

N71-34991

BELLCOMM, INC.

955 L'ENFANT PLAZA NORTH, S.W.

WASHINGTON, D. C. 20024

B71 03073

SUBJECT: The Univac 1108 Virtual Mass  
Trajectory Simulation Program  
Case 103-9

DATE: March 30, 1971

FROM: R. F. Jessup

ABSTRACT

The Virtual Mass trajectory simulation program, VMASS, is a system of Fortran subroutines designed to aid in the analysis of spacecraft trajectories. It employs the Virtual Mass technique for numerical propagation or integration of the spacecraft state vector. This procedure was developed to achieve high computational speed and precision, and is documented in Reference 1. The subroutines implementing the Virtual Mass technique are presented here. This implementation features user coded Fortran-like trajectory descriptions and output controls. In addition, it contains an integral link to standard JPL planetary ephemeris files.

FILE  
COPY

SUBJECT: The Univac 1108 Virtual Mass  
Trajectory Simulation Program  
Case 103-9

DATE: March 30, 1971

FROM: R. F. Jessup

MEMORANDUM FOR FILE

1. INTRODUCTION

The basic objective of the Virtual Mass Trajectory Simulation Program is to numerically propagate a spacecraft trajectory from an initial to a final state, subject to various constraints. These constraints are given by an events list and a data deck. The data deck, in Fortran NAME LIST form, gives static constraints and program controls. The events list, written as a set of Fortran statements called a procedure, is compiled into Fortran programs which simulate the trajectory under study. In this way the events list may contain dynamic as well as static constraints; in addition, it controls output printing and plotting.

To the analyst familiar with Fortran, this memorandum is intended to be a user's guide. To the programmer experienced in Fortran, it is intended to be a system description as well as a base for further development and wider application. Several examples are used to illustrate data deck and events list preparation.

2. EVENTS LIST PREPARATION

In the Virtual Mass program a trajectory is simulated as one or more subtrajectories or segments: each is characterized by a specific spacecraft configuration and environment, and each has distinct input and output requirements. The program transitions from one segment to another are called "events," and are indicated by calls to a subroutine EVNACC. These calls, along with trajectory printing and plotting descriptions and calculations specifying the configuration and/or environment, are organized into a Fortran procedure called an events list. The events list contains a sequence of calls to EVNACC, each identifying one event and giving the condition under which the event is to take place. Associated with each event there are one-time calculations for beginning the trajectory segment, and auxiliary calculations for each computing step taken until

another event has been reached. The one-time calculations are useful for program initialization or for termination of trajectory segments. AUPLOT subroutine calls (Ref. 3) for collecting data and requesting plots are also placed in the events list. Auxiliary equations can be used for output calculations, for setting aside data for plots, and for calculations controlling subsequent events. Trajectory printing which takes place at each event and at selected computing intervals in between may also be indicated in the events list. In general, any of the program COMMON variables may be printed or plotted as desired. They may not be modified or scaled, however, except as provided in the data deck section described below. When modification is appropriate for output preparation or for special applications, assignments should be made to VAC, a sub-block of the COMMON block VAR. Section 6 illustrates such storage assignments and gives events lists for sample Virtual Mass applications.

### 3. DATA DECK PREPARATION

Standard input to the Virtual Mass Trajectory Simulation program is entered as a Fortran NAME LIST called DATADK. The variables listed below are among those most frequently assigned. Other program locations may be reached as necessary by use of a subscript on an appropriate COMMON block name. For this purpose the user is referred to the elements COMMON and DDREAD in the Appendix. Input data may be scaled or otherwise preprocessed as desired by one-time calculations for the first trajectory event or by an .INITAL subroutine supplied by the user. .INITAL is called after the data deck is read into the computer.

As a rule, VMASS variables beginning with the letters I through N, with the exception of L, have integer values; L is used for logical variables. All other variables are double precision. Arrays for state vector or table use are dimensional as shown in Table 2. The principal DATADK variables are as follows:

RSCDDK, VSCDDK - Initial spacecraft state vector. This state vector must be converted in .INITAL or the Events List to inertial coordinates and scaled as necessary to AU and AU/day for compatibility with the JPL planetary ephemeris. The result must be assigned to RSCEIN and VSCEIN, the simulator locations for latest inertial spacecraft state.

- TEPOJD - Initial Julian date.
- TEPOJS - Incremental time within the day, in seconds.
- LPRNOW - Logical print-now option. Causes the program to perform trajectory printing concurrent with calculations.
- LPRLAT - Logical print later option. Causes the program to write the COMMON block VAR on secondary storage so that trajectory printing may be performed as a separate program execute under control of element PRTRAJ.
- LSTDDK - Logical last data deck indicator. The program will process a sequence of data decks if desired; the one submitted last should contain this indicator. Data entered for one data deck will be restored as input for the next, provided that IEVREP has been assigned an event name.
- IEVREP - Repeat event option. When assigned the name of an event, either in the data deck or in the one-time statements of the event, the program state given in the VAR block will be restored prior to reading of any subsequent data decks.
- IEVSTR - Start event option. Indicates to the program the initial event. If not assigned, the first event encountered in the events list will be regarded as initial.
- IEVSTP - Stop event option. Indicates to the program the final event. In multiple data deck runs IEVSTP must be assigned an event name, either in the data deck or in a one-time statement. In single data deck runs IEVSTP need not be assigned provided the final event contains a CALL EXIT one-time statement.
- RTOPRF - Position vector precision tolerance. Computation precision throughout the Virtual Mass program is keyed to this number. If not specified by the user, it has a standard value of  $10^{-15}$ .

- LPRCOM - Logical print COMMON option. Causes the program to tabulate all of the COMMON blocks as they appear after data deck reading. This tabulation is automatically produced whenever a program error termination is made. It may be requested by the user through a CALL PRTCOM statement.
- JK - Planetary reference list. In normal interfacing with the JPL planetary ephemeris the sun and all the planets except earth are referenced as single points. The earth and its moon are treated as two separate points. The list JK may be reassigned as desired for a modified ephemeris. For such purposes the user should consult the program elements COMMON and LODCOM. Section 6 gives an example with an earth, moon, and sun ephemeris. Whenever the list JK is reassigned, JKN should also be reassigned to show the number of entries.
- IPNOWM - With respect to the print now option, this gives the number of computing intervals between desired printouts.
- IPLATM - With respect to the print later option, this gives the number of computing intervals between copying of the COMMON block VAR onto secondary storage.
- VAR - COMMON block for working variables. Specific assignments should appear with a subscript as appropriate from program element COMMON.
- IOU - COMMON block for input and output unit assignments. See VAR. All 10 references in the program are symbolic (e.g., IOEPH for the ephemeris file) in order to facilitate program expansion and recompilation on other computers.
- FAC - COMMON block for constants. See VAR. These constants are initially assigned by the program element LODCOM. They may be reassigned in the data deck.
- PHC - COMMON block for physical constants. See VAR.
- EPH - COMMON block for ephemeris data. See VAR.
- ISW - COMMON block for integer working variables. See VAR.

LSW - COMMON block for logical working variables. See VAR.

VAC - User's storage sub-block (50 locations) in the block VAR. VAC is intended to facilitate communication between the events list and trajectory print procedures.

#### 4. VIRTUAL MASS INTERNAL PROGRAM FEATURES

As indicated in Section 6, the Virtual Mass program events lists procedures are compiled into various Fortran sub-routines. These subroutines are then combined with other Virtual Mass and Fortran library routines into an absolute element. During operation on the UNIVAC 1108 computer the routine SIMXXX retains overall control, and contains the compiled events list and auxiliary equations. Execution follows the flow diagram of Figure 1 and includes program initialization, trajectory calculations, and output printing and plotting.

Virtual Mass input data is supplied in Fortran NAME LIST form through the subroutine DDREAD. Preloading of the program COMMON blocks takes place from storage by LODCOM or from the intermediate roll back file by ROLBCK. Control of these routines for the various program initiation steps is taken by the subroutine SETUP.

Increments in the spacecraft trajectory are calculated from the input data under control of the subroutine STEPIN. The events list coding defines the specifications in effect over distinct trajectory segments. Data governing the events is placed in the IEV table by EVNACC. Control of the events is taken by the subroutine EVNDT. EVNDT keeps a running account of the difference between current time and the projected time for occurrence of each event. When current and projected time coincide for some event, that event is said to be activated. User-supplied one-time calculations are then made. In addition, the auxiliary and end-of-cycle equations supplied for the event are flagged for calculation in each step of the trajectory segment. Virtual Mass calculations appear in the subroutine STEPIN.

Program output for printing and for intermediate storage, as may be required in targeting and multiple data deck runs, is under control of the subroutine RECORD. For each event, and for selected spacing of intermediate computing increments, RECORD calls for user-coded printing to occur. Under the "print now" option, printing is concurrent with computing; under

"print later," the common block KVAR is copied onto the intermediate file IOPRL. Printing then takes place under control of PRTRAJ in a separate approach to the computer. For each event encountered, RECORD also calls for the copying of KVAR onto intermediate file IOROL in case of need to make program recovery to the event.

Output for graphs may be kept under user control under provisions of the graphical data facility, AUPLOT (Ref. 3). Typical applications are illustrated in Section 6, with examples of data collection and plot request calls included in the events lists. AUPLOT prepares its own intermediate file IOPLT which is processed into graphs in a separate computer operation.

## 5. ADDITIONAL FEATURES

The Virtual Mass Program is designed to operate in a double precision mode on the Univac 1108 computer. Computational precision is keyed to the single variable RTOPRF which is normally set to  $10^{-15}$ . RTOPRF may be set to a larger value, such as  $10^{-10}$ , if faster-running approximations are desired.

The program normally operates in the 1108 batch mode. However, small problems, such as those illustrated in Section 6, are easily run from a time-sharing terminal. In either case, several data decks may be submitted as successive inputs to an events list simulation. For such successive inputs the program will automatically restart at the start-up event or at any event in which the one-time statement IEVREP = IEVCUR appears. On restart indication by IEVREP, the program will be restored to its earlier state.

In order to facilitate continued development and possible conversion to other computers, use of the Fortran V procedure feature is limited to a form of program editing. In addition, a program listing is provided in the Appendix, and Tables 1 through 5 show COMMON symbols usage, subroutine references, and references to Fortran V features.

## 6. TYPICAL APPLICATIONS

Two typical applications of the Virtual Mass program are illustrated in this section, including input decks and program printout.

### 6.1 An Earth to Mars Trajectory

This example illustrates the use of the JPL planetary ephemeris files with the Virtual Mass program to calculate an Earth to Mars trajectory, beginning in December 1964. The Virtual Mass program contains linkage to the JPL planetary ephemeris files as an integral feature. Interface with the files is through a JPL supplied subroutine READE and a pair of Julian date variables: TEPEJD and TEPEJS for day, and fraction thereof in seconds, respectively. For compatibility purposes the Virtual Mass program assumes internal units of AU (astronomical units) and AU per day. Inputs for this example, which were taken from the JPL double precision trajectory program DPTRAJ (Ref. 4), are in kilometers and kilometers per second. Note that an appropriate conversion is made in the one-time calculations for event 1. For convenience in comparison with DPTRAJ, the output is converted back to kilometers and kilometers per second.

This example is described through the following graphs and printouts:

<u>Figure</u>	<u>Description</u>
1	Univac 1108 EXEC VIII input deck for the basic example, RUN100
2	Program printout from RUN100
3	Input deck for the extended example, RUN110, containing graphical output statements from AUPLOT.
4	Program printout from RUN110
5-10	Output graphs from RUN110

Subroutine EVNACC, which is called from the procedure QEVENT, maintains a table of trajectory events, along with the condition which indicates to the simulator that the event has been reached. For event startup this condition is TTRE = FO. Midcourse and cutoff events are indicated by A1 = C1 and A2 = C2, respectively. The first three arguments of EVNACC identify, respectively:

1. the name of an event to be placed in the table,

2. the name of any other event which must be reached before this event is placed, and
3. the name of any event whose being reached is to force removal of this event.

When an event is reached, the one-time statements following the call are executed. The final argument is used to skip over the one-time statements as appropriate.

The variables A1 and A2 are used by the simulator as auxiliary calculations to control indication of events. They are supplied through the QAUXEQ procedure as follows. The auxiliary equations are evaluated for each trajectory computing interval, and applicable entries in the events table are updated. As an event is approached, i.e., when the difference between A1 and C1 or A2 and C2, etc. becomes small, the simulator adjusts the computing interval for optimum agreement. Indication that the event has been reached is then made, and the program control will make appropriate table updates and one-time calculations on its passage through the EVNACC calls.

Trajectory printing as shown in the QPRINT procedure includes the spacecraft state vector relative to Earth, Mars, and the Sun. The events list is given twice, once with printed output and again with graphics. Of particular interest is the series of 3-dimensional views from above the ecliptic plane. The motion of the Virtual Mass is shown along a trajectory passing near the dominant gravitating bodies. Note that plot data collection is made in the procedure QEOCEQ. QEOCEQ is called for each computing step except those taken on a preliminary basis during iteration to an event.

## 6.2 A Restricted Three Body Trajectory

This example simulates a spacecraft in free flight about two gravitating bodies which are in circular orbit about each other (the circular restricted three body problem). In this case, a special ephemeris routine (EPH2CO) was used without reference to the JPL files. The ability of the program to minimize variations in the value of the Jacobi energy is illustrated in Figure 19. The description of this example is given in the following graphs and printouts:

<u>Figure</u>	<u>Description</u>
11	Univac 1108 EXEC VIII input deck for the basic example, RUN300.
12	Program printout from RUN300.
13	Input deck for the extended example, RUN310, containing graphical output statements from AUPLOT.
14	Program printout from the extended example, RUN310.
15-18	Three-dimensional trajectory plots for reference precision values RTOPRF of $10^{-19}$ , $10^{-15}$ , $10^{-10}$ , and $10^{-5}$ , respectively.
19	Plot of Jacobi energy variation from starting value versus trajectory time for the four values of RTOPRF.
20	Plot of computing steps versus trajectory time for the four values of RTOPRF.
21	Plot of final position variation from JPL results versus the four values of RTOPRF.

## 7. SUMMARY

The Virtual Mass technique has been implemented in a computer program for the Univac 1108 computer. Program capability includes open loop trajectory simulation with linkage to the JPL ephemeris files. Considerations for further development include incorporation of the capability to simulate thrust, drag and solar radiation pressure; expansion to an n-body integrator; targeting routines, and provision of an events list translator. The translator would enable styling the events list in the form of a general purpose simulation language.

This program was developed in collaboration with D. H. Novak, originator of the Virtual Mass technique.

*R F Jessup*  
R. F. Jessup

2011-RFJ-vh

Attachments

# BELLCOMM. INC.

## REFERENCES

1. Novak, D. H., "Virtual Mass Technique for Computing N-Body Solutions," Technical Memorandum No. 70-2011-3, Bellcomm, Inc., Washington, D. C., December 28, 1970.
2. Novak, D. H., "Computationally Convenient Forms for Conic Section Equations," Memorandum for File B71 01045, Bellcomm, Inc., Washington, D. C., January 8, 1971.
3. Jessup, R. F., "AUPLOT II - A System of Data Handling and Plot Instruction Subroutines for Computer Graphics," Technical Memorandum No. 70-2011-2, Bellcomm, Inc., Washington, D. C., November 20, 1970.
4. Khatib, A. R., "Double-Precision Trajectory Program: DPTRAJ," Space Program Summary 37-49, Vol. II: The Deep Space Network, Jet Propulsion Laboratory, Pasadena, California, January 31, 1968, pp. 52-55.

Figure 1

## Earth to Mars Trajectory Instructions

```

@ ELY,DLI RUN100
ELY PROCESSOR LEVEL      3
000001  000  @ HDG  VMASS EARTH TO MARS / K JESSUP
000002  000  @ USE  LAUPLT,AUPLT*RFJI
000003  000  @ PDP,FIL PRB100
000004  000  @EVENT PROC
000005  000  C REFERENCE - DOUBLE PRECISION TRAJECTORY PROGRAM DPTRAJ, JPL, JULY 69
000006  000          CALL EVNACC (' 0', F0, ' 0', TEPEJ, F0, $210)
000007  000  C          CONVERT EARTH CENTERED INPUT (RSCDDK, VSCDDK) TO INERTIAL
000008  000          CALL EPHEM
000009  000          RUNIT = AUTOKM
000010  000          VUNIT = AUTOKM / F86400
000011  000          DO 120 I = 1, 3
000012  000              RSCEIN(I) = ROOEIN(KEARTH-1+I) + RSCDDK(I) / RUNIT
000013  000          120  VSCEIN(I) = VOOEIN(KEARTH-1+I) + VSCDDK(I) / VUNIT
000014  000  C          INITIALIZE SIMULATOR
000015  000          LPRNOW = .TRUE.
000016  000          CALL VMXACT
000017  000          IEVREP = IEVCUR
000018  000          GO TO 9915
000019  000          210  CALL EVNACC (' 1', ' 0', ' 1', A1, C1, $220)
000020  000          GO TO 9915
000021  000          220  CALL EVNACC (' 2', ' 1', ' 2', A2, C2, $230)
000022  000          GO TO 9915
000023  000          230  CALL EVNACC (' 3', ' 2', ' 3', A3, C3, $240)
000024  000          DIMENSION RSCX04(3) / .2506011977060753D5, -.1119064506631482D6,
000025  000          1  -.5673159182445349D5 /
000026  000          DIMENSION VSCX04(3) / -.4016695419022551D1, -.1144767285967780D1,
000027  000          1  .3056484365226338D0 /
000028  000          DO 232 I = 1, 3
000029  000              TV1(I) = RSCX04(I) - RUNIT * RSCE00(KMARS-1+I)
000030  000          232  TV2(I) = VSCX04(I) - VUNIT * VSCE00(KMARS-1+I)
000031  000              T1 = VALUE (TV1)
000032  000              T2 = VALUE (TV2)
000033  000          WRITE (IOPRT, 234) TV1, T1, TV2, T2
000034  000          234  FORMAT (// ' FINAL STATE VECTOR DISPLACEMENT FROM DPTRAJ' /
000035  000          1  ' POSITION', 4G30.1B / ' VELOCITY', 4G30.1B)
000036  000          LPRNOW = .FALSE.
000037  000          IEVSTP = IEVCUR
000038  000          GO TO 9915
000039  000          240  CONTINUE
000040  000          END
000041  000  @AUXEQ PROC
000042  000          TEPEJ = TEPEJD + TEPEJS / F86400
000043  000          IF (IEVCUR .NE. ' 0') GO TO 510
000044  000          A1 = VALUE (RVMEIN)
000045  000          RETURN
000046  000          510  IF (IEVCUR .NE. ' 1') GO TO 520
000047  000          A2 = VALUE (RSCE00(KMARS))
000048  000          RETURN
000049  000          520  CONTINUE
000050  000          END
000051  000  @STORE PROC
000052  000          EQUIVALENCE (VAC(1),TEPEJ,A3), (VAC(2), A1 ),
000053  000          1 (VAC(3), A2 ), (VAC(4), RUNIT ), (VAC(5), VUNIT )
000054  000          DATA C1 / .100 / C2 / .100 / C3 / .2438956630258042D7 /
000055  000          END

```

```

000056 000 QPRINT PROC
000057 000 CALL SCALE (RUNIT, RSCE00(KEARTH), TV1)
000058 000 CALL SCALE (VUNIT, VSCE00(KEARTH), TV2)
000059 000 CALL SCALE (RUNIT, RSCE00(KSUN ), TV3)
000060 000 CALL SCALE (VUNIT, VSCE00(KSUN ), TV4)
000061 000 CALL SCALE (RUNIT, RSCE00(KMARS ), TV5)
000062 000 CALL SCALE (VUNIT, VSCE00(KMARS ), TV6)
000063 000 WRITE (IOPRT, 710) (TV1(I), I = 1, 18)
000064 000 710 FORMAT ( // ' RSCE03', 3G30.18/ ' VSCE03', 3G30.18/
000065 000 1 ' RSCE10', 3G30.18/ ' VSCE10', 3G30.18/
000066 000 2 ' RSCE04', 3G30.18/ ' VSCE04', 3G30.18)
000067 000 PRINT 720, TEPEJ, INCNIR
000068 000 720 FORMAT (' TEPEJ ', 330.18, 18X, 'INCNTR', I6)
000069 000 END
000070 000 GEOCEQ PROC
000071 000 END
000072 000 @ COPY,S LVMASS.COMMON,COMMON
000073 000 @ COPY,S LVMASS.SIMXXX,SIMXXX
000074 000 @ COPY,S LVMASS.PRTXXX,PRTXXX
000075 000 @ FOR,w SIMXXX,SIMXXX
000076 000 @ FOR,w PRTXXX,PRTXXX
000077 000 -22,23
000078 000 INCLUDE QFAC
000079 000 INCLUDE QTEM
000080 000 INCLUDE QEPH
000081 000 @ MAP LVMASS.MAP,MAP100
000082 000 @ ASG,T 12,T,1277
000083 000 @ XQT MAP100
000084 000 $DATADK
000085 000 RSCDDK = -.15521917D7, .11758164D7, .54651270D6,
000086 000 VSCDDK = -.24260004D1, .13151313D1, .86732622D0,
000087 000 TEPOJD = .2438735D7, TEPOJS = 14965.D0,
000088 000 RTOPRF = 1.0-19,
000089 000 $END
000090 000 $DATADK
000091 000 RTOPRF = 1.0-18,
000092 000 $END
000093 000 $DATADK
000094 000 LSTDDK = .TRUE.,
000095 000 RTOPRF = 1.0-17,
000096 000 $END

