMENSION TIME(2)
      DATA SUB1/6HINISOL/SUB2/6HFILGRD/SUB3/6HCOMCHA/SUB4/
     .6HAUXPRO/SUB5/6HPLTCON/
C
C
C     NAMELIST PLTDAT-THIS NAMELIST DEFINES ONE PLOTTING
C     FRAME. WHEN SPECIFYING NFR FRAMES, THIS NAMELIST WILL
C     BE READ NFR TIMES IN SEQUENCE STORING THE INPUTS BY
C     FRAMES.
C
      NAMELIST/PLTDAT/IOS,IUN,ISO,IVA,FTLT,ALAB,OLAB,LABC,
     .AMI,AMA,OMI,OMA,CMI,CMA,IAOS,P1IN,P2IN,CIN,ICD,INS,LAS,
     .ISI,NEU
      EQUIVALENCE (J,JX)
C
C     INITILIZE
C
      INCLUDE INITIL,LIST
C
C     READ NAMELIST RUNDAT
C
      READ(5,RUNDAT)
C
C     READ NAMELIST SAVDAT FROM FILE IUNIT, 5, OR BOTH (5 IS READ LAST)
C
      IF(KON(1).EQ.1) GO TO 10
      REWIND IUNIT
      READ(IUNIT,SAVDAT)
      IF(ICHSA.NE.1)GO TO 20
   10 READ(5,SAVDAT)
C
C     READ PLOT DATA IF NFR GT 0
C
   20 IF(NFR.EQ.0) GO TO 30
C
C     READ INPUTS THAT DEFINE A SINGLE FRAME AND STORE IN ARRAYS WHICH
C     HAVE ONE OF THEIR SUBSCRIPTS CORRESPONDING TO THE FRAME NUMBER
C
      DO 25 J=1,NFR
      READ(5,PLTDAT)
      IPCP(J)=IOS
      LAC(J)=LABC
      AMIN(J)=AMI
      AMAX(J)=AMA
      OMIN(J)=OMI
      OMAX(J)=OMA
      DO 22 L=1,8
```

```
      TLT(L,J)=FTLT(L)
      ALA(L,J)=ALAR(L)
22    OLA(L,J)=OLAR(L)
      NFP(J)=NFU
      DO 25 K=1,NFU
      IFLU(J,K)=IUN(K)
      ISOL(J,K)=ISO(K)
      IVAR(J,K)=IVA(K)
      IAO(J,K)=IAOS(K)
      P1INC(J,K)=P1IN(K)
      P2INC(J,K)=P2IN(K)
      CMIN(J,K)=CMI(K)
      CMAX(J,K)=CMA(K)
      DELC(J,K)=CIN(K)
      IVS(J,K)=ICD(K)
      IS(J,K)=INS(K)
      LS(J,K)=LAS(K)
25    ICS(J,K)=ISI(K)
30    CONTINUE
      INCLUDE SIMDEN,LIST
C
C     POSITION PLOT FILES BEING USED PAST USRDAT
C
      IF(NF14.NE.1.OR.NF14.EQ.IUNIT) GO TO 101
      REWIND 14
      CALL RPD(14)
101   IF(NF15.NE.1.OR.NF15.EQ.IUNIT) GO TO 102
      REWIND 15
      CALL RPD(15)
102   IF(NF16.NE.1.OR.NF16.EQ.IUNIT) GO TO 103
      REWIND 16
      CALL RPD(16)
103   IF(NF17.NE.1.OR.NF17.EQ.IUNIT) GO TO 104
      REWIND 17
      CALL RPD(17)
104   IF(NF18.NE.1.OR.NF18.EQ.IUNIT) GO TO 105
      REWIND 18
      CALL RPD(18)
C
C     WRITE SAVDAT ON FILES BEING SAVED THIS PASS
C
105   IF(ISAI.EQ.0) GO TO 106
      LOI=8+NOI
      NOI=NOI+1
      IF(LOI.GT.10) GO TO 190
      REWIND LOI
      WRITE(LOI,SAVDAT)
106   IF(ISAF.EQ.0) GO TO 107
      LOF=11+NOF
      NOF=NOF+1
      IF(LOF.GT.13) GO TO 190
      REWIND LOF
      WRITE(LOF,SAVDAT)
107   IF(ISAC.EQ.0) GO TO 108
      LOC=14+NOC
      NOC=NOC+1
      IF(LOC.GT.16) GO TO 190
```