```

Figure 2

Earth to Mars Trajectory Output

VIRTUAL MASS TRAJECTORY 08 JUN 70 15:36:55

0 AT 0 DAYS 0 HR 0 MIN 0 SEC; JULIAN 2438735.0 DAYS .149650+005 SEC (1964 DEC 5 16: 9:25); COMPUTED 08 JUN 70 15:38:29

RSCE03	- .15521916999999998+007	.11750163999999998+007	.54651269999999935+006
VSCE03	- .24260003999999997+001	.18151312999999999+001	.86732621999999994+000
RSCE10	.40621692264262558+008	.130703700775912621+009	.567375381543019336+008
VSCE10	- .314683763039406437+002	.953919393201119771+001	.421809095464935946+001
RSCE04	.189795871644716737+009	- .457915839192475830+008	- .283052261211136422+008
VSCE04	- .131440551501514505+002	.21200841951255072+002	.908028458992784089+001
TEPEJ	.243673517320601851+007	INCNTR 0	

EVNDR \*\*\* DIVERGENCE  
1 1 603 1161 14 0 .344258176815814550-003 -.107336015076064939-016

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+005 SEC (1964 DEC 14 15:48:34); COMPUTED 08 JUN 70 15:38:50

RSCE03	- .339041922743966861+007	.260364788868617698+007	.122749937548072506+007
VSCE03	- .231024373348006574+001	.187957770021362892+001	.894328930443651885+000
RSCE10	.158090592682783597+008	.13650500635702950+009	.593048292702085816+008
VSCE10	- .323262788911615265+002	.53389080248532350+001	.239373857162319780+001
RSCE04	.178798719769476169+009	- .30490078488106155+008	- .217396542812426426+008
VSCE04	- .150738611420174361+002	.181852023437669104+002	.782629215593127879+001
TEPEJ	.243674415873242562+007	INCNTR 163	

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+005 SEC (1964 DEC 14 15:48:34); COMPUTED 08 JUN 70 15:38:50

RSCE03	- .339041922743966861+007	.260364788868617698+007	.122749937548072506+007
VSCE03	- .231024373348006574+001	.187957770021362892+001	.894328930443651885+000
RSCE10	.158090592682783597+008	.13650500635702950+009	.593048292702085816+008
VSCE10	- .323262788911615265+002	.53389080248532350+001	.239373857162319780+001
RSCE04	.178798719769476169+009	- .30490078488106155+008	- .217396542812426426+008
VSCE04	- .150738611420174361+002	.181852023437669104+002	.782629215593127879+001
TEPEJ	.243674415873242562+007	INCNTR 163	

RSCE03	- .144432498307061050+008	.173300617586243623+008	.790043911568959934+007
VSCE03	- .358104858109378964+001	.509801317388704618+001	.225091375809769230+001
RSCE10	- .123539669721137946+009	.108477874163413242+009	.474270462874571451+008
VSCE10	- .241212414533276660+002	- .152131802153376107+002	- .65577641883555163+001
RSCE04	.102440721957332507+009	.154000657887324199+008	- .136530801515766954+007
VSCE04	- .148698219333432744+002	.299207102396274033+001	.155030323455757030+001
TEPEJ	.243679837163260961+007	INCNTR 500	

EVNDR \*\*\* DIVERGENCE  
2 2 605 1163 13 0 .975393754408302284-003 -.542101086242752217-018

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279279+005 SEC (1965 JUN 6 19:45:27); COMPUTED 08 JUN 70 15:39:42

RSCE03	-.157244478780364407+009	.318117744753778155+008	.141331581980307050+008
VSCE03	-.202507691780482155+002	-.119203928541169008+002	-.520609033907776245+001
RSCE10	-.194647030055254268+009	-.103142414483644423+009	-.443896091916908894+008
VSCE10	.812608544971407444+001	-.187742890018778291+002	-.817823592962931182+001
RSCE04	.143437736750915125+008	.418324760621460142+007	-.742894422677959277+006
VSCE04	-.455269829227484057+001	-.14955552389570971+001	.934105943661758311-001
TEPEJ	.243691832324002325+007	INCNTR 753	

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279279+005 SEC (1965 JUN 6 19:45:27); COMPUTED 08 JUN 70 15:39:42

RSCE03	-.157244478780364407+009	.318117744753778155+008	.141331581980307050+008
VSCE03	-.202507691780482155+002	-.119203928541169008+002	-.520609033907776245+001
RSCE10	-.194647030055254268+009	-.103142414483644423+009	-.443896091916908894+008
VSCE10	.812608544971407444+001	-.187742890018778291+002	-.817823592962931182+001
RSCE04	.143437736750915125+008	.418324760621460142+007	-.742894422677959277+006
VSCE04	-.455269829227484057+001	-.14955552389570971+001	.934105943661758311-001
TEPEJ	.243691832324002325+007	INCNTR 753	

RSCE03	-.211306631843935291+009	-.23399260251481596+008	-.993620817322751492+007
VSCE03	-.132683664461072241+002	-.235027562459873856+002	-.102374936386666371+002
RSCE10	-.159608595661414963+009	-.154666882658850964+009	-.668621395333525690+008
VSCE10	.142421132208548874+002	-.143453382595632411+002	-.626539867217079430+001
RSCE04	.958026638100172615+006	.148801450618769870+006	-.116511501896118243+006
VSCE04	-.423856333025538467+001	-.119418677206020171+001	.267419073743687283+000
TEPEJ	.243695409509752317+007	INCNTR 1000	

3 AT 221 DAYS 10 HR 58 MIN 9 SEC; JULIAN 2438956.0 DAYS .544543+005 SEC (1965 JUL 15 3: 7:34); COMPUTED 08 JUN 70 15:40:25

RSCE03	-.214132798350909612+009	-.286193758550610588+008	-.122114087458806924+008
VSCE03	-.125523352535545032+002	-.241643678275188782+002	-.105078843773603645+002
RSCE10	-.156652769211011667+009	-.157708886279141807+009	-.682163329505802188+008
VSCE10	.145600649481108093+002	-.139355786747700086+002	-.607159805411802536+001
RSCE04	.256607233956298866+005	-.111906539094329124+006	-.567317156276665034+005
VSCE04	-.431669466926663034+001	-.114476771707534006+001	.305648064012906282+000
TEPEJ	.243695665025804199+007	INCNTR 1413	

FINAL STATE VECTOR DISPLACEMENT FROM DPTRAJ

POSITION	-.603625022356692607+000	.884311809245446057-001	.123803213013502500+000	.622503395113027136+000
VELOCITY	-.749755720663999536-006	.431107560058471839-006	.572509727517792572-006	.941674501135263381-006

VIRTUAL MASS TRAJECTORY 08 JUN 70 15:40:26

0 AT 0 DAYS 0 HR 0 MIN 0 SEC; JULIAN 2438735.0 DAYS .149650+005 SEC (1964 DEC 5 16: 9:25); COMPUTED 08 JUN 70 15:40:33

RSC003	- .15521916999999999999+007	.11750163999999999999+007	.5465126999999999999935+006
VSCE03	- .242600039999999999997+001	.18151312999999999999+001	.867326219999999999994+000
RSCE10	.406216922642625580+008	.130703700775912621+009	.567375381543019336+008
VSCE10	- .314683703039406437+002	.903919393201119771+001	.421809095464935946+001
RSCE04	.189795871644716737+009	- .457915839192475830+008	- .283052261211136422+008
VSCE04	- .131440551501514505+002	.212006419512550728+002	.908028458992784089+001
TEPEJ	.243673517320601851+007	INCNTR 0	

EVINDT \*\*\* DIVERGENCE  
 1 1 603 1161 14 0 .826917422237503080-003 -.348028897367846923-016

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+005 SEC (1964 DEC 14 15:48:34); COMPUTED 08 JUN 70 15:40:37

RSC003	- .339041922600075380+007	.200394788604821297+007	.122749937520827759+007
VSCE03	- .231024373417344770+001	.187907769907291665+001	.894328929948513133+000
RSCE10	.196090592659951050+008	.13650500635263443+009	.593048292700221587+008
VSCE10	- .320262788918988024+002	.533809080102586728+001	.239373857098966016+001
RSCE04	.176798719768182000+009	- .304900078485132024+008	- .217396542811173873+008
VSCE04	- .190738611428370849+002	.181802023423918046+002	.782629215533866969+001
TEPEJ	.243674415673242029+007	INCNTR 115	

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+005 SEC (1964 DEC 14 15:48:34); COMPUTED 08 JUN 70 15:40:37

RSC003	- .339041922600075380+007	.200394788604821297+007	.122749937520827759+007
VSCE03	- .231024373417344770+001	.187907769907291665+001	.894328929948513133+000
RSCE10	.196090592659951050+008	.13650500635263443+009	.593048292700221587+008
VSCE10	- .320262788918988024+002	.533809080102586728+001	.239373857098966016+001
RSCE04	.176798719768182000+009	- .304900078485132024+008	- .217396542811173873+008
VSCE04	- .190738611428370849+002	.181802023423918046+002	.782629215533866969+001
TEPEJ	.243674415673242029+007	INCNTR 115	

RSC003	- .134677783244204517+009	.424812841601604145+008	.187966166934305132+008
VSCE03	- .206033142668701930+002	- .771306512790405946+001	- .337692229881432601+001
RSCE10	- .202305238088469065+009	- .820272284745426211+008	- .351954341214974320+008
VSCE10	.303704449068721740+001	- .199799828050093284+002	- .869645761581737779+001
RSCE04	.194704679448189350+008	.59113529868727054+007	- .785708375735651996+006
VSCE04	- .489048485602048705+001	- .108002237544438310+001	- .168690369263072990-001
TEPEJ	.243690572261211600+007	INCNTR 50	

EVINDT \*\*\* DIVERGENCE  
 2 2 805 1163 14 0 .534923430377383062-003 -.151788304147970621-017

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279279+005 SEC (1965 JUN 6 19:45:27); COMPUTED 08 JUN 70 15:40:53

RSCE03	-.157244478780736181+009	.318117744566055486+008	.141331581898471691+008
VSCE03	-.202507691771570985+002	-.119203928550967255+002	-.520609033950608854+001
RSCE10	-.194847030047291975+009	-.103142414504429627+009	-.443896092007473219+008
VSCE10	.812608545101397631+001	-.187742890013534400+002	-.817823592940431944+001
RSCE04	.143437736793301434+008	.418324759050403006+007	-.742894429305072021+006
VSCE04	-.455269829157119226+001	-.14955552367720080+001	.934105944667749943-001
TEPEJ	.243691832324002665+007	INCNTR 520	

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279279+005 SEC (1965 JUN 6 19:45:27); COMPUTED 08 JUN 70 15:40:53

RSCE03	-.157244478780736181+009	.318117744566055486+008	.141331581898471691+008
VSCE03	-.202507691771570985+002	-.119203928550967255+002	-.520609033950608854+001
RSCE10	-.194847030047291975+009	-.103142414504429627+009	-.443896092007473219+008
VSCE10	.812608545101397631+001	-.187742890013534400+002	-.817823592940431944+001
RSCE04	.143437736793301434+008	.418324759050403006+007	-.742894429305072021+006
VSCE04	-.455269829157119226+001	-.14955552367720080+001	.934105944667749943-001
TEPEJ	.243691832324002665+007	INCNTR 520	

3 AT 221 DAYS 10 HR 58 MIN 9 SEC; JULIAN 2438956.0 DAYS .544543+005 SEC (1965 JUL 15 3: 7:34); COMPUTED 08 JUN 70 15:41:05

RSCE03	-.214132798335868781+009	-.286193758608018822+008	-.122114087480940274+008
VSCE03	-.12552332080620359+002	-.241643677882598541+002	-.105078843621804213+002
RSCE10	-.156652769195972928+009	-.157708886284283421+009	-.682163329527938964+008
VSCE10	.145300649934094429+002	-.139355786355113594+002	-.607159804893824463+001
RSCE04	.256607384358249699+005	-.111906544236930735+006	-.567317178418367361+005
VSCE04	-.431669462397406889+001	-.114476767781656226+001	.305648079192742610+000
TEPEJ	.243895663025804199+007	INCNTR 966	

FINAL STATE VECTOR DISPLACEMENT FROM DPTRAJ

POSITION	-.618665217439911430+000	.935737825346905085-001	.126017383246221470+000	.638265685218974857+000
VELOCITY	-.795046482098159592-006	.391848782267320561-006	.357329891189760857-006	.955684157129674259-006



VSC010	.81208546010923055+001	-.167742889970421346+002	-.817823592781141615+001
RSCE04	.143437737006951047+008	.418324751143471175+007	-.742894462030051163+006
VSC004	-.455269823721685579+001	-.14955552239800396+001	.934105950706054625-001
TEPEJ	.243691832324005367+007	INCNTR 359	

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279279+005 SEC (1965 JUN 6 19:45:27); COMPUTED 08 JUN 70 15:41:28

RSCE03	-.137244470796029287+009	.318147743531921416+006	.141331581447467910+008
VSC003	-.20207091713121085+002	-.119203028630456950+002	-.520609034310783734+001
RSCE10	-.194047029996319427+009	-.103142414623648250+009	-.443896092527882568+008
VSC010	.81208546010923055+001	-.167742889970421346+002	-.817823592781141615+001
RSCE04	.143437737006951047+008	.418324751143471175+007	-.742894462030051163+006
VSC004	-.455269823721685579+001	-.14955552239800396+001	.934105950706054625-001
TEPEJ	.243691832324005367+007	INCNTR 359	

RSCE03	-.212283039626351225+009	-.251474357426816121+008	-.106999689306137767+009
VSC003	-.130214601712035374+002	-.237402187095143081+002	-.103421940961186272+002
RSCE10	-.194047029996319427+009	-.195707145121741524+009	-.673252046748644564+008
VSC010	.143623905604722870+002	-.14220960552+749020+002	-.621200401959359494+001
RSCE04	.043280903429541219+006	.60179435301+502890+005	-.965230336932313190+005
VSC004	-.424014356496021535+001	-.11939525576912951+001	.268567669557245319+000
TEPEJ	.243691832324005367+007	INCNTR 500	

3 AT 221 DAYS 10 HR 53 MIN 9 SEC; JULIAN 2438956.0 DAYS .544543+005 SEC (1965 JUL 15 3: 7:34); COMPUTED 08 JUN 70 15:41:37

RSCE03	-.214132793251912557+009	-.236193758025620669+008	-.122114087573768955+008
VSC003	-.12552332959614607+002	-.241603675682281622+002	-.105078842759119999+002
RSCE10	-.194047029996319427+009	-.19770886300042340+009	-.682163329620762179+008
VSC010	.143623905604722870+002	-.13930578415+790094+002	-.607159796266956413+001
RSCE04	.206608220930837005+005	-.111906565994279399+006	-.567317271233726893+005
VSC004	-.431069437552710940+001	-.114476745770447759+001	.305648165461334444+000
TEPEJ	.243691832324005367+007	INCNTR 667	

FINAL STATE VECTOR DISPLACEMENT FROM OPTRAJ			
POSITION	-.702622476256570921+000	.119331131199013726+000	.135298919199442480+000
VELOCITY	-.104049544159665553-000	.174816697081796391-006	.271061299354953544-006
			.724765763196019207+000
			.109173162551067398-005

Figure 3

## Earth to Mars Trajectory with Graphics

```

VMASR*RFJK.RUN110
1  @ HDG  VIRTUAL MASS 110 EARTH TO MARS WITH GRAPHICS / R JESSUP
2  @ PDP,FLI QEVENT/RUN110
3  QEVENT PROC
4  C      EARTH TO MARS FLIGHT WITH EPHEMERIS FILE
5          CALL EVNACC (' 0', FU, ' 0', TEPEJ, FU, $210)
6  C      CONVERT EARTH CENTERED INPUT (RSCDDK, VSCDDK) TO INERTIAL
7          CALL EPHEM
8          IF (.NOT. LOGOUT) GO TO 110
9          WRITE (IOMES, 100)
10         CALL PRTVAR
11         CALL EXIT
12 100     FORMAT (' EPHEM CALLED BY SIMXXX')
13 110     RUNIT = AUTOKM
14         VUNIT = AUTOKM / F86400
15         DO 120 I = 1, 3
16             RSCEIN(I) = RDOEIN(KEARTH-1+I) + RSCDDK(I) / RUNIT
17 120     VSCEIN(I) = VDOEIN(KEARTH-1+I) + VSCDDK(I) / VUNIT
18         CALL PLTITL ('SPACE CRAFT AND VIRTUAL MASS DEPARTING EARTH')
19         REAL EYE(3), ORIGIN(3)
20         DATA EYE / .0, -31.9, 91.6 / ORIGIN / 3*8. /
21         CALL PRJEYE (EYE, 1)
22         XL = -2.300
23         XU = .700
24         YL = -1.200
25         YU = 1.800
26  C      INITIALIZE SIMULATOR
27         CALL VMXACT
28         LPRNOW = .TRUE.
29         GO TO 9915
30 210    CALL EVNACC (' 1', ' 0', ' 1', A1, C1, $220)
31         CALL PLOT
32         GO TO 9915
33 220    CALL EVNACC (' 2', ' 1', ' 2', A2, C2, $230)
34         CALL PLTITL ('TRANS MARS MOVEMENT')
35         CALL PLOT
36         GO TO 9915
37 230    CALL EVNACC (' 3', ' 2', ' 3', A3, C3, $240)
38         CALL PLTITL ('SPACE CRAFT AND VIRTUAL MASS APPROACHING MARS')
39         CALL PLOT
40         CALL PLTEND
41         CALL EXIT
42 240    CONTINUE
43     END
44  @ PDP,FLI QAUCEQ/RUN110
45  QAUCEQ PROC
46         TEPEJ = TEPEJD + TEPEJS / F86400
47         IF (IEVCUR .NE. ' 0') GO TO 510
48         A1 = VALUE (RVMEIN)
49         RETURN
50 510    IF (IEVCUR .NE. ' 1') GO TO 520
51         A2 = VALUE (RSCE00(KMARS))
52         RETURN
53 520    CONTINUE
54     END
55  @ PDP,FLI QEOCEQ/RUN110
56  QEOCEQ PROC

```

```

57      DO 530 I = 0, 2
58      CALL COLECT (' X' + I, SNGL (RSCEIN(I+1)))
59      CALL COLECT ('RVMEIX' + I, SNGL (RVMEIN(I+1)))
60      CALL COLECT (KEARTH + I, SNGL (ROOEIN(KEARTH+1)))
61      CALL COLECT (KMARS + I, SNGL (ROOEIN(KMARS +1)))
62      530 CALL COLECT (KSUN + I, SNGL (ROOEIN(KSUN +1)))
63      IF (MOD (INCNTR, IPLOTM) .NE. 0) RETURN
64      IF (INCNTR .NE. 0) GO TO 540
65      CALL PLTOUT
66      CALL PLTICS (' X', SNGL (XL), SNGL (XU))
67      CALL PLTICS (' Y', SNGL (YL), SNGL (YU))
68      540 CALL PLOT
69      END
70      @ PDP,FLI QSTORE/RUN110
71      QSTORE PROC
72      EQUIVALENCE (VAC(1),TEPEJ,A3), (VAC(2), A1 ),
73      2 (VAC(3), A2 ), (VAC(4), RUNIT ), (VAC(5), VUNIT ),
74      3 (VAC(7), XL ),
75      4 (VAC(8), XU ), (VAC(9), YL ), (VAC(10), YU )
76      C NOTE. THE DATA STATEMENT MAY NOT BE USED FOR ASSIGNMENTS TO COMMON
77      C SEE LODCOM WHICH CLEARS COMMON TO 0 ON PROGRAM STARTUP
78      DATA C1 / .1000 / C2 / .100 / C3 / .243895663025804207 /
79      IXP (X) = IFIX (1024. * SNGL ((X - XL) / (XU - XL)))
80      IYP (Y) = IFIX (1024. * SNGL ((Y - YL) / (YU - YL)))
81      END
82      @ PDP,FLI QPRINT/RUN110
83      QPRINT PROC
84      CALL SCALE (RUNIT, RSCE00(KEARTH), TV1)
85      CALL SCALE (VUNIT, VSCE00(KEARTH), TV2)
86      CALL SCALE (RUNIT, RSCE00(KSUN ), TV3)
87      CALL SCALE (VUNIT, VSCE00(KSUN ), TV4)
88      CALL SCALE (RUNIT, RSCE00(KMARS ), TV5)
89      CALL SCALE (VUNIT, VSCE00(KMARS ), TV6)
90      WRITE (IOPRT, 710) (TV1(I), I = 1, 18)
91      710 FORMAT ( // ' RSCE03', 3G30.18/ ' VSCE03', 3G30.18/
92      1 ' RSCE04', 3G30.18/ ' VSCE04', 3G30.18/
93      2 ' RSCE05', 3G30.18/ ' VSCE05', 3G30.18)
94      PRINT 720, TEPEJ, INCNTR
95      720 FORMAT (' TEPEJ ', G30.18, 18X, ' INCNTR', I6)
96      RETURN
97      ENTRY PLOT
98      750 CALL PLTSIM (1)
99      CALL PLCHAR ('S')
100     CALL Q3DPER (' X', ' Y', ' Z')
101     CALL PLCHAR ('V')
102     CALL Q3DPER ('RVMEIX', 'RVMEIY', 'RVMEIZ')
103     CALL PLCHAR (0)
104     CALL Q3DPER (KEARTH, KEARTH+1, KEARTH+2)
105     CALL Q3DPER (KMARS , KMARS +1, KMARS +2)
106     CALL Q3DPER (KSUN , KSUN +1, KSUN +2)
107     CALL AURITE (IXP (ROOEIN(KEARTH)+.1), IYP (ROOEIN(KEARTH+1)),
108     1 1023, 90, 05, 'EARTH')
109     CALL AURITE (IXP (ROOEIN(KMARS )=.5), IYP (ROOEIN(KMARS +1)),
110     1 1023, 90, 04, 'MARS')
111     CALL AURITE (IXP (ROOEIN(KSUN )+.1), IYP (ROOEIN(KSUN +1)),
112     1 1023, 90, 03, 'SUN')
113     END

```

```
114 @ COPY,S LVMASS.SIMXXX,SIMXXX
115 @ COPY,S LVMASS.PRTXXX,PRTXXX
116 @ COPY,S LVMASS.COMMON,COMMON
117 @ FOR,W SIMXXX,SIMXXX
118 @ FOR,W PRTXXX,PRTXXX
119 -23,23
120 INCLUDE @FAC
121 INCLUDE @TEM
122 INCLUDE @EPH
123 @ MAP LVMASS.MAP,MAP110
124 @ XQT MAP110
125 $DATADK
126 RSCDDK = -.15521917D7, .11758164D7, .54651270D6,
127 VSCDDK = -.24260004D1, .18151313D1, .86732622D0,
128 TEPOJD = .2438735D7, TEPOJS = 14965.D0,
129 LSTDDK = .TRUE.,
130 IPLOTM = 16.
131 $END
132 @ FREE 12.
133 @ ASG, TM PLOTFILE.T,PLOT
134 @ XQT LAUPLT.PHASE2
```

Figure 4

Print-out from Earth Mars Trajectory with Graphics

VIRTUAL MASS TRAJECTORY 25 SEP 70 18:08:48

0 AT 0 DAYS 0 HR 0 MIN 0 SEC; JULIAN 2438735.0 DAYS .149650+005 SEC (1964 DEC 5 16: 9:25); COMPUTED 25 SEP 70 18:14:58

RSCE03	-.155219169999999998+007	.117581639999999988+007	.546512699999999935+006
VSCE03	-.242600039999999997+001	.181513129999999999+001	.867326219999999994+000
RSCE10	.406216922642625586+008	.130753700775912621+009	.567375381543019336+008
VSCE10	-.314683764039406437+002	.953919393201119771+001	.421809095464935946+001
RSCE04	.189795871644716737+009	-.457915839192475830+008	-.283052261211136422+008
VSCE04	-.131440551501514505+002	.212038419512550728+002	.908028458992784089+001
TEPEJ	.243873517320601851+007	INCNTR	0

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+006 SEC (1964 DEC 14 15:48:34); COMPUTED 25 SEP 70 18:15:04

RSCE03	-.339041929273824328+007	.260364781445131577+007	.122749934381066057+007
VSCE03	-.231024381159084105+001	.187957756738488906+001	.894328872781588364+000
RSCE10	.158090589888813432+008	.136530500586141258+009	.593048292492335094+008
VSCE10	-.323262789747368843+002	.533869063002880436+001	.239373849675471889+001
RSCE04	.178798719613137236+009	-.304960078067549465+008	-.217396542634683846+008
VSCE04	-.150738612358263212+002	.181832021817972444+002	.782629208615123540+001
TEPEJ	.243874415873250818+007	INCNTR	39

1 AT 8 DAYS 23 HR 39 MIN 9 SEC; JULIAN 2438744.0 DAYS .137145+006 SEC (1964 DEC 14 15:48:34); COMPUTED 25 SEP 70 18:15:06

RSCE03	-.339041929273824328+007	.260364781445131577+007	.122749934381066057+007
VSCE03	-.231024381159084105+001	.187957756738488906+001	.894328872781588364+000
RSCE10	.158090589888813432+008	.136530500586141258+009	.593048292492335094+008
VSCE10	-.323262789747368843+002	.533869063002880436+001	.239373849675471889+001
RSCE04	.178798719613137236+009	-.304960078067549465+008	-.217396542634683846+008
VSCE04	-.150738612358263212+002	.181832021817972444+002	.782629208615123540+001
TEPEJ	.243874415873250818+007	INCNTR	39

EVNDT \*\*\* DIVERGENCE

IEV( 1) 2 2 605 1173 14 -0 .51589908-002 -.17347235-017

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279280+005 SEC (1965 JUN 6 19:45:27); COMPUTED 25 SEP 70 18:15:20

RSCE03	-.157244479098136979+009	.318117720416785605+008	.141331571367142137+008
VSCE03	-.242507690435170196+002	.119203930392418903+002	-.520609041991881336+001
RSCE10	-.194847028878296496+009	-.103142417278367705+009	-.443896104095630021+008
VSCE10	.812608565756160410+001	.187742889172192646+002	-.817823589329647885+001
RSCE04	.143437741842034969+008	.418324572163613574+007	.742895204847060535+006
VSCE04	-.455269819136649691+001	-.149555549409733672+001	.93410608388519278-001
TEPEJ	.243891832324063290+007	INCNTR	171

VIRTUAL MASS 110 EARTH TO MARS WITH GRAPHICS / R JESSUP

2 AT 183 DAYS 3 HR 36 MIN 2 SEC; JULIAN 2438918.0 DAYS .279280+005 SEC (1965 JUN 6 19:49:27); COMPUTED 25 SEP 70 18:15:20

RSCE03	- .157244479098136979+009	.318117720416785605+008	.141331571367142137+008
VSCE03	- .202507690435170196+002	- .119203930392418903+002	- .620609041991881336+001
RSCE10	- .194847022878296496+009	- .103142417278367705+009	- .443886104095630021+008
VSCE10	.812608565756160410+001	- .187742889172192646+002	- .217823589329647885+001
RSCE04	.143437741842034969+008	.418324572163613574+007	- .742895204847060535+006
VSCE04	- .455269819136649691+001	- .149555549409733672+001	.934106083888519278-001
TEPEJ	.243891832324063290+007	INCNT 171	

3 AT 221 DAYS 10 HR 58 MIN 9 SEC; JULIAN 2438956.0 DAYS .544543+009 SEC (1965 JUL 15 3: 7:34); COMPUTED 25 SEP 70 18:15:31

RSCE03	- .214132796112941517+009	- .286193760880640609+008	- .122114088372728523+008
VSCE03	- .125523261213441699+002	- .241643609287399452+002	- .105078815435830248+002
RSCE10	- .156652766973040836+009	- .157768886511543777+009	- .682163330419719304+008
VSCE10	.145300720801269252+002	- .139355717759905850+002	- .607159523034047324+001
RSCE04	.256629613645570736+005	- .111904771495006057+006	- .567318070167339697+005
VSCE04	- .431668753725688159+001	- .114476081829608474+001	.305650897790385680+000
TEPEJ	.243895663025804199+007	INCNT 317	