```
      REWIND LOC
      WRITE(LOC,SAVDAT)
  108 IF(ISAA.EQ.0) GO TO 109
      LOA=17+NOA
      NOA=NOA+1
      IF(LOA.GT.16) GO TO 190
      REWIND LOA
      WRITE(LOA,SAVDAT)
  109 CONTINUE
C
C     GO TO THE FUNCTION TO BE PERFORMED NEXT
C
      J=1
  120 KN=KON(J)
      GO TO (130,140,150,160,170),KN
C
C     CALL INISOL
C
  130 CALL OCLOCK(TIME)
      WRITE(6,1002)SUB1,TIME
      NSE=0
      CALL INISOL
      WRITE(6,1003)NSE,SUB1
      GO TO 180
C
C     CALL FILGRD
C
  140 CALL OCLOCK(TIME)
      WRITE(6,1002)SUB2,TIME
      NSE=0
      CALL FILGRD
      WRITE(6,1003)NSE,SUB2
      GO TO 180
C
C     CALL COMCHA
C
  150 CALL OCLOCK(TIME)
      WRITE(6,1002)SUB3,TIME
      CALL COMCHA
      GO TO 180
C
C     CALL AUXPRO
C
  160 CALL OCLOCK(TIME)
      WRITE(6,1002)SUB4,TIME
      CALL AUXPRO(NAUX)
      GO TO 180
C
C     CALL PLTCON
C
  170 CALL OCLOCK(TIME)
      WRITE(6,1002)SUB5,TIME
      CALL PLTCON
  180 J=J+1
      IF(KON(J)-6) 120,1,200
C
C     WRITE EXIT REASON MESSAGE
```

```
C
 190          WRITE(6,1001) NOI,NOF,NOC,NOA
C
C     CALL EXIT
C
 200          CALL EXIT
C
C     FORMAT STATEMENTS
C
 1001         FORMAT(1H ,'****MORE THAN THE ALOTTED NUMBER OF SAVE '
             .'FILES WERE USED','  NOI = ',I3,2X,'NOF = ',I3,2X,'NOC = ',
             .I3,2X,'NOA = ',I3/)
 1002         FORMAT(1H0,'*****CALL ',A6,' AT TIME ',2A6,'*****'/)
 1003         FORMAT(1H0,'*****SYSTEM EVALUATED ',I4,' TIMES IN ',A6,
             .'*****'/)
              END
```

```
INTAP.PERXFR
EVEL      3
   C
   C  TITLE          PERIAPSE TRANSFER
   C
   C  AUTHOR         R.W.GRUTZNER
   C
   C  SPONSOR        A.A.VANDERVEEN
   C
   C  DATE           5-12-68
   C
   C  PURPOSE        COMPUTE ECCENTRICITIES OF TWO HYPERBOLII GIVEN THEIR
   C                 SEMI-TRANSVERSE AXES SUCH THAT A PERIAPSE TRANSFER
   C                 IS ACHEIVED.
   C
   C  METHOD         NEWTON-RAPHSON ITERATION TO FIND THE REAL ROOT OF
   C                 A FOURTH ORDER EQUATION IN (1./ECC)
   C
   C  INPUT          THROUGH LIST
   C                      BA               BEND ANGLE
   C                      A2(2)            SEMI-TRANSVERSE AXES OF HYPERBOLII
   C
   C  OUTPUT         THROUGH LIST
   C                      E(2)             ECCENTRICITIES OF HYPERBOLII
   C                      RTRANS           PERIAPSE TRANSFER RADIUS
   C
                 SUBROUTINE PERXFR(A2,BA,E,RTRANS)
   C
                 COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                 DIMENSION A2(2),E(2)
   C
                 A=(A2(1)-A2(2))*COS(BA)
                 B=-(A2(1)*COS(BA)+A2(2))
                 C=A2(1)*SIN(BA)
                 D=-(A2(1)-A2(2))*SIN(BA)
   C
                 C4PRIM=-(D**2+A**2)
                 C4=1.
                 C3=-2.*(C*D+A*B)/C4PRIM
                 C2=(D**2-B**2-C**2)/C4PRIM
                 C1=2.*C*D/C4PRIM
                 C0=C**2/C4PRIM
   C
   C     INITIAL GUESS THETA=45 DEGREES
   C
                 DELX=0.
                 THETA=45.*DTR
                 X=SIN(THETA)
                 DO 1 IJ=1,20
                 X=X+DELX
                 FOFX=X*(X*(X*(X*C4+C3)+C2)+C1)+C0
                 IF(ABS(FOFX).LT.(.0001)) GO TO 2
                 SLOPE=X*(X*(4.*C4*X+3.*C3)+2.*C2)+C1
                 DELX=-(FOFX/SLOPE)
   1             CONTINUE
   2             THETA1=ASIN(X)
                 THETA2=BA-THETA1
```