# SPACE CRAFT AND VIRTUAL MASS DEPARTING EARTH

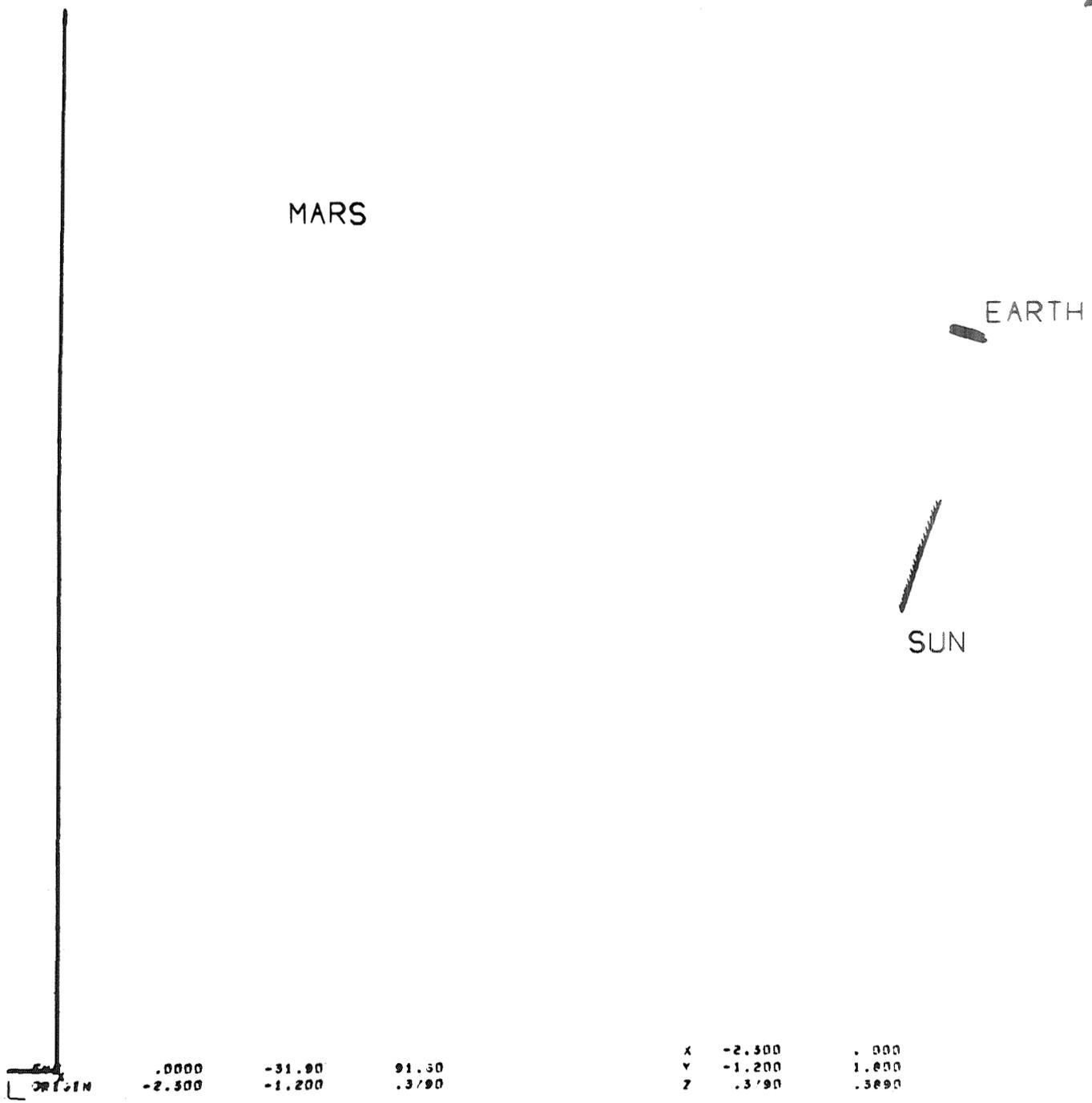


FIGURE 5 - EARTH TO MARS TRAJECTORY: VIEW 1

TRANS MARS MOVEMENT

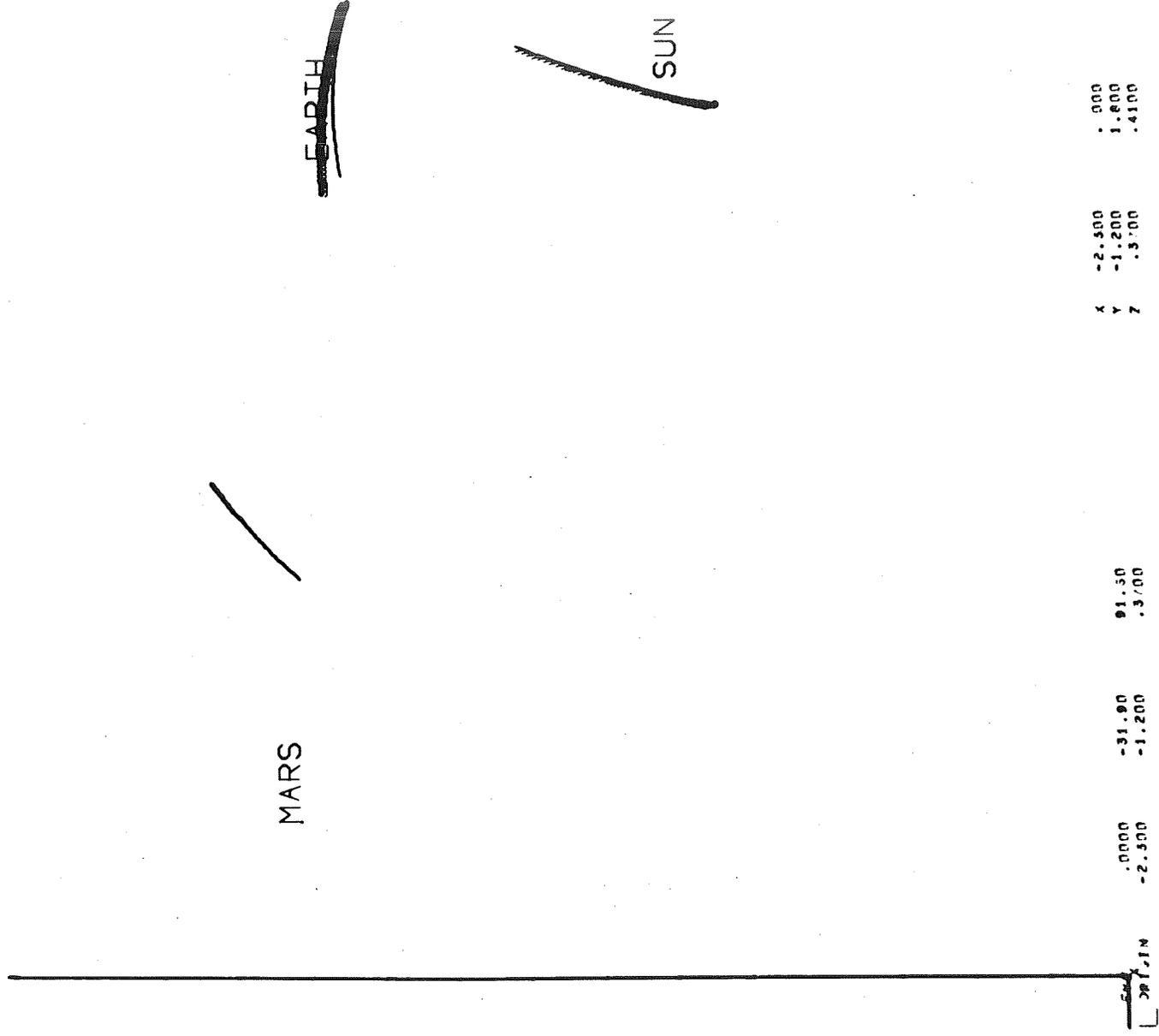


FIGURE 6 - EARTH TO MARS TRAJECTORY: VIEW 2

# TRANS MARS MOVEMENT

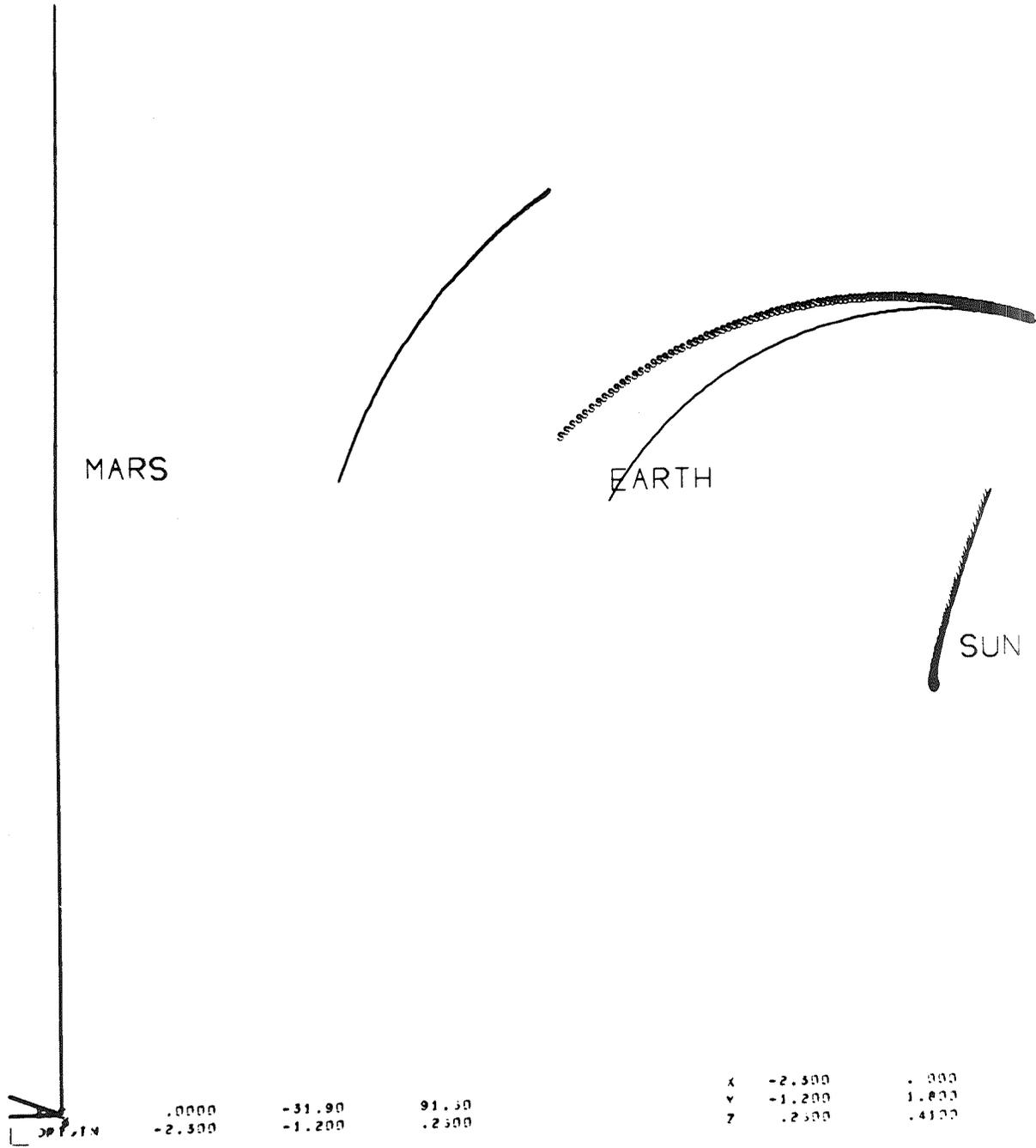


FIGURE 7 - EARTH TO MARS TRAJECTORY: VIEW 3

# SPACE CRAFT AND VIRTUAL MASS APPROACHING MARS

00000000

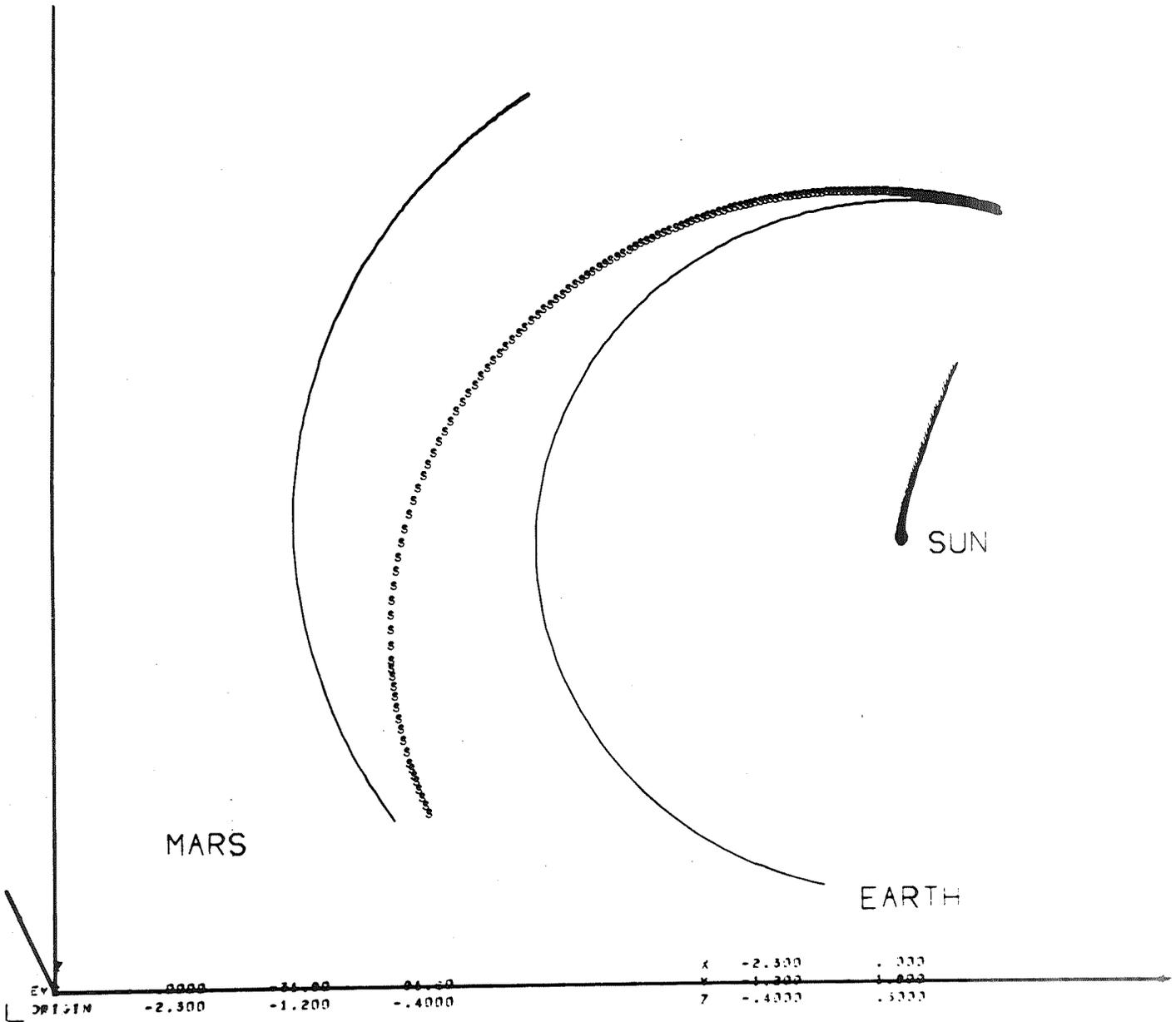


FIGURE 8 - EARTH TO MARS TRAJEACTION: VIEW 4

# SPACE CRAFT AND VIRTUAL MASS APPROACHING MARS

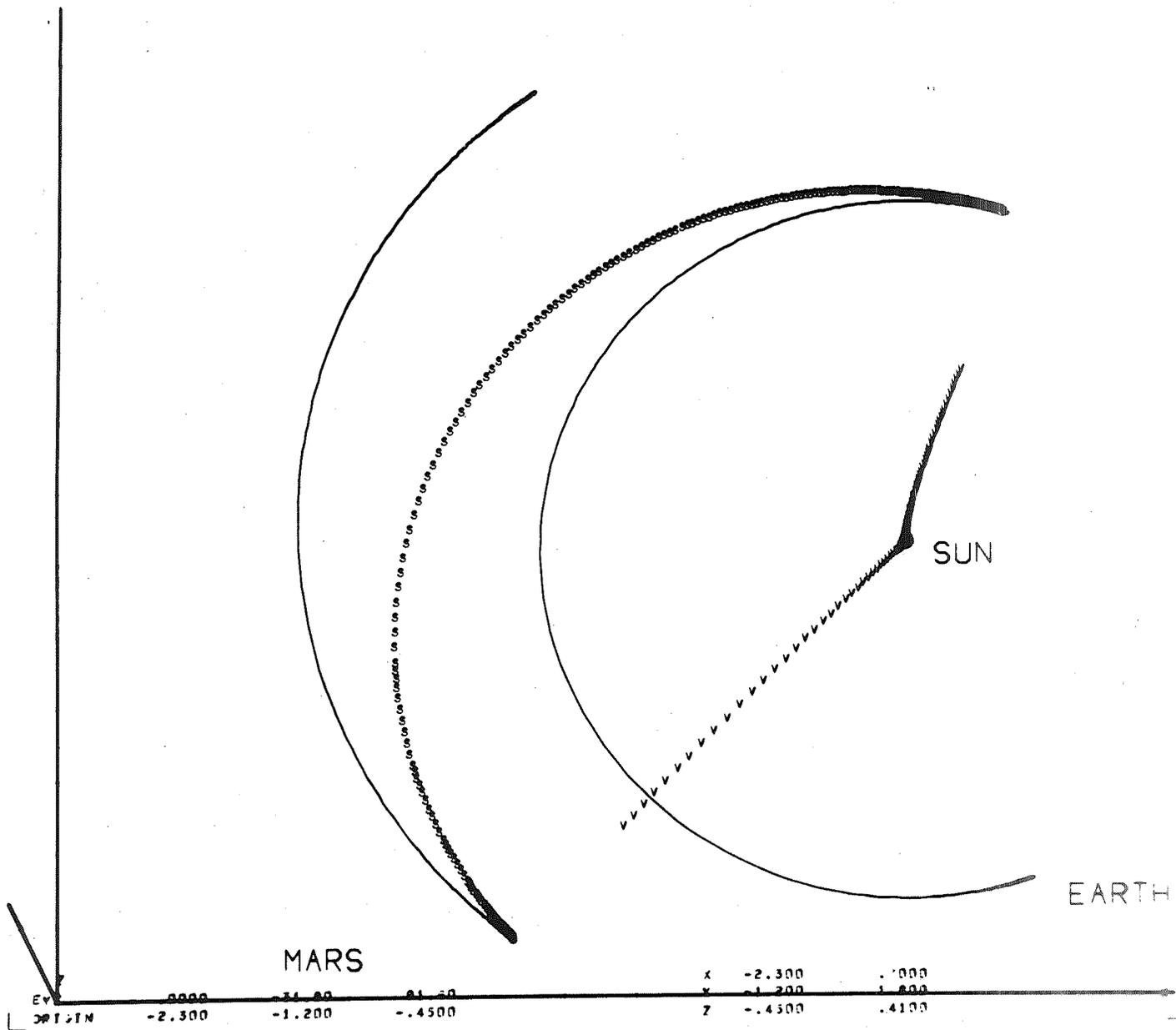


FIGURE 9 - EARTH TO MARS TRAJECTORY: VIEW 5

# SPACE CRAFT AND VIRTUAL MASS APPROACHING MARS

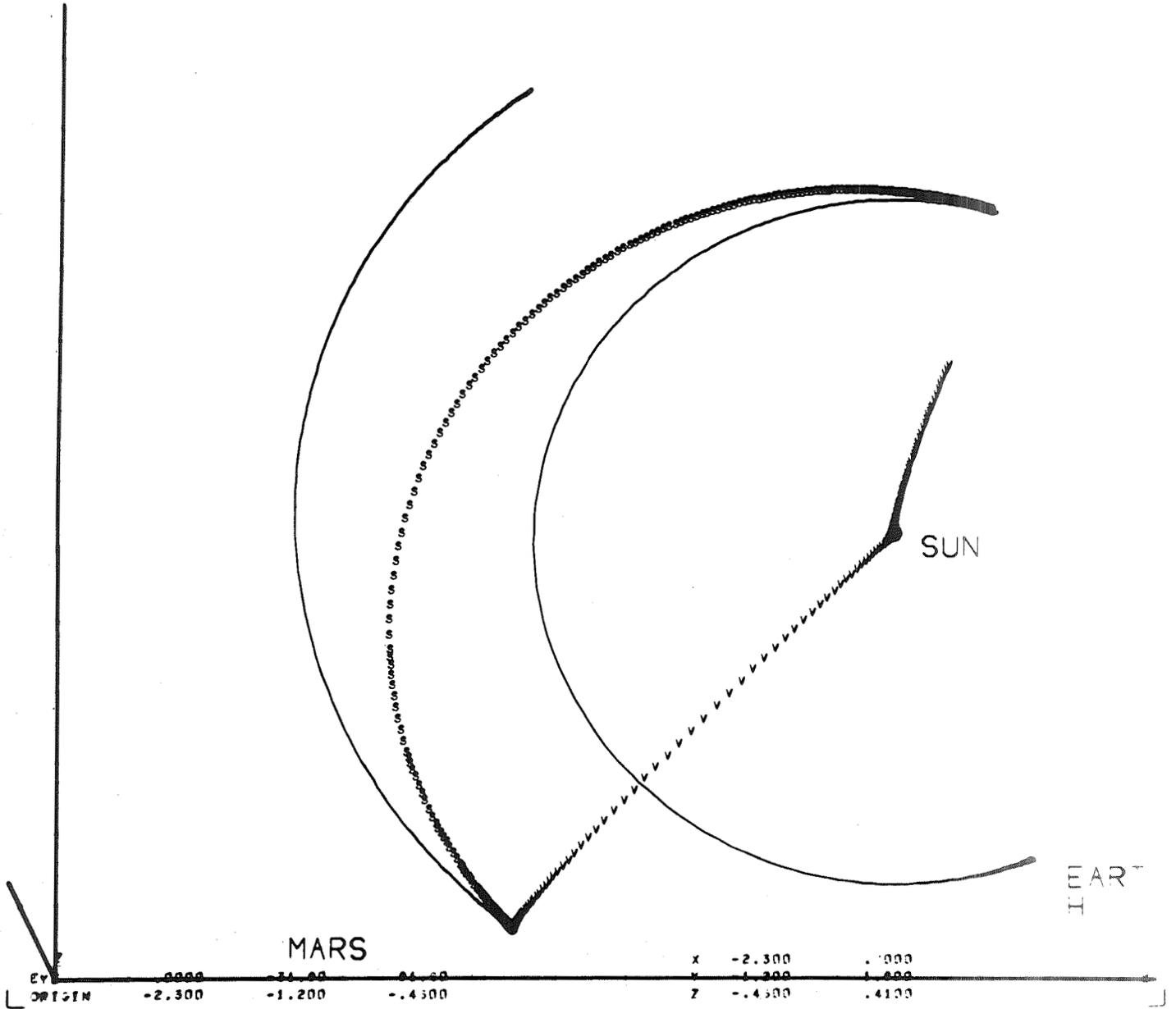


FIGURE 10 - EARTH TO MARS TRAJECTORY: VIEW 6

## Restricted 3-Body Trajectory Instructions

```

VMASS*RFJK.RUN300
1   @ HDG      V MASS EARTH-MOON WITH EP400 / R JESSUP
2   @ PDP,FLI QEVENT/RUN300
3   QEVENT PROC
4   C          REFERENCE: J H NOVAK, 'VIRTUAL MASS TECHNIQUE FOR COMPUTING
5   C          SPACE TRAJECTORIES,' ER 14045, CONTRACT NAS 9-4370,
6   C          MARTIN COMPANY, JANUARY 1966.
7   C          CALL EVNACC ('START', F0, 'START', TTRE, F0, $310)
8   C          EVENT START ONE TIME CALCULATIONS. CONVERT INPUT TO DIMENSIONLESS
9   C          UNITS FOR COMPATABILITY WITH 2-BODY EPHEMERIS
10          G02 = .01214328900
11          G01 = F1 - G02
12          RJUNIT = 207747.200
13          TJUNIT = RT00 / .5490149300
14          VJUNIT = RJUNIT / TJUNIT
15          CALL SCALE (F1 / RJUNIT, RSC00K, RSCEIN)
16          CALL SCALE (F1 / VJUNIT, VSC00K, VSCEIN)
17  C          INITIALIZE SIMULATOR
18          TEE = - 93.59117700 / TUNIT
19          TJT00A = TUNIT / F24
20          CALL EPHEM
21          CALL VMXACT
22          LPRNOW = .TRUE.
23          GO TO 9915
24  310 CALL EVNACC ('MDCORS', 'START', 'MDCORS', 70.00, TTREHR, $320)
25          GO TO 9915
26  320 CALL EVNACC ('CUTOFF', 'MDCORS', 'CUTOFF', A2, F0, $340)
27          IEVSTP = IEVCUR
28          LPRNOW = .FALSE.
29          GO TO 9915
30  340 CONTINUE
31  END
32  @ PDP,FLI QAJXEQ/RUN300
33  QAJXEQ PROC
34          TTREHR = TTRE * TUNIT
35          ENERGY = (F2 * (G01 / RSC001 + G02 / RSC002)
36  1          - F2 * (RSCEIN(2) * VSCEIN(1) - RSCEIN(1) * VSCEIN(2))
37  2          - DOT (VSCEIN, VSCEIN)) * VJUNIT * VJUNIT
38  C          AUXILLIARY EQUATIONS FOR CALCULATIONS CONTROLLING EVENTS MDCORS
39  C          AND CUTOFF
40          IF (IEVCUR .EQ. 'START') RETURN
41          IF (IEVCUR .EQ. 'MDCORS') A2 = DOT (RSC002, VSCE02)
42  END
43  @ PDP,FLI QEOCEQ/RUN300
44  QEOCEQ PROC
45  END
46  @ PDP,FLI QPRINT/RUN300
47  QPRINT PROC
48  C          PRINT SPACE CRAFT STATE IN INPUT UNITS
49          NAME LIST / PRT001 / INCNTR, TTREHR, ENERGY, RSCXIN, VSCXIN
50          CALL SCALE (RJUNIT, RSCEIN, RSCXIN)
51          CALL SCALE (VJUNIT, VSCEIN, VSCXIN)
52          WRITE (IOPRT, PRT001)
53  END
54  @ PDP,FLI QSTORE/RUN300
55  QSTORE PROC
56          EQUIVALENCE          (VAC(1), TTREHR),          (VAC(2), A1          ),

```

```

57      1  (VAC(3), A2      ),  (VAC(4), RUNIT ),  (VAC(5), TUNIT ),
58      2  (VAC(6), VUNIT ),  (VAC(7), TEE   ),  (VAC(8), ENERGY),
59      3  (VAC(9), RSCXIN),  (VAC(10), VSCXIN),
60      4  (G00  , G01   ),  (G00(20)  , G02   ),
61      5  (RSCE00, RSCE01),  (RSCE00(22), RSCE02),
62      6  (VSCE00, VSCE01),  (VSCE00(22), VSCE02),
63      7  (RSCD00, RSCD01),  (RSCD00(22), RSCD02)
64      DIMENSION RSCXIN(3), VSCXIN(3)
65      END
66      @ COPY,S  LVMASS.COMMON,COMMON
67      @ COPY,S  LVMASS.SIMXXX,SIMXXX
68      @ COPY,S  LVMASS.PRTXXX,PRTXXX
69      @ COPY,S  LVMASS.EPH2CO,EPH2CO
70      @ FOR,W   SIMXXX,SIMXXX
71      @ FOR,W   PRTXXX,PRTXXX
72      @ FOR,W   EPH2CO,EPH2CO
73      @ MAP     LVMASS.MAP,MAP300
74      @ XQT     MAP300
75      $DATAOK
76      RSCDOK = -1126.08800, -5433.09510n, 195.972700,
77      VSCDOK = 18364.87900, 3152.53210n, 10624.889000,
78      LSTDOK = .TRUE.,
79      $END

```



## Restricted 3-Body Trajectory with Graphics

VMASS#RFJK.RJN310

```

1   @ HDG      V MASS EARTH-MOON WITH EPHEM AND OVERLAYS / R JESSJP
2   @ PDP,FLI SEVENT/RJN310
3   @EVENT PROC
4   C   REFERENCE: D H NOVAK, 'VIRTUAL MASS TECHNIQUE FOR COMPUTING
5   C   SPACE TRAJECTORIES,' ER 14045, CONTRACT NAS 9-4370,
6   C   MARTIN COMPANY, JANUARY 1966.
7   C   CALL EVNACC ('START', F0, 'START', TTRE, F0, $310)
8   C   EVENT START ONE TIME CALCULATIONS. CONVERT INPUT TO DIMENSIONLESS
9   C   UNITS FOR COMPATABILITY WITH 2-BODY EPHEMERIS
10          G02 = .01214328900
11          G01 = F1 - G02
12          RJNIT = 207747.200
13          TJNIT = RTOD / .5490149300
14          VJNIT = RJNIT / TJNIT
15          CALL SCALE (F1 / RJNIT, RSCDDK, RSCEIN)
16          CALL SCALE (F1 / VJNIT, VSCDDK, VSCEIN)
17   C   INITIALIZE SIMULATOR
18          TEE = - 93.59117700 / TJNIT
19          TJTODA = TJNIT / F24
20          CALL EPHEM
21          CALL VMXACT
22          IEVREP = IEVCUR
23          IENVA = 'ENVAR' + IRUN
24          ISTEP = 'STEPS' + IRUN
25          ITTRE = 'TTREHR' + IRUN
26          IRUN = IRUN + 1
27          DO 112 I = 1, 3
28              CALL CULOJT ('RVMEUW' + I)
29              CALL CULOJT ('R02EUW' + I)
30   112      CALL CULOJT ('RSCEUW' + I)
31          LPRNOW = .TRUE.
32          GO TO 9915
33   310      CALL EVNACC ('MDCORS', 'START', 'MDCORS', 70.00, TTREHR, $320)
34          CALL EOCEQ
35          GO TO 9915
36   320      CALL EVNACC ('CUTOFF', 'MDCORS', 'CUTOFF', A2, F0, $340)
37          CALL EOCEQ
38          IEVSTP = IEVCUR
39          LPRNOW = .FALSE.
40   C   PLOT RESULTS
41          CALL PTX ('RSCEUX', 'RSCEUY')
42          REAL EYE(3) / .1E6, .15E8, .1E7 / ORIGIN(3) / 3*0. /
43          CALL Q3DORG (ORIGIN, 1)
44          CALL Q3DEYE (EYE, 1)
45          CALL Q3DSHA (.0, 3)
46          CALL PLTSIM (1)
47          CALL PLTICS ('RSCEUX', -2000., .15E5)
48          CALL PLTICS ('RSCEUY', -.1E5, .21E6)
49          CALL PLTICS ('RSCEUZ', -100., 9100.)
50          CALL Q3DPER ('RSCEUX', 'RSCEUY', 'RSCEUZ')
51          CALL Q3DSHA (0., 0)
52          CALL Q3DPER ('RVMEUX', 'RVMEUY', 'RVMEUZ')
53          CALL Q3DPER ('R02EUX', 'R02EUY', 'R02EUZ')
54          IF (IRUN .NE. 1) GO TO 314
55          X0 = RSCEIN(1)
56          Y0 = RSCEIN(2)

```

```

57          Z0 = RSCEIN(3)
58          GO TO 315
59          314      T1 = (X0-RSCEIN(1))**2 + (Y0-RSCEIN(2))**2 + (Z0-RSCEIN(3))**2
60          CALL COLECT ('DEV', SNGL (SQRT (T1) * RUNIT) * 1.E+04)
61          CALL COLECT ('P', SNGL (RTOPRF) * 1.E+15)
62          CALL PLTOJT
63          CALL PLTDMP
64          316      IF (.NOT. LSTDDK) GO TO 9915
65          CALL PLTSIM (1)
66          CALL PLTLOG (IENVA + 1 - IRUN, 1)
67          CALL PLTICS (IENVA + 1 - IRUN, 1.E+00, 1.E+10)
68          DO 324 I = 1, IRUN
69             ENCODE (K, 322) I
70          322      FORMAT (I3)
71          CALL PLTAGS (K)
72          CALL PXY (ITTRE+I-IRUN, IENVA+I-IRUN)
73          324      CALL QXY (ITTRE+I-IRUN, IENVA+I-IRUN)
74          CALL AJRITE (12, 500, 1024, 180, 15, 'SCALED BY 1.E12')
75          CALL PLTSIM (1)
76          CALL PLTLOG (ISTEP + 1 - IRUN, 1)
77          CALL PLTICS (ISTEP + 1 - IRUN, 1.E+00, 1.E+04)
78          DO 326 I = 1, IRUN
79             ENCODE (K, 322) I
80          CALL PLTAGS (K)
81          CALL PXY (ITTRE+I-IRUN, ISTEP+I-IRUN)
82          326      CALL QXY (ITTRE+I-IRUN, ISTEP+I-IRUN)
83          CALL PLTAGS (0)
84          CALL PLTSIM (0)
85          CALL PLTICS ('DEV', 1.E0, 1.E8)
86          CALL PLTICS ('P', 1.E0, 1.E10)
87          CALL PLTLOG ('DEV', 1)
88          CALL PLTLOG ('P', 1)
89          CALL QXY ('P', 'DEV')
90          CALL AJRITE (12, 500, 1024, 180, 15, 'SCALED BY 1.E04')
91          CALL AJRITE (550, 12, 1024, 090, 15, 'SCALED BY 1.E15')
92          CALL PLTEND
93          CALL EXIT
94          340      CONTINUE
95          END
96          @ PDP,FLI QAUXEQ/RUN310
97          QAUXEQ PROC
98          TTREHR = TTRE * TUNIT
99          ENERGY = (F2 * (G01 / RSCD01 + G02 / RSCD02)
100             1      - F2 * (RSCEIN(2) * VSCEIN(1) - RSCEIN(1) * VSCEIN(2))
101             2      - DOT (VSCEIN, VSCEIN)) * VJUNIT * VJUNIT
102          IF (END .EQ. F0) END = ENERGY
103          C      AUXILLIARY EQUATIONS FOR CALCULATIONS CONTROLLING EVENTS MDCORS
104          C      AND CUTOFF
105          IF (IEVCUR .EQ. 'START') RETURN
106          IF (IEVCUR .EQ. 'MDCORS') A2 = DOT (RSCE02, VSCE02)
107          END
108          @ PDP,FLI QEOCEQ/RUN310
109          QEOCEQ PROC
110          DO 410 I = 1, 3
111             CALL COLECT ('RSCEIN' + I, SNGL (RSCEIN(I) * RUNIT))
112             CALL COLECT ('RVMEIN' + I, SNGL (RVMEIN(I) * RUNIT))
113          410      CALL COLECT ('R02EIN' + I, SNGL (R02EIN(I) * RUNIT))

```

```

114 CALL COLECT (ISTEP, FLOAT (INCNTR))
115 CALL COLECT (ITRE, SNGL (TTR=HR))
116 CALL COLECT (IENVA, A3S (SNGL ((ENERGY - EN0) / EN0)) * 1.E12)
117 END
118 @ POP, FLI @PRINT/RUN310
119 @PRINT PROC
120 NAME LIST / PRT001 / INCNTR, TTR=HR, ENERGY, RSCXIN, VSCXIN
121 CALL SCALE (RUNIT, RSCXIN, RSCXIN)
122 CALL SCALE (VUNIT, VSCXIN, VSCXIN)
123 WRITE (IDPRT, PRT001)
124 END
125 @ POP, FLI @STORE/RUN310
126 @STORE PROC
127 EQUIVALENCE (VAC(1), TTR=HR), (VAC(2), A1 ),
128 1 (VAC(3), A2 ), (VAC(4), RUNIT ), (VAC(5), TUNIT ),
129 2 (VAC(6), VUNIT ), (VAC(7), TEE ), (VAC(8), ENERGY),
130 3 (VAC(9), RSCXIN), (VAC(10), VSCXIN),
131 4 (G00 , G01 ), (G00(2) , G02), (R00EIN(22), R02EIN),
132 5 (RSCE00, RSCE01), (RSCE00(22), RSCE02),
133 6 (VSCE00, VSCE01), (VSCE00(22), VSCE02),
134 7 (RSCD00, RSCD01), (RSCD00(22), RSCD02)
135 DIMENSION RSCXIN(3), VSCXIN(3), R02EIN(3)
136 END
137 @ COPY, S LVMASS.COMMON, COMMON
138 @ COPY, S LVMASS.PRTXXX, PRTXXX
139 @ COPY, S LVMASS.EPH2CO, EPH2CO
140 @ COPY, S LVMASS.SIMXXX, SIMXXX
141 @ FOR, W SIMXXX, SIMXXX
142 @ FOR, W PRTXXX, PRTXXX
143 @ FOR, W EPH2CO, EPH2CO
144 @ MAP LVMASS.MAP, MAP310
145 @ ASG, TH 31, T, TAPE
146 @ XQT MAP310
147 $DATA DK
148 RSCDDK = -1126.08800, -5433.09510n, 195.972700,
149 VSCDDK = 18364.87900, 3152.53210n, 10624.889000,
150 RTOPRF = 1.0-19,
151 $END
152 $DATA DK
153 RTOPRF = 1.0-15,
154 $END
155 $DATA DK
156 RTOPRF = 1.0-10,
157 $END
158 $DATA DK
159 LSTDDK = .TRUE.,
160 RTOPRF = 1.0-05,
161 $END
162 @ ASG, TM PLOTFILE, T, PLOT
163 @ XQT LAUPLT.PHASE2

```

Figure 14

Program Printout Restricted 3-Body Trajectory with Graphics

```

VIRTUAL MASS TRAJECTORY 15 SEP 70 15:06:00
START AT 0 DAYS 0 HR 0 MIN 0 SEC JULIAN .0 DAYS .000000+000 SEC (*** NOV 24 12: 0: 0; COMPUTER 15 SEP 70 15:06:48)

$PRT001
INCNTR = +0
TTRHR = .000000000000000000+000
ENERGY = .7033989738784728160+007
RSCXIN = -.112608800000000000+004, -.5433095099999999990+004,
ENERGY = .195972700000000000+003
VSCXIN = .1936487899999999990+005, .3152532099999999990+004,
          .105248890000000000+005

$END
$PRT001
INCNTR = +500
TTRHR = .9978154827183130810+001
ENERGY = .7033989738749377280+007
RSCXIN = .1235454727831490850+005, .5016813060931723390+005,
ENERGY = .3029834723950083100+004
VSCXIN = -.4663634187511650230+002, .4409571384414596610+004,
          .4045954343637637630+002

$END
$PRT001
INCNTR = +1000
TTRHR = .4792269523513173560+002
ENERGY = .7033989738716197250+007
RSCXIN = .2427014951674744610+004, .1667050851959292180+006,
ENERGY = .4460179588325481300+004
VSCXIN = -.2980090787340987880+003, .1968185404175201900+004,
          -.1733550253990320700+003

$END
$PRT001
INCNTR = +1500
TTRHR = .6536193471130190960+002
ENERGY = .7033989738564804350+007
RSCXIN = -.1870860279089495000+004, .1978962264504271010+006,
ENERGY = .1261499797652221500+004
VSCXIN = -.1091455946199195770+003, .1685699271469121700+004,
          -.2007876804392912910+003

$END
$PRT001
INCNTR = +2000
TTRHR = .6965898197057704300+002
ENERGY = .7033989738567510140+007
RSCXIN = -.1229291945275529760+004, .2054233744888216160+006,
ENERGY = .2777950023160452410+003
VSCXIN = .9918828494102389680+003, .1864058174453735790+004,
          -.3202295018132790310+003

$END

```

V MASS EARTH-MOON WITH EPH2CO AND OVERLAYS / R JESSUP

EVN01 \*\*\* DIVERGENCE  
 IEV( 1) MDCORS MDCORS 1301 601 14 0 .50846147-004 -.1102230-015  
 MDCORS AT 2 DAYS 22 HR 0 MIN 0 SEC; JULIAN .0 DAYS .000000+000 SEC (\*\*\* NOV 24 12: 0: 0); COMPUTED 15 SEP 70 15:07:27

\$PRT001  
 INCNTR = +2154  
 TTRHR = .70000000000000000010+002  
 ENERGY = .7033989738663854950+007  
 RSCKIN = -.7786671894276625950+003, .2060337240408850040+006,  
 .1564250156999047880+003  
 VSCKIN = .1746081693902975160+004, .1628318262595714460+004,  
 -.4018806266350590310+003  
 \$END  
 MDCORS AT 2 DAYS 22 HR 0 MIN 0 SEC; JULIAN .0 DAYS .000000+000 SEC (\*\*\* NOV 24 12: 0: 0); COMPUTED 15 SEP 70 1

\$PRT001  
 INCNTR = +2154  
 TTRHR = .70000000000000000010+002  
 ENERGY = .7033989738663854950+007  
 RSCKIN = -.7786671894276625950+003, .2060337240408850040+006,  
 .1564250156999047880+003  
 VSCKIN = .1746081693902975160+004, .1628318262595714460+004,  
 -.4018806266350590310+003  
 \$END

EVN01 \*\*\* DIVERGENCE  
 IEV( 1) CUTOFF CUTOFF 605 -115 13 0 -.12810293-004 -.17300741-019  
 CUTOFF AT 2 DAYS 22 HR 20 MIN 19 SEC; JULIAN .0 DAYS .000000+000 SEC (\*\*\* NOV 24 12: 0: 0); COMPUTED 15 SEP 70 1

\$PRT001  
 INCNTR = +2400  
 TTRHR = .7033875285751197090+002  
 ENERGY = .7033989738666801900+007  
 RSCKIN = .5541931646782329460-001, .2063730363999001760+006,  
 .1543725018157747560-001  
 VSCKIN = .2693238215660974620+004, .8832681890109800380-001,  
 -.5044340419478941760+003  
 \$END



## V MASS EARTH-MOON WITH EPH200 AND OVERLAYS / R JESSUP

\$PRT001			
INCNT	=	+533	
TTREHR	=	.703387527405850257D+002	
ENERGY	=	.703398968893852078D+007	
RSCXIN	=	.556258211240098226D-001,	.206373036471956097D+006,
		.154723814276197593D-001	
VSCXIN	=	.269323809276369012D+004,	.884395020287538311D-001,
		-.504434140338039995D+003	

\$END



V MASS EARTH-MOON WITH EP-2CO AND OVRRLAYS / R JESSUP

VIRTUAL MASS TRAJECTORY 15 SEP 70 15:07:54

START AT 0 DAYS 0 HR 0 MIN 0 SEC JULIAN .0 DAYS .000000+000 SEC (\*\*\*\* NOV 24 12: 0: 0) COMPUTED 15 SEP 70 15:07:54

\$PRT001  
 INCNTR = +0  
 ITREHR = .000000000000000000J+000  
 ENERGY = .703398973878472816J+007  
 RSCXIN = -.112608800000000000J+004, -.54330950999999998J+004,  
 .195972700000000000J+003  
 VSCKIN = .18354878999999999J+005, .31525320999999999J+004,  
 .105248890000000000J+005

\$END  
 WDCORS AT 2 DAYS 22 HR 0 MIN 0 SEC JULIAN .0 DAYS .000000+000 SEC (\*\*\*\* NOV 24 12: 0: 0) COMPUTED 15 SEP 70 15:07:55

\$PRT001  
 INCNTR = +23  
 ITREHR = .700000009250060634J+002  
 ENERGY = .701441934773541375J+007  
 RSCXIN = -.535213590055575204J+003, .206205140786107953J+006,  
 .173831996214953321J+003  
 VSCKIN = .182200020259774286J+004, .149619469167857785J+004,  
 -.435821591018548576J+003

\$END  
 WDCORS AT 2 DAYS 22 HR 0 MIN 0 SEC JULIAN .0 DAYS .000000+000 SEC (\*\*\*\* NOV 24 12: 0: 0) COMPUTED 15 SEP 70 15:07:55

\$PRT001  
 INCNTR = +23  
 ITREHR = .700000009250060634J+002  
 ENERGY = .701441934773541375J+007  
 RSCXIN = -.535213590055575204J+003, .206205140786107953J+006,  
 .173831995214953321J+003  
 VSCKIN = .182200020259774286J+004, .149619469167857785J+004,  
 -.435821591018548576J+003

\$END  
 CUTOFF AT 2 DAYS 22 HR 18 MIN 4 SEC JULIAN .0 DAYS .000000+000 SEC (\*\*\*\* NOV 24 12: 0: 0) COMPUTED 15 SEP 70 15:07:55

\$PRT001  
 INCNTR = +25  
 ITREHR = .703011475454539789J+002  
 ENERGY = .701559299123461998J+007  
 RSCXIN = .376131863458108257J+002, .205485634504034547J+006,  
 .243964990591502517J+002  
 VSCKIN = .251704720789947284J+004, .140654821069277357J+003,  
 -.542668792892040924J+003

\$END

RSCEU7

EYE	.1000+06	.3000+08	-1.1000+37	RSCEU7	-2000.	2000.00
ORIGIN	-2000.	-1.1000+05	-100.0	RSCEU5	-.1000+05	.2100+06
				RSCEU7	-100.0	9100.

FIGURE 15 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF =  $10^{-19}$ )

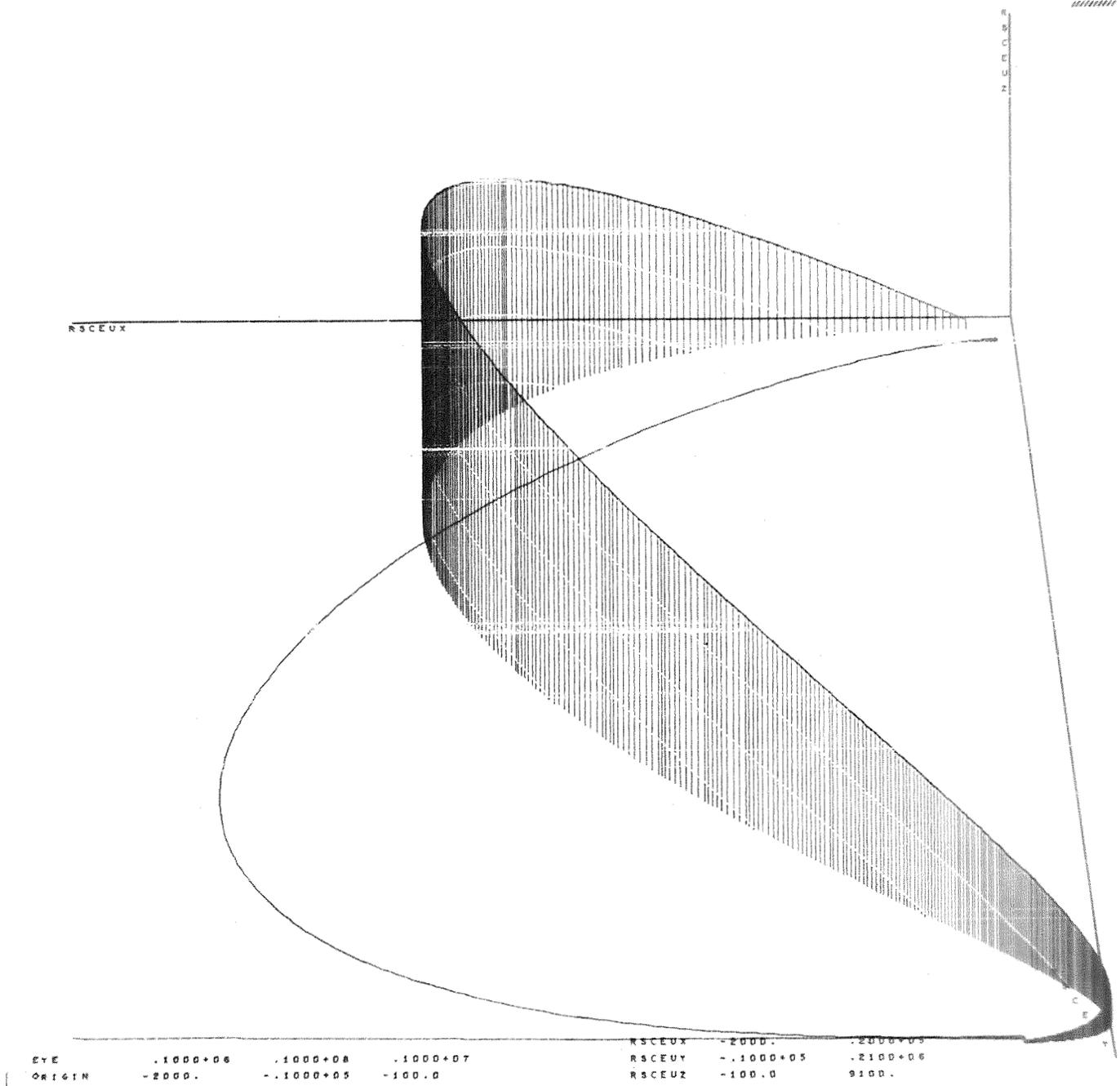
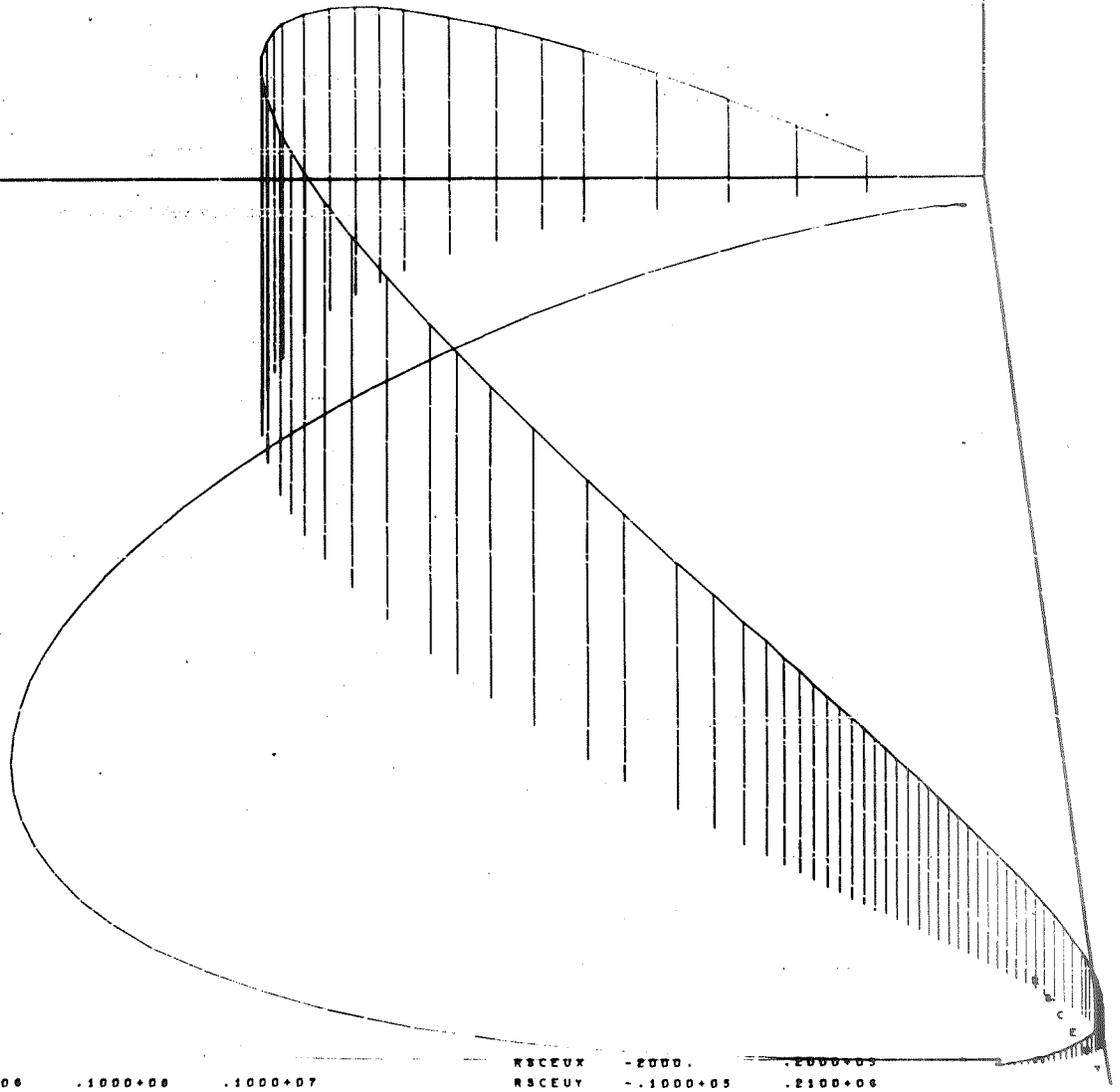


FIGURE 16 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF =  $10^{-15}$ )

///////

P  
S  
E  
U  
N

RSCEUX



EYE	.1000+06	.1000+06	.1000+07
ORIGIN	-2000.	-.1000+05	-100.0

RSCEUX	-2000.	.2000+05
RSCEUY	-.1000+05	.2100+06
RSCEUZ	-100.0	9100.