```
E(1)=1./SIN(THETA1)
E(2)=1./SIN(THETA2)
RTRANS=A2(1)*(E(1)-1.)
RETURN
END
```

```
INTAP.PRTCHA
EVEL      3
   C
   C  TITLE          PRINT CHARACTERISTICS
   C
   C  AUTHOR         R.W.GRUTZNER
   C
   C  SPONSOR        A.A.VANDERVEEN
   C
   C  DATE           6-11-68
   C
   C  PURPOSE        PRINT THE PARAMETERS AND CHARACTERISTICS ARISING
   C                 FROM EACH GRID POINT
   C
   C
   C  CALL           CALL PRTCHA(P,NP,C,NC)
   C
   C  INPUT          THROUGH LIST
   C                 P          VECTOR OF PARAMETERS
   C                 NP         NUMBER OF PARAMETERS
   C                 NC         NUMBER OF CHARACTERISTICS
   C
   C  OUTPUT         THROUGH LIST
   C                 C          VECTOR OF CHARACTERISTICS
   C
   C        SUBROUTINE STATEMENT
   C
                     SUBROUTINE PRTCHA(P,C)
   C
   C        SPECIFICATION STATEMENTS
                     INCLUDE PARSTA,LIST
                     INCLUDE KOMUSE,LIST
                     DIMENSION C(MK),P(MP)
   C
                     IF(IPRINT.NE.0) GO TO 6
   C
   C        WRITE SHORT RECORD
   C
                     IF(IFIRST.NE.0) GO TO 1
                     WRITE(6,2003)
                     IFIRST=1
                     LINES=4
    1                I=1
                     J=5
                     DO 4 IJ=1,NPONTS
                     IF(IENC(IJ).EQ.2) GO TO 2
                     WRITE(6,2000) (C(K),K=I,J),(C(J+K),K=2,4)
                     GO TO 3
    2                WRITE(6,2001) (C(K),K=I,J),(C(J+K),K=2,4),C(J+15)
    3                I=I+LENGTH(IJ)
                     J=J+LENGTH(IJ)
    4                CONTINUE
                     LINES=LINES+NPONTS
                     IF((LINES+NPONTS+1).LE.54) GO TO 5
                     IFIRST=0
                     RETURN
    5                WRITE(6,2004)
                     LINES=LINES+1
```

```
C         WRITE LONG RECORD
C
C         ...WRITE LAMBERT AND ASYMPTOTE DATA
C
  6                 L=0
                    WRITE(6,1001)
                    DO 10 I=1,LEGS
                    IF(LAMBERT(I).EQ.0) GO TO 10
                    L1=L+LAMBCR(I)
                    L2=L+LENGTH(I)
                    WRITE(6,1002) C(L1+2),C(L+1),C(L1+5),C(L1+6),C(L1+3),
          .         C(L1+4),C(L+3),C(L+9),C(L+7),C(L+6)
                    WRITE(6,1102)C(L1+1),C(L2+1),C(L1+9),C(L1+10),C(L1+7),
          .         C(L1+8),C(L2+4),C(L2+5),C(L2+3),C(L2+2)
                    L=L2
 10                 CONTINUE
C
C         ...WRITE ENCOUNTER DATA
C
                    L=0
                    DO 100 I=1,NPONTS
                    IENTYP=IENC(I)
                    IF(IENTYP.EQ.0) GO TO 100
                    GO TO (20,30,40,50,60,20,80,20), IENTYP
C
C         ...WRITE DEPARTURE, ARRIVAL, OR DEBOOST DATA
C
 20                 IF(IENTYP.EQ.1) WRITE(6,1[illegible])
                    IF(IENTYP.EQ.6) WRITE(6,1011)
                    IF(IENTYP.EQ.8) WRITE(6,1010)
                    WRITE(6,1003)
                    WRITE(6,1004)(C(L+K),K=13,15),C(L+10),C(L+23),
          .         (C(L+K),K=28,37),C(L+24)
                    WRITE(6,1005)
                    WRITE(6,1006)(C(L+K),K=16,22),(C(L+K),K=25,27)
                    L=L+LENGTH(I)
                    GO TO 100
C
C         ...WRITE BALLISTIC FLYBY DATA
C
 30                 WRITE(6,1007)
                    WRITE(6,1008) C(L+1),C(L+[illegible]),(C(L+K),K=[illegible],23),C(L+11),
          .         C(L+12)
                    L=L+LENGTH(I)
                    GO TO 100
C
C         ...WRITE SINGLE IMPULSE FLYBY DATA
C
 40                 WRITE(6,1013)
                    WRITE(6,1014)
                    WRITE(6,1015)(C(L+K),K=13,15),C(L+10),(C(L+K),K=16,23)
                    WRITE(6,1016)
                    WRITE(6,1017)(C(L+K),K=24,30)
                    L=L+LENGTH(I)
                    GO TO 100
```