FIGURE 17 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF =  $10^{-10}$ )

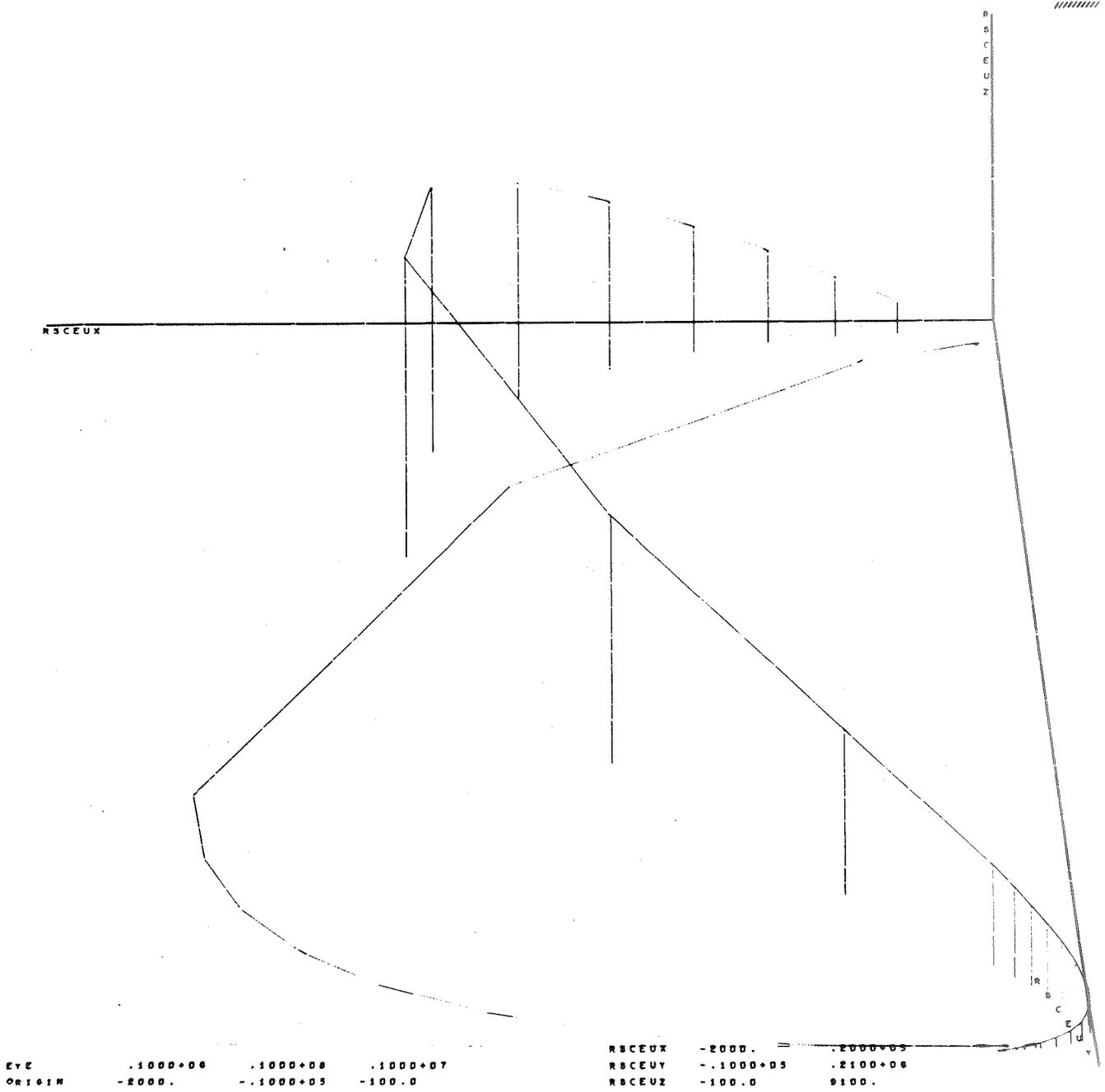


FIGURE 18 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF =  $10^{-5}$ )

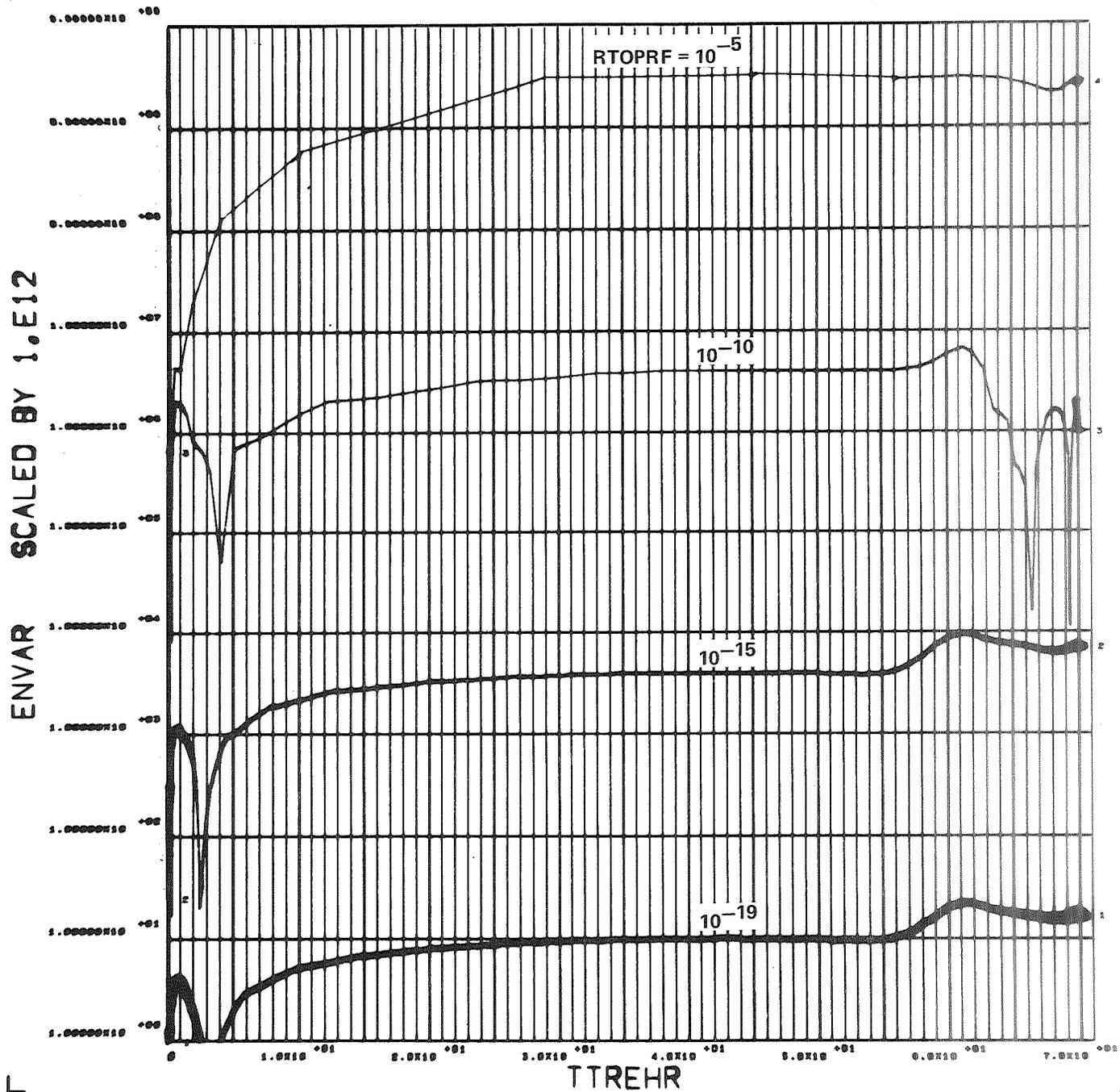


FIGURE 19 - JACOBI ENERGY VARIATIONS VS TRAJECTORY TIME

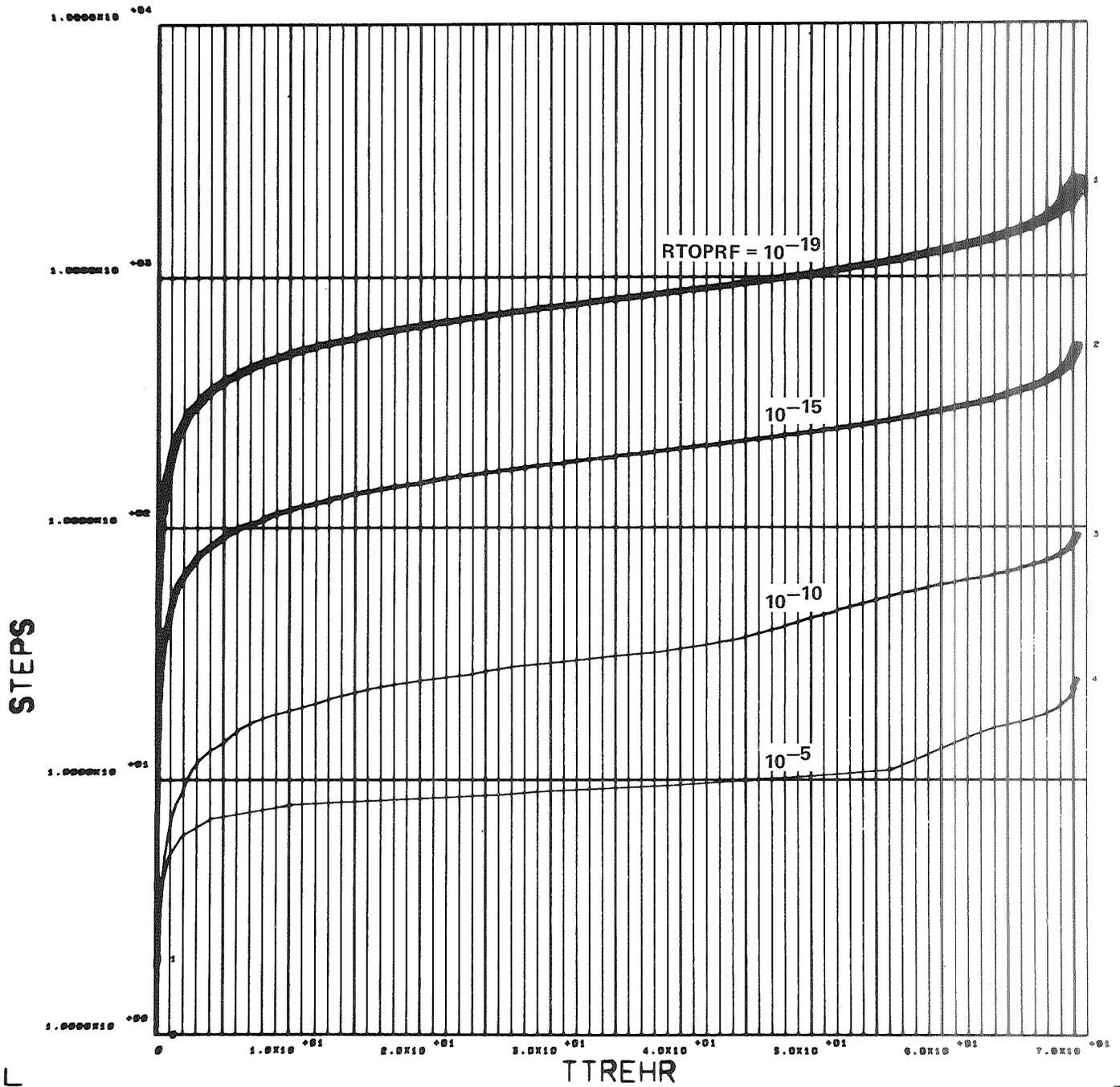


FIGURE 20 - COMPUTING STEPS VS TRAJECTORY TIME

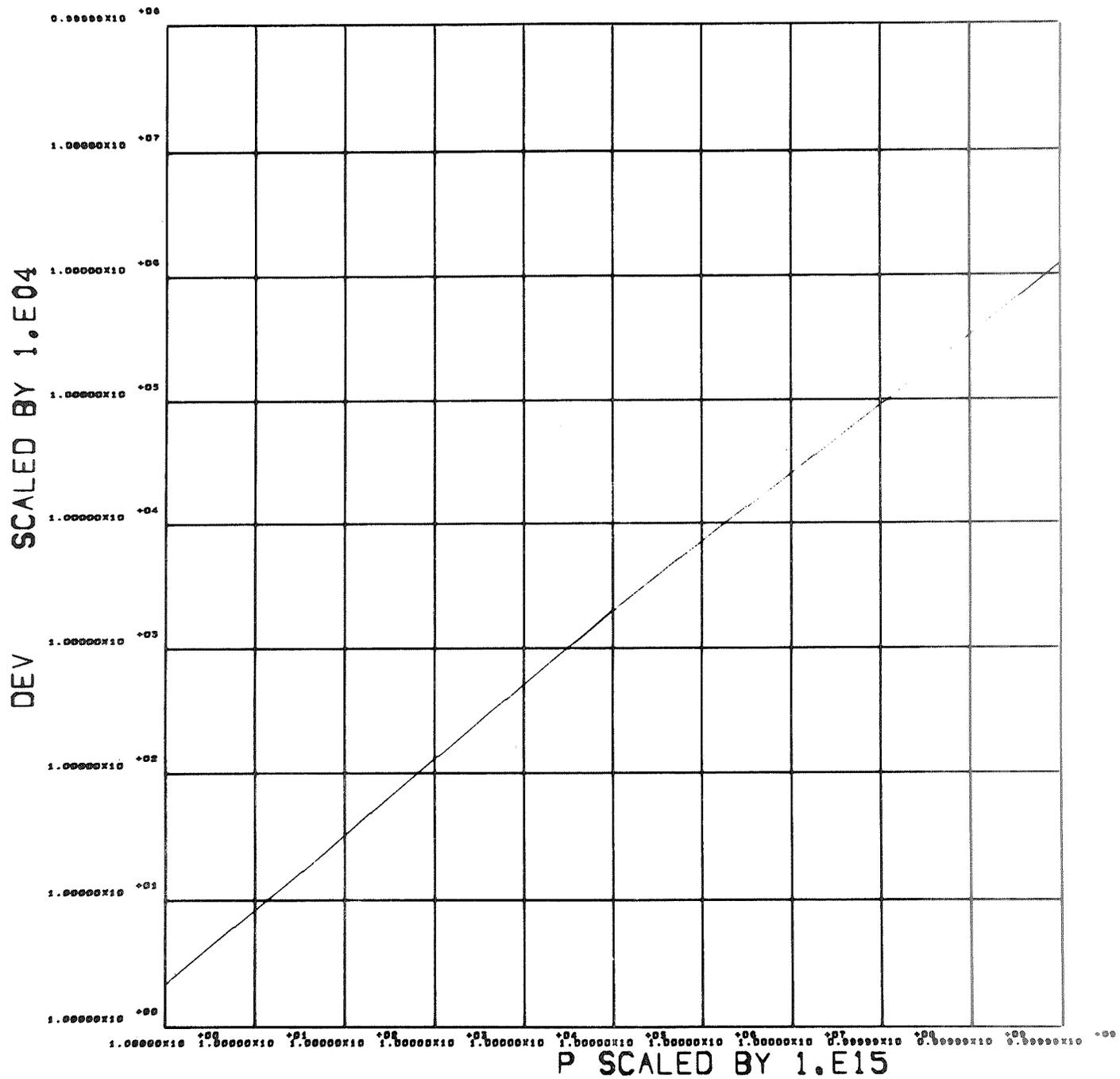


FIGURE 21 - FINAL POSITION VARIATION VS TRAJECTORY REFERENCE PRECISION

Table 1

Virtual Mass Program Symbol Conventions

Most of the VMASS program symbols are formed in accordance with the following conventions. The general intent of the conventions is to convey information about physical quantities, indicating what body or system they are referred to and what coordinate and/or time system they are measured in. As shown in Table 2, some combinations suggested by the convention are not used in the present version of VMASS, but are reserved for future expansion of the program.

<u>Position</u>	<u>Character</u>	<u>Physical Quantity</u>
1	A	Acceleration
	D	Deflection or angle
	E	Eccentricity
	G	Mass times Universal Gravitation Constant
	H	Angular Momentum
	Q	Mass rate times Universal Gravitation Constant
	R	Position
	T	Time
	V	Velocity
	W	Angular rate
<u>Position</u>	<u>Character</u>	<u>System Corresponding to Physical Quantity</u>
2,3	AT	Atomic time
	SC	Spacecraft
	VM	Virtual Mass
	EP	Ephemeris

Table 1 (Contd.)

<u>Position</u>	<u>Character</u>	<u>System Corresponding to Physical Quantity</u>			
2,3	TR	Trajectory			
	RC	Reference conic			
	UN	Universal time			
	01 or ME	Mercury			
	02 or VE	Venus			
	03 or EA	Earth			
	04 or MA	Mars			
	05 or JU	Jupiter	(Numbers indicate non-rotating planetocentric systems; letters indicate rotating planetocentric systems.)		
	06 or SA	Saturn			
	07 or UR	Uranus			
	08 or NE	Neptune			
09 or PL	Pluto				
10 or SU	Sun				
11 or MO	Moon				
<u>Position</u>	<u>Character</u>	<u>Point at Which Quantity Evaluated</u>			
4	E	End of computing interval			
	B	Beginning of computing interval			
	D	Magnitude corresponding to E			
	A	Magnitude corresponding to B			
	O	Program start-up			
	I	Increment			

Table 1 (Contd.)

<u>Position</u>	<u>Character</u>	<u>Reference Coordinate System</u>
5,6	IN	Inertial
	VM	Virtual Mass
	01 or ME	Mercury
	02 or VE	Venus
	03 or EA	Earth
	04 or MA	Mars
	05 or JU	Jupiter
	06 or SA	Saturn
	07 or UR	Uranus
	08 or NE	Neptune
	09 or PL	Pluto
	10 or SU	Sun
	11 or MO	Moon

(Numbers indicate non-rotating planetocentric systems; letters indicate rotating planetocentric systems.)

Table 2

Definitions of VMASS COMMON Symbols

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances*</u>
AU	-	IETBL <sub>1,1</sub>	Astronomical unit expressed in desired READE output units	DDREAD, LODCOM, READE
AUTOKM	-	FAC <sub>33</sub>	Kilometers per AU	RUN100, RUN110,
CT2	-	VAR <sub>12</sub>	Taylor's series term	KEPLER
DANEOO	1	EPH <sub>17,1</sub>	Right ascension	PLANET
DLNEOO	1	EPH <sub>19,1</sub>	Longitude of node at reference time	PLANET
DOBEOO	1	EPH <sub>18,1</sub>	Obliquity	PLANET
DTAI	-	VAR <sub>15</sub>	True anomaly increment	DDREAD, LODCOM, STEPIN
DTAIDN	-	VAR <sub>16</sub>	Decrement factor for DTAI changes	DDREAD, LODCOM, STEPIN
DTAIMN	-	VAR <sub>17</sub>	Minimum value for DTAI	DDREAD, LODCOM, STEPIN
DTAIUP	-	VAR <sub>18</sub>	Increment factor for DTAI changes	DDREAD, LODCOM, STEPIN
DTOR	-	FAC <sub>31</sub>	Radians per degree	LODCOM

---

\*See also elements COMMON and PRTVAR

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
EMRAT	-	IETBL <sub>1,7</sub>	Earth-Moon mass ratio	DDREAD, LODCOM,
EPH			Ephemeris data array	DDREAD, EPHEM EPH2CO, PRTIME
ERCAVM	-		ERCBVM	CONIC, CONICT, KEPLER
ERCBVM	3		Back value of ERCEVM	CONIC, CONICT, KEPLER
ERCEVM	3		Eccentricity vector of reference conic relative to V.M. at begin- ning of interval	
EVN	4,21	IEV <sub>1</sub>	Events table	EVNACC, EVNDT, INDEX
FAC			Multiplying factor data array	DDREAD
FINTES	-	FAC	Small number	CONICT, EVNDT, KEPLER, LODCOM
FTTOKM	-	FAC	Kilometers per foot	LODCOM
F <sub>i</sub>	-	FAC <sub>j</sub>	Floating point constant i, i = 0, 1, 2, ..., appearing throughout the VMAS system	
GOO	1	EPH <sub>15,1</sub>	Array of gravi- tational con- stant times masses	EPH2CO, LODCOM, RUN300, RUN310, VMXACT

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
GOOR	1	EPH <sub>16,1</sub>	Array of GOO data ratio to total mass	EPHEM, LODCOM
GVMB	-	VAR <sub>24</sub>	Back value of GVME	CONIC, CONICT, INDEX, KEPLER, PREPAR
GVME			V.M. magnitude times Univ. Grav. Const. at beginning of interval	
HRCAVM	-	VAR <sub>27</sub>	HRCBVM	CONIC, CONICT, KEPLER
HRCBVM	3	VAR <sub>95</sub>	Back value of HRCEVM	CONIC, CONICT, KEPLER
HRCEVM			Reference conic angular momentum relative to V.M. at beginning of interval	
ICW	-	IETBL <sub>2,1</sub>	Indicator for initial call to ephemeris files	EPHEM, GETTA, LODCOM
ICENT	-	IETBL <sub>2,2</sub>	Indicator for central body in ephemeris data	EPHEM, LODCOM, READE
IDTRAJ	3	ISW <sub>42</sub>	Input trajectory identification	DDREAD, PRTRAJ, SETUP
IEV	8,21	EVN <sub>1</sub>	Events table, integer form	EVNACC, EVNDT, INDEX

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
IEVCUR	-	ISW <sub>3</sub>	Current event	EVNACC, PRTIME, PRTRAJ, ROLLIO, RUN100, RUN110, RUN300, RUN310, SETUP
IEVFLG	-	ISW <sub>4</sub>	Events table update flag	EVNACC, EVNDT, PRTRAJ, SIMXXX,
IEVM	-	ISW <sub>1</sub>	Events table maximum size	EVNACC, LODCOM,
IEVN	-	ISW <sub>2</sub>	Events table current size	EVNACC, EVNDT, INDEX
IEVREP	2	ISW <sub>45</sub>	Events inter- val to be repeated	DDREAD, RUN100, RUN310, SETUP
IEVRUN	2	ISW <sub>47</sub>	Events inter- val to be simulated	DDREAD
IEVSTP	-	ISW <sub>6</sub>	Stopping event	DDREAD, PRTRAJ, RUN100, RUN110, RUN300, RUN310, SIMXXX
IEVSTR	-	ISW <sub>7</sub>	Starting event	DDREAD, PRTRAJ, SIMXXX
INCNTR	-	ISW <sub>9</sub>	Computing increment counter	INDEX, RECORD, RUN100, RUN110, RUN300, RUN310
IOCDR	-	IOU <sub>1</sub>	Input card reader	DDREAD, LODCOM, SETUP
IOEPH	-	IOU <sub>4</sub>	Ephemeris file	DDREAD, GETTA, LODCOM
IOMES	-	IOU <sub>5</sub>	Diagnostic message printer	CONICT, DDREAD, EPHEM, EVNACC, EVNDT, KEPLER, LODCOM, PRTRAJ, SETUP, SIMXXX

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
IOPCH	-	IOU <sub>3</sub>	Output card punch	DDREAD, LODCOM
IOPRL	-	IOU <sub>7</sub>	Print later file	DDREAD, LODCOM, PRTRAJ, RECORD, ROLLIO
IOPRT	-	IOU <sub>2</sub>	Output printer	EPHEM, LODCOM, PRTIME, RECORD, ROLLIO, RUN100, RUN110, RUN300, RUN310, SETUP, STEPIN
IOROL	-	IOU <sub>8</sub>	Roll back file	DDREAD, LODCOM, ROLLIO, SETUP
IOSCR	-	IOU <sub>6</sub>	Scratch file	DDREAD, LODCOM
IOU	25	KVAR <sub>951</sub>	Input-output file name array	DDREAD
IPLATM	-	ISW <sub>10</sub>	Computing increments between writes to IOPRL	DDREAD, LODCOM, RECORD
IPLOTM	-	ISW <sub>12</sub>	Computing increments between calls for plots	DDREAD, LODCOM, RUN110
IPNOWM	-	ISW <sub>13</sub>	Computing increments between calls to PRTOU	DDREAD, LODCOM, RECORD
IREQ	13	IETBL <sub>23</sub>	READE table of output requests	EPHEM, LODCOM, READE