```
C
C        ...UNASSIGNED OPTIONS
C
 50               GO TO 100
 60               GO TO 100
C
C        ...WRITE PLANE CHANGE MANEUVER DATA
C
 80               WRITE(6,1020)
                  WRITE(6,1021) C(L+10),(C(L+K),K=13,16)
                  L=L+LENGTH(I)
C
 100              CONTINUE
                  RETURN
C
C        FORMAT STATEMENTS
C
 1001             FORMAT(1H1,16HTRANS ORBIT CHAR,22X,17HPLANET   THETA    ,
     .            53HINC     HE VEL   DP ANG   R.ASCN  DECLIN  H.E.S.    DATE/
     .            48X,47H(DEG)   (DEG)    (EMOS)    (DEG)    (DEG)    (DEG)   ,
     .            16H(EMOS)   (J.DATE)/)
 1002             FORMAT(1H ,9H ECCEN = ,F8.5,11X,6HDEPART,6X,F3.0,F8.2,
     .            F8.2,F8.4,3F8.2,F8.4,F11.2)
 1102             FORMAT(1H ,9H S.M.A.= ,F8.5,11X,6HARRIVE,6X,F3.0,F8.2,
     .            F8.2,F8.4,3F8.2,F8.4,F11.2/)
 1003             FORMAT(1H ,2X,39HDEL.PSI   PSI 1    PSI 2   B.ANGLE  P.ALT
     .            48H V-CIRC  VP-ELL  VP-HY 1 VP-HY 2 DV-C 1  DV-C 2
     .            42H DVC-TOT DV-E 1  DV-E 2  DVE-TOT PER. RAD./4X,
     .            52H(DEG)    (DEG)    (DEG)    (DEG)    (NM)    (FPS)    (FPS)
     .            49H    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)    (FPS)
     .            27H    (FPS)    (FPS)   (PL.RAD.))
 1004             FORMAT(1H ,1X,5F8.2,10F8.1,3X,F7.4)
 1005             FORMAT(1H0,1X,38H INCLIN  C.OMEGA OMEGA 1 OMEGA 2 OMEGA
     .            42H P ALPHA P DELTA P  A/RP   SMA.ELL APO-VEL/4X
     .            53H(DEG)   (DEG)    (DEG)   (DEG)    (DEG)    (DEG)    (DEG)
     .            24H   (N.DIM)   (NM)     (FPS))
 1006             FORMAT(1H ,7F8.2,F8.4,F8.0,F8.1)
 1007             FORMAT(1H0,'FLYBY CHARACTERISTICS',90X,'SUB-SOLAR PT.'/
     .            1X,46HPLANET BEND ANG  OMEGA    INCL    OMEGA1  OMEGA2
     .            43H  OMEGAP  ALPHAP  DELTAP  PER RAD   PER ALT
     .            35H   PER VEL  DELTA V   R.ASC   DECLIN/10X,
     .            45H(DEG)    (DEG)    (DEG)    (DEG)    (DEG)    (DEG)
     .            45H    (DEG)    (DEG)   (P.RAD)     (N.M.)   (FT/SEC)
     .            25H    (EMOS)    (DEG)    (DEG))
 1008             FORMAT(1H ,1X,F3.0,F10.2,7F8.2,F9.4,2F10.2,F9.4,2F8.2)
 1009             FORMAT(1H ,25HDEPARTURE CHARACTERISTICS)
 1010             FORMAT(1H0,23HARRIVAL CHARACTERISTICS)
 1011             FORMAT(1H0,24HSTOPOVER CHARACTERISTICS)
 1012             FORMAT(1H1)
 1013             FORMAT(1H0,31HIMPULSIVE FLYBY CHARACTERISTICS)
 1014             FORMAT(1H ,2X,42HDEL.PSI   PSI 1    PSI2   B.ANGLE    ALTOPT
     .            24HTR.OPT   DVOPT    GAMMV
     .            30HTHET1    THET2    VT-IN    VT-OUT/4X,
     .            40H(DEG)    (DEG)    (DEG)    (DEG)      (NM)
     .            33H(PL.RAD) (FPS)    (DEG)    (DEG)
     .            21H(DEG)    (FPS)    (FPS))
 1015             FORMAT(1H ,1X,4F8.2,F9.2,F8.4,F8.1,3F8.2,2F8.1)
```

```
 1016        FORMAT(1H0,1X,38H INCLIN  C.OMEGA OMEGA 1 OMEGA 2 OMEGA
     .       20H T  ALPHA T  DELTA T/4X,
     .       40H(DEG)   (DEG)   (DEG)   (DEG)   (DEG)
     .       15H (DEG)    (DEG))
 1017        FORMAT(1H ,5F8.2,1X,F8.2,1X,F8.2)
 1020        FORMAT(1H0,'PLANE CHANGE MANEUVER'/3X,
     .       'B.ANGLE   ALPHA V   DECLIN V   DEL.INC.    '
     .       'DELTAV'/4X,'(DEG)      (DEG)      (DEG)       (DEG)      '
     .       '(EMOS)')
 1021        FORMAT(1H ,1X,F8.2,F10.2,F11.2,F10.2,F10.4)
C
C      FORMAT STATEMENTS
C
 2000        FORMAT(F6.1,F11.2,F14.4,2F10.2,F14.4,2F10.2)
 2001        FORMAT(F6.1,F11.2,F14.4,2F10.2,F14.4,2F10.2,F14.4)
 2003        FORMAT(1H1,29X,'INBOUND ASYMPTOTE'16X,
     .       'OUTBOUND ASYMPTOTE'/' PLANET'4X,'TIME'10X,'H.E.S.'4X,
     .       'R.ASC.'4X,'DECLIN',8X,'H.E.S.'4X,'R.ASC.'4X,'DECLIN'
     .       5X,'PASSAGE RAD.'/25X,'(EMOS)'4X,'(DEG)'6X,'(DEG)'8X,
     .       '(EMOS)'4X,'(DEG)'5X,'(DEG)'8X,'(P.RAD)'/)
 2004        FORMAT(1H ,'*********************************************'
     .       '*****************************************************')
C
             END
```

```
INTAP.RADMAG
EVEL       3
   C
   C  TITLE          RADIUS MAGNITUDE
   C
   C  AUTHOR         R.W.GRUTZNER
   C
   C  SPONSOR        A.A.VANDERVEEN
   C
   C  DATE           4-30-67
   C
   C  PURPOSE        COMPUTE RADIUS MAGNITUDE AND DECOMPOSE ALONG PERIHELION
   C                 AND SEMI-LATUS RECTUM DIRECTIONS
   C
   C  INPUT          THROUGH LIST
   C                       A                   SEMI-MAJOR AXIS
   C                       E                   ECCENTRICITY
   C                       ECAN                ECCENTRIC ANOMALY
   C                       TRNM                TRUE ANOMALY
   C
   C  OUTPUT         THROUGH LIST
   C                       RADIUS              MAGNITUDE OF RADIUS
   C                       RCOMP(2)            COMPONENTS OF RADIUS
   C
          SUBROUTINE RADMAG(A,E,ECAN,TRNM,RADIUS,RCOMP)
   C
                 DIMENSION RCOMP(3)
   C
                 RADIUS=A*(1.0-E*COS(ECAN))
                 RCOMP(1)=RADIUS*COS(TRNM)
                 RCOMP(2)=RADIUS*SIN(TRNM)
                 RCOMP(3)=0.
                 RETURN
                 END
```

```
INTAR.RPD
EVEL      3
    C TITLE         RPD - READ PAST NAMELIST USRDAT
    C
    C AUTHOR        R.F. LONG
    C
    C DATE          7/26/69
    C
    C PURPOSE       TO POSITION READ HEADS PAST NAMELIST USRDAT WITHOUT
    C               PUTTING THE DATA INTO THE MAIN PROGRAM
    C
    C METHOD        READ NAMELIST USRDAT BUT DO NOT TRANSMIT DATA TO PAP
    C
    C INPUT         THROUGH ARGUMENT LIST
    C
    C               IFILE      FILE NUMBER TO BE READ
    C
    C OUTPUT        NONE
    C
    C       SUBROUTINE STATEMENT
    C
                    SUBROUTINE RPD(IFILE)
    C
    C       SPECIFICATION STATEMENT
    C
                    INCLUDE PARSTA,LIST
                    INCLUDE SAVNL,LIST
                    INCLUDE RPDDS,LIST
    C       READ SAVNL FROM IFILE
    C
                    READ(IFILE,SAVDAT)
                    RETURN
                    END
```