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
IRTIME	3	ISW <sub>24</sub>	Current trajectory identification	PRTRAJ, SETUP
ISW	50	KVAR <sub>701</sub>	Integer variables data array	DDREAD
ITEPOG	6	-	Initial ephemeris time in Gregorian units	DDREAD
ITRAT	-	ISW <sub>15</sub>	STEPIN iteration counter	PREPAR, STEPIN
IVAR	1000	VAR <sub>1</sub>	Array VAR in integer form	
JK	100	KVAR <sub>751</sub>	Array of indices to gravitating bodies	DDREAD, EPHEM, EPH2CO, INDEX, LODCOM, PLANET, VMXACT
JKN	-	ISW <sub>16</sub>	Current size of JK	DDREAD, EPHEM, EPH2CO, INDEX, LODCOM, VMXACT
JKPTSN	-	ISW <sub>18</sub>	Number of gravitating bodies given mass point distributions	EPHEM, PLANET
JKSPEC	-	ISW <sub>20</sub>	Ephemeris data block size	LODCOM, PLANET
KEARTH	-	ISW <sub>32</sub>	Index to earth data in ephemeris blocks	LODCOM, RUN100, RUN110
KEMB	-	ISW <sub>41</sub>	Ephemeris index to Earth-Moon data	LODCOM

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
KEPMAX	-	ISW <sub>17</sub>	Maximum number of terms in KEPLER series	KEPLER, LODCOM
KJUP	-	ISW <sub>34</sub>	Ephemeris index to Jupiter data	LODCOM
KMARS	-	ISW <sub>33</sub>	Ephemeris index to Mars data	LODCOM, RUN100, RUN110
KMERC	-	ISW <sub>30</sub>	Ephemeris index to Mercury data	LODCOM
KNEP	-	ISW <sub>37</sub>	Ephemeris index to Neptune data	LODCOM
KPLUTO	-	ISW <sub>38</sub>	Ephemeris index to Pluto data	LODCOM
KSAT	-	ISW <sub>35</sub>	Ephemeris index to Saturn data	LODCOM
KSUN	-	ISW <sub>39</sub>	Ephemeris index to Sun data	LODCOM, RUN100, RUN110
KURAN	-	ISW <sub>36</sub>	Ephemeris index to Uranus data	LODCOM
KVAR	1000	COMMON	Working data COMMON array	DEVAR, LODCOM, RECORD, ROLLIO
KVENUS	-	ISW <sub>31</sub>	Ephemeris index to Venus data	LODCOM

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
LIMPEV	-	LSW <sub>2</sub>	Flag showing iteration to an event	EVNDT, KEPLER, SIMXXX, STEPIN
LINTDN	-	LSW <sub>18</sub>	Flag showing reduction of computing increment	EVNDT, STEPIN
LLDCOM	-	LSW <sub>14</sub>	Flag showing COMMON is to be loaded	PRTRAJ, SETUP, SIMXXX
LOGOUT	-	LSW <sub>1</sub>	Flag showing subroutine log out on error condition	CONICT, DDREAD, EVNACC, KEPLER, ROLLIO, SETUP, SIMXXX, STEPIN
LOOP	-	LIMPEV	See LIMPEV	KEPLER
LRESKP	-	LSW <sub>16</sub>	Flag showing by-pass of trajectory recording	EVNACC, RECORD
LSTDDK	-	LSW <sub>4</sub>	Flag showing last data deck of a series has been read	DDREAD, RUN100, RUN110, RUN300, RUN310, SETUP
LSW	50	KVAR <sub>901</sub>	Array of logical flags	DDREAD
LXECEV	-	LSW <sub>3</sub>	Flag showing execution of an event	EVNACC, EVNDT, PRTXXX, RECORD, SIMXXX
MONTH	12	COMMON	Table of names of the months	PRTIME, RTCLOK

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
MVAR	-	ISW <sub>10</sub>	Maximum size of array KVAR	LODCOM, SETUP
NMAX	-	ISW <sub>22</sub>	Maximum number of terms in CONICT	CONICT, LODCOM
PHC	50	COMMON	COMMON array for physical constants	DDREAD
QVME	-	VAR <sub>36</sub>	Current value virtual mass rate	INDEX, PREPAR, STEPIN, VMXACT
RE	-	IETBL <sub>13</sub>	Earth radius	DDREAD, LODCOM, READE
RETOKM	-	FAC <sub>34</sub>	Kilometers per RE	LODCOM
RMNEOO	1	EPH <sub>17,1</sub>	Planetographic mass point position	
ROOD	1	EPH <sub>21,1</sub>	Gravitating body radius	
ROOEIN	1	EPH <sub>1,1</sub>	Array of current gravitating bodies inertial positions	EPHEM, EPH2CO, PLANET, RUN100, RUN110, RUN310, VMXACT
RRCAVM	-	VAR <sub>38</sub>	RRCBVM	CONIC, INDEX, PREPAR, STEPIN
RRCBVM	3	VAR <sub>110</sub>	Reference conic position vector relative to the V.M. at beginning of computation interval	CONIC, CONICT, INDEX, KEPLER, PREPAR

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
RRCDVM	-	VAR <sub>39</sub>	RRCEVM	INDEX, KEPLER, PREPAR, STEPIN
RRCEVM	3	VAR <sub>116</sub>	Reference conic posi- tion vector relative to the V.M. at end of compu- tation interval	INDEX, KEPLER, PREPAR
RRCIVM	3	VAR <sub>101</sub>	RRCEVM incre- ment (RRCEVM - RRCBVM)	CONICT, KEPLER
RSCAOO	1	EPH <sub>13,1</sub>	Back values of RSCDOO	INDEX
RSCAVM	-	VAR <sub>38</sub>	RSCBVM	
RSCBIN	3	VAR <sub>122</sub>	Back value of RSCEIN	INDEX
RSCBVM	3	VAR <sub>110</sub>	Back value of RSCEVM	
RSCDDK	3	VAR <sub>137</sub>	Spacecraft position from data deck	DDREAD, RUN100, RUN110, RUN300, RUN310
RSCDOO	1	EPH <sub>14,1</sub>	RSCEOO	INDEX, RUN300, RUN310, VMXACT
RSCDVM	-	VAR <sub>41</sub>	RSCEVM	INDEX, STEPIN, VMXACT
RSCEIN	3	VAR <sub>128</sub>	Inertial spacecraft position at end of com- puting interval	DDREAD, INDEX, RUN100, RUN110, RUN300, RUN310, STEPIN, VMXACT

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
RSCEVM	3		Spacecraft position relative to the V.M. at end of computing interval	
RSCEOO	1	EPH <sub>7,1</sub>	Array of spacecraft positions relative to the gravitating bodies at end of computing interval	RUN100, RUN110, RUN300, RUN310, VMXACT
RTO	-	RTOPRF	See RTOPRF	CONICT
RTOD	-	FAC <sub>30</sub>	Degrees per radian	LODCOM, RUN300, RUN310
RTOPRF	-	VAR <sub>47</sub>	Position tolerance precision reference	DDREAD, LODCOM, RUN100, RUN110, RUN310, STEPIN
RVMBIN	3	VAR <sub>158</sub>	Back value of RVMEIN	INDEX, PREPAR
RVMEIN	3	VAR <sub>164</sub>	Inertial Virtual Mass position at end of computing interval	INDEX, PREPAR, RUN100, RUN110, RUN310, VMXACT
SPLEOO	1	EPH <sub>20,1</sub>	Sidereal rate	PLANET
SV <sub>i</sub>	3	TEM <sub>j</sub>	Temporary vector storage	
S <sub>i</sub>	-	TEM <sub>j</sub>	Temporary scalar storage	

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
TABOUT	6,12	IETBL4 <sub>1</sub>	Ephemeris output from READE	EPHEM, READE
TEE	-	-	Initial ephemeris time, giving orientation of 2-body gravitational system	EPH2CO, RUN300, RUN310
TEPB	-	TTRB	See TTRB	INDEX
TEPE	-	TTRE	See TTRE	INDEX
TEPEJD	-	VAR <sub>1</sub>	Current Julian day	EPHEM, PLANET, PRTIME, RUN100, RUN110
TEPEJS	-	VAR <sub>2</sub>	Current Julian day fraction in seconds	EPHEM, PLANET, PRTIME, RUN100, RUN110
TEPI	-	TTRI	See TTRI	KEPLER, STEPIN
TEPOJD	-	VAR <sub>3</sub>	Initial Julian day	DDREAD, EPHEM, PLANET, PRTIME, RUN100, RUN110
TEPOJS	-	VAR <sub>4</sub>	Initial Julian day fraction in seconds	DDREAD, EPHEM, PLANET, PRTIME, RUN100, RUN110
TEVI	-	VAR <sub>9</sub>	Time increment to nearest event	EVNACC, EVNDT, STEPIN
TPD	-	IETBL1 <sub>5</sub>	Simulation time units per ephemeris day	DDREAD, LODCOM, READE

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
TTO	-	VAR <sub>53</sub>	Time tolerance precision reference	CONICT, EVNDT, KEPLER, STEPIN
TTRB	-	VAR <sub>6</sub>	Back value of TTRE	STEPIN
TTRE	-	VAR <sub>7</sub>	Elapsed trajectory time at end of computing interval	EPHEM, EPH2CO, PRTIME, RUN300, RUN310, STEPIN
TTRI	-	VAR <sub>11</sub>	TIRE increment	KEPLER, STEPIN
TUTODA	-	VAR <sub>54</sub>	Ephemeris days per simulation time units	EPHEM, LODCOM, PRTIME, RUN300, RUN310
TV <sub>i</sub>	3	TEM <sub>j</sub>	Temporary vector storage	-
T <sub>i</sub>	-	TEM <sub>j</sub>	Temporary scalar storage	-
VAC	50		COMMON storage for user-coded trajectory descriptions	DDREAD, RUN100, RUN110, RUN300, RUN310
VAR	300	KVAR <sub>1</sub>	COMMON storage for program working variables	DDREAD, DEVAR, EVNACC
VOOEIN	1	EPH <sub>4,1</sub>	Inertial velocities of gravitating points at end of computing interval	EPHEM, EPH2CO, PLANET, RUN100, RUN110, VMXACT

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
VRCBVM	3	VAR <sub>113</sub>	Back value of VRCEVM	CONIC, INDEX, KEPLER, PREPAR, STEPIN
VRCEVM	3		Reference conic velocity vector relative to the V.M. at end of computing interval	
VSCBIN	3	VAR <sub>125</sub>	Back value of VSCEIN	INDEX
VSCBVM	3	VAR <sub>149</sub>	Back value of VSCEVM	
VSCDDK	3	VAR <sub>140</sub>	Spacecraft velocity from data deck	DDREAD, RUN100, RUN110, RUN300, RUN310
VSCEIN	3	VAR <sub>131</sub>	Inertial spacecraft velocity at end of computing interval	DDREAD, INDEX, RUN100, RUN110, RUN300, RUN310, STEPIN, VMXACT
VSCEOO	1	EPH <sub>10,1</sub>	Spacecraft velocity relative to gravitating body at end of computing interval	RUN100, RUN110, RUN300, RUN310, VMXACT
VSCEVM	3	VAR <sub>155</sub>	Spacecraft velocity relative to the V.M. at end of computing interval	INDEX, PREPAR, STEPIN, VMXACT
VVMBIN	3	VAR <sub>161</sub>	Back value of VVMEIN	INDEX, PREPAR

Table 2 (Contd.)

<u>Symbol</u>	<u>Dimension</u>	<u>Equivalence</u>	<u>Definition</u>	<u>Appearances</u>
VVMEIN	3	VAR <sub>167</sub>	Inertial virtual mass velocity at end of computing interval	INDEX, PREPAR, VMXACT
XV <sub>i</sub>	3	TEM <sub>j</sub>	Temporary vector storage	-
X <sub>i</sub>	-	TEM <sub>j</sub>	Temporary scalar storage	-

Table 3

VMASS Error and Diagnostic Printouts

CONICT \*\*\* TIME SERIES DIVERGENT

This error message indicates a divergent series calculation in subroutine CONICT ( $z > 1$ ). It usually results from a too tight precision requirement or from gross error in input data.

CONICT \*\*\* TIME SERIES TERMS EXCEEDING  $i$

This diagnostic printout indicates slow convergence in CONICT. The maximum permissible number of terms may be raised by a change in NMAX. Increasing the value of RTOPRF for looser precision may also be appropriate.

DDREAD \*\*\* DATA DECK LOGOUT

This printout results from a data deck setting of LOGOUT to time. See subroutine SETUP for applications.

EVNACC \*\*\* EVENTS TABLE OVERFLOW

More than IEVM events are scheduled for placement in the events table IEV. Simplify the trajectory description, that is, recode with fewer events, or recompile system. On recompilation it is necessary to increase IEVM in LODCOM and to increase IEV dimensions in COMMON.

EVNDT \*\*\* EVENT DIVERGENCE

On iteration to an event the program has taken more than 12 steps, or the event appears to have been passed. An event is passed if TEVI is negative. Execution of the event is initiated, and a printout is given for the table entries of the event.

KEPLER \*\*\* CONVERGENCE NOT UNIFORM

This error message indicates non-uniform convergence in the series calculation of subroutine KEPLER. It usually results from too tight precision requirements or from a gross error in input data.

Table 3 (Contd.)

KEPLER \*\*\* TERMS IN SERIES EXCEEDING i

This diagnostic printout indicates slow convergence in KEPLER. The maximum permissible number of terms may be raised by a change in KEPMAX. Increasing the value of RTOPRF for looser precision may also be appropriate.

PRTRAJ \*\*\* END-OF-FILE

Processing of the print later file has reached an end of file condition before the specified stopping event. This message may also appear if the search for specified trajectory identification or starting event is not successful. In such a case check the input data values of IDTRAJ and IEVSTR.

READE \*\*\* DATE TOO EARLY FOR EPH

Current Julian date-time group (TEPEJD, TEPEJS) is for later date than that covered by the assigned ephemeris file.

READE \*\*\* IMPROPER OUTPUT REQUEST

The array IREQ does not contain proper entries for gravitating body positions and velocities. Check for modifications following calls to LODCOM.

READE \*\*\* INVALID CENTRAL BODY

The variable ICENT does not contain proper entry for central body selection. Check for modifications following calls to LODCOM.

READE \*\*\* ICW OUT OF RANGE

The variable ICW does not contain proper entry for indicating type of record to be read by READE. Check for modifications following calls to LODCOM.

ROLBCK \*\*\* REQUESTED EVENT NOT ON ROLL BACK FILE

A search for an earlier event recorded on the roll back file is not successful. Check calling sequence.

BELLCOMM, INC.

Table 4

VMASS Program Storage Requirements

<u>Element</u>	<u>Storage Requirements</u>		
	<u>Code</u>	<u>Data</u>	<u>COMMON</u>
COMMON			
CVAR	0	0	1160
CEPH	0	0	420
CFAC	0	0	58
CPHC	0	0	50
CTEM	0	0	100
CMON	0	0	12
CETBL1	0	0	8
CETBL2	0	0	15
CETBL3	0	0	1836
CETBL4	0	0	148
CETBL5	0	0	156
CETBL9	0	0	7
REC1	0	0	24
REC2	0	0	25
Subtotal	0	0	4019
CONIC	41	9	CVAR, CTEM
CONICT	153	31	CVAR, CTEM, CFAC
CROSS	66	34	
DDREAD	20	152	CVAR, CEPH, CPHC, CFAC, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9

Table 4 (Contd.)

<u>Element</u>	<u>Storage Requirements</u>		
	<u>Code</u>	<u>Data</u>	<u>COMMON</u>
DEVAR	27	9	CVAR, CTEM
DOT	24	9	
DTJTOG	236	40	CFAC, CTEM
EPHEM	182	54	CVAR, CPHC, CTEM, CEPH, CFAC, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
EVNACC	196	24	CVAR, CTEM, CFAC
EVNDT	174	29	CVAR, CTEM, CFAC
GETTA	161	32	CVAR, REC1, REC2, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
INDEX	71	11	CVAR, CEPH
INITAL	10	5	
KEPLER	151	33	CVAR, CFAC, CTEM
LODCOM	359	117	CVAR, CEPH, CFAC, CPHC, CTEM, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
PLANET	195	30	CVAR, CEPH, CTEM, CFAC, CPHC, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
PREPAR	125	11	CVAR, CTEM, CFAC
PRTCOM	195	65	CVAR, CTEM, CFAC, CPHC

Table 4 (Contd.)

<u>Element</u>	<u>Code</u>	<u>Data</u>	<u>COMMON</u>
PRTIME	113	50	CVAR, CTEM, CFAC, CMON
PRTRAJ	77	36	CVAR
PRTFAC	41	299	CVAR, CEPH, CPHC, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
PRTVAR	144	449	CVAR, CEPH, CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
READE	537	164	CETBL1, CETBL2, CETBL3, CETBL4, CETBL5, CETBL9
RECORD	40	5	CVAR
ROLLIO	103	24	CVAR
RTCLOK	49	13	CMON
SCALE	30	14	
SETUP	79	25	CVAR
STEPIN	174	20	CVAR, CFAC, CTEM
UNITV	33	12	CTEM
VALUE	26	10	
VMXACT	150	19	CVAR, CFAC, CPHC, CTEM, CEPH
Subtotal	3982	1835	
RUN100			
SIMXXX	232	80	CVAR, CFAC, CEPH, CTEM
PRTXXX	89	38	CVAR, CFAC, CEPH, CTEM

Table 4 (Contd.)

<u>Element</u>	<u>Storage Requirements</u>		
	<u>Code</u>	<u>Data</u>	<u>COMMON</u>
RUN110			
SIMXXX	321	163	CVAR, CFAC, CEPH, CTEM
PRTXXX	87	38	CVAR, CFAC, CEPH, CTEM
RUN300			
SIMXXX	190	53	CVAR, CFAC, CEPH, CTEM
PRTXXX	27	194	CVAR
EPH2CO	82	11	
RUN310			
SIMXXX	583	178	CVAR, CFAC, CEPH, CTEM
PRTXXX	27	239	CVAR
EPH2CO	82	11	
Subtotal	1720	1005	
Total	5702	2840	

Table 5

## VMASS References to Fortran V Special Features

<u>Identification</u>	<u>Definition</u>	<u>Program Appearances</u>
ENTRY	Location of sub-routine starting point other than at its beginning. Used mainly to combine several small subroutines into one element.	DTJTOG, LODCOM, PRTCOM, PRTVAR, ROLLIO, SIMXXX, PRTFAC
INCLUDE	Edit a Fortran Procedure into an element. Used to edit COMMON elements and user coded trajectory descriptions into appropriate subroutines. Permits maintenance of one copy of the COMMON cards.	CONIC, CONICT, CROSS, DDREAD, DEVAR, DOT, DTJTO6, EPHEM, EPH2CO, EVNACC, EVNDT, GETTA, INDEX, KEPLER, LODCOM, PLANET, PREPAR, PRTCOM, PRTIME, PRTRAJ, PRTVAR, PRTXXX, READE, ROLLIO, SCALE, SETUP, SIMXXX, STEPIN, UNITV, VALUE, VMXACT
PROC	Define a Fortran Procedure, under control of the EXEC VIII PDP system. Use is limited to defining Fortran card sets to be edited into VMASS subroutines. Applies to COMMON and trajectory descriptions.	COMMON, RUN100, RUN110, RUN300, RUN310

APPENDIX

VMASS Program Descriptions

This appendix provides a description of each program element in the VMASS system. The description includes element names and calling sequences, purposes, lower level elements, programming language, computations, flowcharts, and listings. Of the 32 elements in VMASS as implemented on the UNIVAC 1108 computer, 30 are written in Fortran V, and two are written in UNIVAC Assembler code. The other elements include a Fortran V procedure for convenient representation of COMMON storage and an EXEC VIII Map element for composing executable elements.

Error and diagnostic printouts produced by each program are tabulated in Table 3, which also suggests possible remedial actions the system user may take. Each printout begins with originating program name. Program storage requirements are given in Table 4. As an aid to making conversions to other computers, Table 5 sets forth VMASS references to UNIVAC Fortran special features.