```
ST    ACCOUNT: AAV                PROJECT: VNDRVEEN

:05.624     IN: 33      OUT: 0          PAGES:    30

 TIME:    12:52:57-OCT 30,1969

 TIME:    12:54:46-OCT 30,1969

S: 62

   691

 1.989
```

```
INTAP.SETUP
IVEL     5
   C
   C  TITLE          [illegible] POINT ARRAY
   C
   C  AUTHOR         [illegible]
   C
   C  DATE           [illegible]
   C
   C  PURPOSE        CONVERT THE TIME POINT ARRAY TO A TOTAL HELIOCENTRIC
   C                 VECTOR
   C
   C  METHOD         VECTOR ADDITION
   C
   C
   C        SUBROUTINE [illegible]
   C
                     SUBROUTINE SETUP(P)
   C
                     [illegible] PARSTA,LIST
                     [illegible] ARGUSE,LIST
                     [illegible] P(MP)
   C
   C        TPA [illegible] REAPUT WITH EXCEPTION OF THE PARAMETER VALUES
   C        WHICH ARE [illegible]TTED THROUGH ARGUMENT LIST 'P'. P(1) IS PRESENT
   C        VALUE OF [illegible] VAR. AND P(2) IS ORDINATE VAR. SET PARAMETERS
   C        INTO [illegible] ARRAY AND COMPUTE VECTOR POSITIONS BY CALLING
   C        EITHER [illegible] OR FIXREL.
   C
                     [illegible]
                     [illegible] J=1,NPUSE
                     [illegible]=ITY(J)
                     [illegible]=ITP(J)
                     [illegible]VR=10*ISUB2+ISUB1
                     [illegible]ORDVR.EQ.IP1) GO TO 1
                     [illegible]ORDVR.EQ.IP2) GO TO 2
                     [illegible](ISUB1,ISUB2)=P(I)
                     [illegible]
                     GO TO 3
    1                [illegible](ISUB1,ISUB2)=P(1)
                     GO TO 3
    2                [illegible](ISUB1,ISUB2)=[illegible](2)
    3                I[illegible](ISUB1.NE.2) GO TO 4
                     T=[illegible](2,ISUB2)
                     CALL FIXPLC(ISUB2,T)
                     GO TO 5
    4                CALL FIXREL(ISUB2)
    5                CONTINUE
   C
   C        COMPUTE [illegible] POSITION VECTOR
   C
                     [illegible] IJ=1,[illegible]
                     [illegible] 6 IK=1,3
    6                [illegible](IK,IJ)=PLCVEC(IK,IJ)+RELVEC(IK,IJ)
                     [illegible]
                     [illegible]
```

```
INTAP.SLOPE
EVEL      3
   C
   C  TITLE          TANGENT TO ELLIPSE
   C
   C  AUTHOR         R.W.GRUTZLER
   C
   C  SPONSOR        A.A.VANDERVEEN
   C
   C  DATE           4-30-67
   C
   C  PURPOSE        COMPUTE ANGLE BETWEEN EXTENDED RADIUS AND TANGENT
   C                 TO ELLIPSE (VELOCITY DIRECTION)
   C
   C  METHOD         EVALUATE LERIVATIVE (DYDX), COMPUTE ARCTAN DYDX AND
   C                 MODIFY BY TRUE ANOMALY
   C
   C  INPUT          THROUGH COMMON
   C                       PI
   C                       TWOPI
   C                 THROUGH LIST
   C                       A                SEMI-MAJOR AXIS
   C                       E                ECCENTRICITY
   C                       RAD              RADIUS MAGNITUDE
   C                       TRNM             TRUE ANOMALY
   C
   C  OUTPUT         THROUGH LIST
   C                       GAMMA            ANGLE BETWEEN EXTENDED RADIUS AND
   C                                        VELOCITY VECTOR
   C
         SUBROUTINE SLOPE(A,E,RAD,TRNM,GAMMA)
   C
                 COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
   C
   C
   C     CHECK FOR PARABOLIC CASE
   C
                 IF(ABS(E-1.).GT.(10.**(-7))) GO TO 1
                 P=RAD*(1.+COS(TRNM))
                 SINGAM=SQRT(P/(2.*RAD))
                 GO TO 2
     1           P=A*(1.-E*E)
                 SINGAM=SQRT((P/RAD)/(2.-RAD/A))
     2           GAMMA=ASIN(SINGAM)
                 TR=AMOD(TRNM,TWOPI)
                 IF(TR.GT.PI) GAMMA=PI-GAMMA
                 RETURN
                 END
```

```
INTAP...
...
      ... T COORDINATE
...
 ... RIGHT ASCENSION AND DECLINATION OF THE VECTOR
 ... TO THE SUN AT THE PASSAGE DATE
 ... POSITION AT SAID DATE AND TRANSFORM
 ... TO PLANET'S EQUATORIAL SYSTEM

                     PLANET NUMBER
                     ENCOUNTER DATE

                     DECLINATION OF SUB-SOLAR POINT
                     RIGHT ASCENSION OF SUB-SOLAR POINT

      ... LIST
      ... LIST
      ...HAFPI,...,...,PTD,DTR
      ...,COMP(3),...

 ... AT TIME

      ...)
      ... TIMEIN,TP,...)
      ...,ECAN)
      ...,TRUM)
      ...(PLNO)
      ... (C...,LAMDA,...)
      ...
      ...0.
      ...0.
 COMP(1)...V(ELEM,R)
 ...(3))
 ...COMP(2),COMP(1))
```

```
INTAP.TRUANM
EVEL      3
   C
   C TITLE          TRUE ANOMALY
   C
   C AUTHOR         R.W.GRUTZNER
   C
   C SPONSOR        A.A.VANDERVEEN
   C
   C DATE           4-30-67
   C
   C PURPOSE        CONVERT FROM ECCENTRIC TO TRUE ANOMALY
   C
   C INPUT          THROUGH COMMON
   C                     PI
   C                THROUGH LIST
   C                     ECAN               ECCENTRIC ANOMALY
   C                     E                  ECCENTRICITY
   C
   C OUTPUT         THROUGH LIST
   C                     TRNM               TRUE ANOMALY
   C
         SUBROUTINE TRUANM(ECAN,E,TRNM)
   C
                    COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
   C
                    TANF2=SQRT((1.0+E)/(1.0-E))*TAN(ECAN/2.0)
                    TRNM=2.0*ATAN(TANF2)
                    IF(TANF2.GT.0.0) GO TO 10
                    TRNM=2.0*PI+TRNM
    10              RETURN
                    END
```

```
INTAP.VASTOT
EVEL      3
      C
      C TITLE           VIRTUAL ASYMPTOTE
      C
      C AUTHOR          ...KRITZNER
      C
      C SPONSOR         ...VANDERVEEN
      C
      C DATE            6-29-67
      C
      C PURPOSE         GIVEN EITHER INBOUND OR OUTBOUND HYP.EXC.VELOCITY
      C                 VECTOR, THE PASSAGE RADIUS AND THE DECLINATION OF
      C                 PERIAPSE THE PROGRAM FINDS THE VIRTUAL SYMMETRIC ASYMPT.
      C
      C METHOD          FIRST FIND THE PERIAPSE VECTOR FROM A KNOWLEDGE OF
      C                 ECCENTRICITY AND PERIAPSE DECLINATION. THE VIRTUAL
      C                 ASYMPTOTE IS THEN THE SUM OF (-1./ECC) TIMES THE PERIAPSE
      C                 VECTOR PLUS (V CROSS PERIAPSE) CROSS PERIAPSE
      C
      C CALL            CALL VASTOT(IENTYP)
      C
      C
      C INPUT           THROUGH COMMON
      C                      VIN               MAGNITUDE OF INBOUND H.E.V.
      C                      RASIN             RIGHT ASCENSION OF INBOUND H.E.V.
      C                      DECLIN            DECLINATION OF INBOUND H.E.V.
      C                      PRAD             PASSAGE RADIUS
      C                      DECLP            DECLINATION OF PERIAPSE
      C                 THROUGH LIST
      C                      IENTYP           ENCOUNTER TYPE
      C OUTPUT          THROUGH COMMON
      C                      VOUT              MAGNITUDE OF OUTBOUND H.E.V.
      C                      RASOUT            RIGHT ASCENSION OF OUTBOUND H.E.V.
      C                      DECOUT            DECLINATION OF OUTBOUND H.E.V.
      C                 NOTE - H.E.V. COMPONENTS ARE INTERCHANCEABLE FOR INPUT
      C                           AND OUTPUT
      C
      C SUBROUTINES
      C                      CROSS            COMPUTE CROSS PRODUCT
      C
                        SUBROUTINE VASTOT(IENTYP)
      C
      C        SPECIFICATION STATEMENTS
      C
                        INCLUDE KOMPEN,LIST
                        INCLUDE KOMPLN,LIST
                        COMMON/CONFAC/PI,HAFPI,TWOPI,GMSUN,RTD,DTR
                        DIMENSION V(3),PERIAP(3),VCROSR(3),VO(3),VEC1(3),VEC2(3)
      C
      C        DETERMINE WHETHER INBOUND OR OUTBOUND ASYMPTOTE IS SPECIFIED
      C
                        ITEST=0
                        IF(IENTYP.EQ.1) GO TO 20
      C
      C        ...INBOUND IS SPECIFIED
      C
```

```
                VEL=VIN
                ALPHA=RASIN+PI
                BETA=-DECLIN
                GO TO 10
C
C     ...OUTBOUND IS SPECIFIED
C
 20             VEL=VOUT
                ALPHA=RASOUT
                BETA=DECOUT
C
C     COMPUTE ORBITAL PARAMETERS
C
 10             VEL=VEL*97702.1
                A2=AMU(IPLND)/VEL**2
                E=1.+PRAD/A2
C
C     COMPUTE COMPONENTS OF GIVEN UNIT VECTOR
C
                V(1)=COS(BETA)*COS(ALPHA)
                V(2)=COS(BETA)*SIN(ALPHA)
                V(3)=SIN(BETA)
C
C     SET UP COEFFICIENTS FOR EQUATION OF THE FORM C1*COS(THETA)+
C     C2*SIN(THETA)=C3 AND SOLVE FOR RIGHT ASCENSION OF PERIAPSE
C
                C1=V(1)*COS(DECLP)
                C2=V(2)*COS(DECLP)
                C3=-V(3)*SIN(DECLP)-1./E
C
C     CHECK FEASIBILITY OF DESIRED PERIAPSE DECLINATION AND
C     RESET VALUE IF NOT REALIZABLE.
C
                RAD=C1**2+C2**2-C3**2
                IF(RAD.GT.0.) GO TO 11
                ITEST=1
                DECLP=SIGN((PI-ACOS(-1./E)),BETA)-BETA
                ALPHP=ALPHA+PI
                GO TO 13
 11             DISC=SQRT(RAD)
                ROOT1=(C1*C3+C2*DISC)/(C1**2+C2**2)
                ROOT2=(C1*C3-C2*DISC)/(C1**2+C2**2)
                ALPHP=ACOS(ROOT1)
                IF(ABS((C3-C1*COS(ALPHP))/C2-SIN(ALPHP)).GT.(.0001))
                                    ALPHP=TWOPI-ALPHP
C
C     COMPUTE COMPONENTS OF UNIT PERIAPSE VECTOR
C
 13             PERIAP(1)=COS(DECLP)*COS(ALPHP)
                PERIAP(2)=COS(DECLP)*SIN(ALPHP)
                PERIAP(3)=SIN(DECLP)
                VCROSP(1)=VCROSS(V,PERIAP)
                IF(ITEST.EQ.1) GO TO 14
C
C     ...THIS TEST PROVIDE FOR TRAVEL AROUND PLANET IN SAME
C        DIRECTION AS ROTATION, ALL CCW AS VIEWED FROM NORTH
```

```
      IF(((VCROSR(3).GT.0.).AND.(IENTYP.EQ.0)).OR.((VCROSR(3)
     .       .LT.0.).AND.(IENTYP.EQ.1))) GO TO 14
      ALPHP=ARCOS(ROOT2)
      IF(ABS((C3-C1*COS(ALPHP))/C2-SIN(ALPHP)).GT.(.0001))
     .                ALPHP=TWOPI-ALPHP
                      GO TO 13
C
C     FIND VIRTUAL ASYMPTOTE
C
 14   AMULT=-1./C
      VEC1(1)=VCROSS(VCROSR,PERIAP)
      VEC2(1)=VSCALR(PERIAP,AMULT)
      VO(1)=VADD(VEC1,VEC2)
C
C     COMPUTE DECLINATION AND RIGHT ASCENSION OF VIRTUAL VECTOR
C
      DEL=ARSIN(VO(3))
      ALP=ARCOS(VO(1)/(SQRT(VO(1)**2+VO(2)**2)))
      IF(VO(2).LT.(0.)) ALP=TWOPI-ALP
      IF(IENTYP.EQ.1) GO TO 18
      VOUT=VIN
      DECOUT=DEL
      RASOUT=ALP
      RETURN
 18   VIN=VOUT
      DECLIN=-DEL
      RASIN=ALP+PI
      RETURN
      END
```

```
INTAP.VELMAG
EVEL      3
   C
   C TITLE         VELOCITY MAGNITUDE
   C
   C AUTHOR        R.W.GROTZER
   C
   C SPONSOR       A.A.VAN DE VEEN
   C
   C DATE          4-30-67
   C
   C PURPOSE       COMPUTE VELOCITY MAGNITUDE
   C
   C INPUT         THROUGH LIST
   C                     RAD              RADIUS MAGNITUDE
   C                     A                SEMI-MAJOR AXIS
   C                     TRNM             TRUE ANOMALY
   C                     GAMMA            FLIGHT PATH ANGLE
   C
   C OUTPUT        THROUGH LIST
   C                     VCOMP(3)         VELOCITY COMPS. ALONG AND NORMAL TO
   C                                      PERIHELION DIRECTION
   C
         SUBROUTINE VELMAG(A,RAD,TRNM,GAMMA,VCOMP)
   C
               DIMENSION VCOMP(3)
   C
               VEL=SQRT((2.0/RAD)-(1.0/A))
               VCOMP(1)=VEL*COS(TRNM+GAMMA)
               VCOMP(2)=VEL*SIN(TRNM+GAMMA)
               VCOMP(3)=0.
               RETURN
               END
```