## VMASS\*RFUK.COMMON

```

1 QTYPE PROC
2 C
3 C VMASS SYMBOL CONVENTIONS
4 C A - H, O - Z DOUBLE PRECISION
5 C L LOGICAL
6 C I - K, M - N INTEGER
7 C
8 IMPLICIT LOGICAL (L)
9 IMPLICIT DOUBLE PRECISION (A-H, O-Z)
10 END
11 QVAR PROC
12 C
13 C THESE FORMALLY NAMED VARIABLES ARE MAINTAINED AND UPDATED THROUGH-
14 C OUT THE VIRTUAL MASS PROGRAM. THEY ARE AVAILABLE TO BOTH PRINT NOW
15 C AND PRINT LATER OPTIONS (I. E., THEY ARE COPIED ON THE ROLL-BACK
16 C AND PRINT LATER FILES AS APPROPRIATE FOR 'LATER' PROCESSING
17 C
18 COMMON / CVAR / KVAR(1160)
19 C
20 DIMENSION VAR(300), IVAR(600), VAC(050)
21 EQUIVALENCE (KVAR, VAR, IVAR), (KVAR(601), VAC)
22 END
23 EVARI PROC
24 C
25 DIMENSION ISW(50), JK(100), JKPTS(2,12)
26 EQUIVALENCE (KVAR(701), ISW), (KVAR(751), JK), (KVAR(851), JKPTS)
27 C
28 EQUIVALENCE (ISW(01), IEVM ), (ISW(02), IEVN ),
29 1 (ISW(03), IEVCUR), (ISW(04), IEVFLG), (ISW(05), IEVSTE),
30 2 (ISW(06), IEVSTP), (ISW(07), IEVSTR),
31 3 (ISW(09), INCNTR), (ISW(10), IPLATM),
32 4 (ISW(12), IPLITM), (ISW(13), IPNOWM),
33 5 (ISW(15), ITRAT ), (ISW(16), JKN ), (ISW(17), KEPMAX),
34 6 (ISW(18), JKPTSN), (ISW(19), MVAR ), (ISW(20), JKSPEC),
35 7 (ISW(22), NMAX ),
36 8 (ISW(24), IRTIME)
37 EQUIVALENCE (ISW(30), KMERC ), (ISW(31), KVENUS),
38 1 (ISW(32), KEARTH), (ISW(33), KMARS ), (ISW(34), KJUP ),
39 2 (ISW(35), KSAT ), (ISW(36), KURAN ), (ISW(37), KNEP ),
40 3 (ISW(38), KPLUTO), (ISW(39), KSUN ), (ISW(40), KMOON ),
41 4 (ISW(41), KEMB ), (ISW(42), IDTRAJ), (ISW(45), IEVREP),
42 5 (ISW(47), IEVRUN)
43 DIMENSION IDTRAJ(3), IRTIME(3), IEVREP(2), IEVRUN(2)
44 END
45 EVARL PROC
46 C
47 DIMENSION LSW(50)
48 EQUIVALENCE (KVAR(901), LSW)
49 C
50 EQUIVALENCE (LSW(01), LOGOUT), (LSW(02), LIMPEV),
51 1 (LSW(03), LXECEV), (LSW(04), LSTDDK),
52 2 (LSW(06), LPRPHC), (LSW(07), LPRFAC),
53 3 (LSW(09), LPRVAR), (LSW(10), LPRBKP), (LSW(11), LPRNOW),
54 4 (LSW(12), LPRLAT), (LSW(13), LPRTEM), (LSW(14), LLDCOM),
55 5 (LSW(15), LPRCOM), (LSW(16), LRESKP), (LSW(17), LPREPH),
56 6 (LSW(18), LINTDN), (LSW(19), LINIT )

```

```

57      END
58      EVARIO PROC
59      C
60          DIMENSION IOU(25)
61          EQUIVALENCE (KVAR(951), IOU)
62      C
63      C      IOCDR      ON-LINE CARD READER
64      C      IOPRT      ON-LINE PRINTER
65      C      IOPCH      ON-LINE CARD PUNCH
66      C      IOPRL      PRINT LATER
67      C      IOROL      UNIT CONTAINING EVENTS STATES FOR 'ROLL-BACK' PURPOSES
68      C      IOEPH      EPHEMERIS DATA FILE
69      C      IOMES      ERROR AND PROGRAM CONTROL FLOW MESSAGES
70      C      IOSCR      SCRATCH FILE - AVAILABLE FOR ANY LOCAL USE
71      C
72          EQUIVALENCE          (IOU(1), IOCDR),      (IOU(2), IOPRT),
73      1      (IOU(3), IOPCH),      (IOU(4), IOEPH),      (IOU(5), IOMES),
74      2      (IOU(6), IOSCR),      (IOU(7), IOPRL),      (IOU(8), IOROL)
75      END
76      EVARS PROC
77      C
78          EQUIVALENCE          (VAR(1), TEPEJD),      (VAR(2), TEPEJS),
79      1      (VAR(3), TEPOJD),      (VAR(4), TEPOJS),
80      2      (VAR(6), TTRB ),      (VAR(7), TTRE ),
81      3      (VAR(9), TEVI ),      (VAR(10), TRCI ),      (VAR(11), TTRI ),
82      4      (VAR(12), CT2 ),
83      5      (VAR(15), DTAI ),      (VAR(16), DTAIDN),      (VAR(17), DTAIMN),
84      6      (VAR(18), DTAIUP),
85      7      (VAR(21), ERCAVM),
86      8      (VAR(24), GVMB ),      (VAR(25), GVME ),
87      9      (VAR(27), HRCAVM)
88          EQUIVALENCE
89      1
90      2      (VAR(35), QVMB ),      (VAR(36), QVME ),
91      3      (VAR(38), RRCAVM),      (VAR(39), RRCDVM),      (VAR(40), RSCAVM),
92      4      (VAR(41), RSCDVM),
93      5      (VAR(44), RTOPCM),
94      6      (VAR(47), RTOPRF),
95      7      (VAR(50), RUTOAU),
96      8      (VAR(53), TTO ),      (VAR(54), TUTODA),
97      9      (VAR(56), VUTOAD)
98          EQUIVALENCE (TEPI, TTRI), (TEPE, TTRE), (TEPB, TTRB),
99      1      (RTO, RTOPRF)
100     END
101     EVARV PROC
102     EQUIVALENCE
103     1
104     2
105     3
106     4      (VAR(92), ERCBVM),      (VAR(95), HRCBVM),
107     5      (VAR(101), RRCIVM),
108     6      (VAR(110), RRCBVM),      (VAR(113), VRCBVM),      (VAR(116), RRCEVM),
109     7      (VAR(119), VRCEVM),      (VAR(122), RSCBIN),      (VAR(125), VSCBIN),
110     8      (VAR(128), RSCEIN),      (VAR(131), VSCEIN),
111     9      (VAR(137), RSCDDK),      (VAR(140), VSCDDK)
112     EQUIVALENCE          (VAR(146), RSCBVM),      (VAR(149), VSCBVM),
113     1      (VAR(152), RSCEVM),      (VAR(155), VSCEVM),      (VAR(158), RVMBIN),

```

```

114      2 (VAR(161), VVMBIN), (VAR(164), RVMEIN), (VAR(167), VVMEIN)
115      C
116      DIMENSION RSCBIN(3), RSCEIN(3), RVMBIN(3), VSCBIN(3), VSCEIN(3),
117      1 VVMBIN(3), VVMEIN(3), RVMEIN(3),
118      2 RSCBVM(3), VSCBVM(3), RSCEVM(3), VSCEVM(3), RRCBVM(3),
119      3 VRCBVM(3), RRCIVM(3), HRCBVM(3), ERCBVM(3), RRCEVM(3),
120      4 VRCEVM(3), RSCDDK(3), VSCDDK(3)
121      END
122      QEPH PROC
123      C
124      C THESE LABELS GIVE THE INSTANTANEOUS EPHEMERIS OF THOSE
125      C BODIES HAVING SIGNIFICANT EFFECT ON THE SPACECRAFT
126      C
127      COMMON / CEPH / EPH(21,20)
128      C
129      EQUIVALENCE (EPH(01,1), ROOEIN), (EPH(04,1), VOOEIN),
130      1 (EPH(07,1), RSCE00), (EPH(10,1), VSCE00), (EPH(13,1), RSCA00),
131      2 (EPH(14,1), RSCD00), (EPH(15,1), GOO ), (EPH(16,1), GOOR ),
132      3 (EPH(17,1), DANE00), (EPH(18,1), DOBE00), (EPH(19,1), DLNE00),
133      4 (EPH(20,1), SPLE00), (EPH(21,1), ROOD ),
134      5 (EPH(17,1), RMNE00)
135      DIMENSION ROOEIN(3), VOOEIN(3), RSCE00(3), VSCE00(3),
136      1 RSCA00(1), RSCD00(1), GOO(1), ROOD(1), DANE00(1),
137      2 DOBE00(1), DLNE00(1), SPLE00(1), GOOR(1),
138      3 RMNE00(3)
139      C
140      C EPHEMERIS LAYOUT
141      C
142      C EACH GRAVITATING POINT I (SEE READE: I = 1, ..., 11 FOR SOLAR SYS)
143      C
144      C EPH(01,I) ROOEIN INERTIAL POSITION
145      C EPH(04,I) VOOEIN INERTIAL VELOCITY
146      C EPH(07,I) RSCE00 SPACE CRAFT POSITION RELATIVE TO I
147      C EPH(10,I) VSCE00 SPACE CRAFT VELOCITY RELATIVE TO I
148      C EPH(13,I) RSCA00 PREVIOUS DISTANCE BETWEEN SPACE CRAFT AND I
149      C EPH(14,I) RSCD00 CURRENT DISTANCE BETWEEN SPACE CRAFT AND I
150      C EPH(15,I) GOO GRAVITATIONAL CONSTANT TIMES MASS OF I
151      C EPH(16,I) GOOR MASS RATIO OF I TO ALL THE PLANETS
152      C
153      C EACH PLANET
154      C
155      C EPH(17,I) DANE00 RIGHT ASCENSION
156      C EPH(18,I) DOBE00 OBLIQUITY
157      C EPH(19,I) DLNE00 LONGITUDE OF NODE AT REFERENCE TIME
158      C EPH(20,I) SPLE00 ANGULAR ROTATION RATE (SIDEREAL)
159      C EPH(21,I) ROOD RADIUS OF (ASSUMED SPHERICAL) I
160      C
161      C EACH MASS POINT FROM PLANET MASS POINT REPRESENTATION
162      C
163      C EPH(17,I) RMNE00 PLANETOGRAPHIC POSITION
164      C
165      JKLOC(I) = 1 + JKSPEC * (I - 1)
166      C
167      C NOTE: LIST JK OF JKN ENTRIES GIVEN SUBSCRIPTS I OF GRAVITATING
168      C POINTS CONTRIBUTING TO CURRENT COMPUTATIONS
169      C
170      END

```

```

171  EEPH  PROC
172  END
173  QMON  PROC
174      COMMON / CMON / MONTH(12)
175      DATA MONTH / 'JAN@01FEB@02MAR@03APR@04MAY@05JUN@06',
176      1          'JUL@07AUG@08SEP@09OCT@10NOV@11DEC@12'
177  END
178  QFAC  PROC
179  C
180  C      THESE NUMERICAL FACTORS AND CONSTANTS ARE FOR SPACE SAVINGS AND
181  C      FOR ASSURING MAXIMUM PRECISION IN DOUBLE PRECISION REPRESENTATIONS
182  C
183      COMMON / CFAC / FAC(58)
184  C
185      EQUIVALENCE          (FAC(1), F0),          (FAC(2), F9TNTH),
186      1 (FAC(3), F1),      (FAC(4), F2),          (FAC(5), F3),
187      2 (FAC(6), F4),      (FAC(7), F5),          (FAC(8), F6),
188      3 (FAC(9), F7),      (FAC(10), F8),         (FAC(11), F9),
189      4 (FAC(12), F10),     (FAC(13), F11),        (FAC(14), F12),
190      5 (FAC(15), F13),     (FAC(16), F14),        (FAC(17), F15),
191      6 (FAC(18), F24 ),    (FAC(19), F60 ),    (FAC(20), F1440 ),
192      7 (FAC(21), F3600 ),  (FAC(22), F86400), (FAC(23), FINITY),
193      8 (FAC(24), FPT15 ),  (FAC(25), FPT4 ),  (FAC(26), F1PT5 ),
194      9 (FAC(27), F2PT5 ),  (FAC(28), F20 ),   (FAC(29), F30 ),
195      EQUIVALENCE          (FAC(30), RTOD ),      (FAC(31), DTOR ),
196      1 (FAC(32), F43200),  (FAC(33), AUTOKM), (FAC(34), RETOKM),
197      2 (FAC(35), FPT5 ),   (FAC(36), F001 ),  (FAC(37), F00001),
198      3 (FAC(38), F1PT1 ),  (FAC(39), FTT0 ),  (FAC(40), FRT0 ),
199      4 (FAC(41), FINTES),  (FAC(42), FTTOKM)
200  END
201  EFAC  PROC
202  END
203  QIEV  PROC
204  C
205  C      TABLE OF 'ACTIVE' EVENTS --- SEE EVNACC FOR LAYOUT
206  C
207      DIMENSION IEV(8,20), EVN(4,20)
208      EQUIVALENCE (KVAR(1001), IEV, EVN)
209  END
210  QPHC  PROC
211  C
212  C      THESE PHYSICAL CONSTANTS DO NOT CHANGE DURING THE COMPUTER RUN
213  C
214      COMMON / CPHC / PHC(50)
215  C
216      EQUIVALENCE          (PHC(01), GS ),      (PHC(02), GMSUN ),
217      1 (PHC(03), PIMASS)
218      DIMENSION PIMASS(12)
219  END
220  EPHC  PROC
221  END
222  QREAD  PROC
223  C
224  C      READ JPL EPHEMERIS AT THE JULIAN EPHEMERIS DATE          READ0060
225  C      GIVEN BY (JED+TSEC/86400.D0)                            READ0070
226  C                                                                READ0080
227  C**  ITEMS COMMUNICATED THROUGH THE CALLING SEQUENCE **      READ0090

```



```

285      C          2 VENUS      6 SAT      10 SUN          READ0670
286      C          3 EARTH     7 URANUS   11 MOON         READ0680
287      C          4 MARS      8 NEP      12 EARTH-MN-BARYCENTER READ0690
288      C                                13 NUTATION     READ0700
289      C * COMMON BLOCK CETBL3 *
290      C      TAB3 829 DOUBLE PREC. WORD BUFFER TO ACCOMODATE J.D. AND EPHEMERIS. READ0720
291      C      NUTAT 204 SINGLE PREC. WORD BUFFER TO ACCOMODATE NUTATION DATA. READ0730
292      C      CKSUM 1 S.P. WORD FOR CHECKSUM. READ0740
293      C** THE FOLLOWING ITEMS ARE OUTPUT THROUGH COMMON ** READ0750
294      C * COMMON BLOCK CETBL4 *
295      C      TABOUT( , ) PLANETARY AND LUNAR OUTPUT, SCALED AND
296      C      TRANSLATED WITH RESPECT TO CENTER. READ0770
297      C      TABOUT (I,J) CONTAINS OUTPUT FOR
298      C      BODY NO. J. (1 .LE. J .LE. 12) READ0790
299      C      THE INDEX I IDENTIFIES COMPONENTS AS FOLLOWS.. READ0800
300      C      1=X      2=Y      3=Z
301      C      4=XDOT   5=YDOT   6=ZDOT
302      C      NUT( )      NUTATION OUTPUT
303      C      NUT(1)=DELTA LONGITUDE
304      C      NUT(2)=DELTA OBLIQUITY
305      C      NUT(3)=TIME DERIVATIVE OF NUT(1)
306      C      NUT(4)=TIME DERIVATIVE OF NUT(2)
307      C * COMMON BLOCK CETBL5 *
308      C      BIVECT( , ) WORKING ARRAY. CONTENTS ARE INTERPOLATED
309      C      AND SCALED BUT NOT TRANSLATED. 1ST INDEX RUNS
310      C      OVER X,Y,Z,XDOT,YDOT,ZDOT AS IN TABOUT
311      C      BUT 2ND INDEX IS DIFFERENT AS FOLLOWS..
312      C      BODIES 1 THRU 9 ARE HELIOCENTRIC.
313      C      1 MERC      5 JUP      9 PLUTO
314      C      2 VENUS     6 SAT      10 MOON REL TO EARTH
315      C      3 EARTHMN  7 URANUS   11 EARTHMN REL TO EARTH
316      C      4 MARS      8 NEP      12 EARTHMN REL TO MOON
317      C                                13 SEE 4092+
318      C
319      C      THE COMMON BLOCK 'CETBL9' IS FOR COMMUNICATION
320      C      BETWEEN RDEP2 AND GETR2.
321      C      COMMON / CETBL1 / IETBL1(8)
322      C      1 / CETBL2 / IETBL2(15)
323      C      2 / CETBL3 / IETBL3(1836)
324      C      3 / CETBL4 / IETBL4(148)
325      C      4 / CETBL5 / IETBL5(156)
326      C      5 / CETBL9 / IETBL9(7)
327      C      INTEGER IREQ(13)
328      C      REAL NUTAT(204), NUT(4), CKSUM
329      C      DOUBLE PRECISION AU, JD1, JDIF, EMRAT, RE, TPD, TDAY,
330      C      1 TAB3(829), TABOUT(6,12), BIVECT(6,13)
331      C      EQUIVALENCE (IETBL1(1), AU), (IETBL1(3), RE),
332      C      1 (IETBL1(5), TPD), (IETBL1(7), EMRAT),
333      C      2 (IETBL2(1), ICW), (IETBL2(2), ICENT), (IETBL2(3), IREQ),
334      C      3 (IETBL3(1), TAB3), (IETBL3(1659), NUTAT), (IETBL3(1863), CKSUM),
335      C      4 (IETBL4(1), TABOUT), (IETBL4(145), NUT), (IETBL5(1), BIVECT),
336      C      5 (IETBL9(1), JD1), (IETBL9(3), TDAY), (IETBL9(5), JDIF),
337      C      6 (IETBL9(7), IERR1)
338      C
339      C      END
340      C      EREADE PROC
341      C      END

```

```

342 QTEM PROC
343 C
344 C THESE TEMPORARY VARIABLES ARE FOR CONVENIENCE AND SPACE ECONOMY.
345 C THEY ARE NOT NECESSARILY MAINTAINED FROM ONE SUBROUTINE TO ANOTHER
346 C
347 C COMMON / CTEM / TEM(100)
348 C
349 EQUIVALENCE (TEM(1), T1), (TEM(2), T2),
350 1 (TEM(3), T3), (TEM(4), T4), (TEM(5), T5),
351 2 (TEM(6), T6), (TEM(7), T7), (TEM(8), T8),
352 3 (TEM(9), T9), (TEM(10), T10), (TEM(11), T11),
353 4 (TEM(12), S1), (TEM(13), S2), (TEM(14), S3),
354 5 (TEM(15), S4), (TEM(16), S5), (TEM(17), S6),
355 6 (TEM(18), S7), (TEM(19), S8), (TEM(20), S9),
356 7 (TEM(21), X1), (TEM(22), X2), (TEM(23), X3),
357 8 (TEM(24), X4), (TEM(25), X5), (TEM(26), X6),
358 9 (TEM(27), X7), (TEM(28), X8), (TEM(29), X9)
359 EQUIVALENCE (TEM(30), TV1), (TEM(33), TV2),
360 1 (TEM(36), TV3), (TEM(39), TV4), (TEM(42), TV5),
361 2 (TEM(45), TV6), (TEM(48), TV7), (TEM(51), TV8),
362 3 (TEM(54), TV9), (TEM(57), TV10), (TEM(60), TV11),
363 4 (TEM(63), SV1), (TEM(66), SV2), (TEM(69), SV3),
364 5 (TEM(72), SV4), (TEM(75), SV5), (TEM(78), SV6),
365 6 (TEM(81), XV1), (TEM(84), XV2), (TEM(87), XV3),
366 7 (TEM(90), XV4), (TEM(93), XV5), (TEM(96), XV6)
367 DIMENSION TV1(3), TV2(3), TV3(3), TV4(3), TV5(3), TV6(3),
368 1 TV7(3), TV8(3), TV9(3), TV10(3), TV11(3),
369 2 SV1(3), SV2(3), SV3(3), SV4(3), SV5(3), SV6(3),
370 3 XV1(3), XV2(3), XV3(3), XV4(3), XV5(3), XV6(3)
371 END
372 ETEM PROC
373 END

```

Program: CONIC

Purpose: Compute the vector orbital elements of the reference conic section, given the position and velocity vectors relative to the virtual mass.

Calls: CROSS, VALUE

Called by: STEPIN

Erros and Diagnostics: None

Language: Fortran subroutine

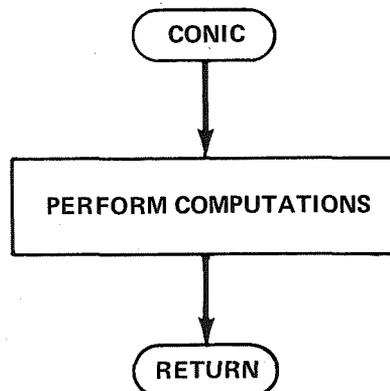
Computations: (See Table 2 for definition of symbols):

$$\text{HRCBVM} = \text{RRCBVM} \times \text{VRCBVM}$$

$$\text{HRCAVM} = |\text{HRCBVM}|$$

$$\text{ERCBVM} = -\frac{\text{RRCBVM}}{\text{RRCAVM}} - \frac{\text{HRCBVM} \times \text{VRCBVM}}{\text{GVMB}}$$

$$\text{ERCAVM} = |\text{ERCBVM}|$$



VMASS\*RFJK.CONIC

```

1      C
2      C  PURPOSE
3      C
4      C  COMPUTES THE VECTOR ORBITAL ELEMENTS OF THE REFERENCE CONIC
5      C  SECTION, GIVEN THE POSITION AND VELOCITY VECTORS RELATIVE TO THE
6      C  VIRTUAL MASS.  OUTPUTS THESE VECTOR ELEMENTS IN THE VIRTUAL MASS
7      C  CENTERED INERTIALLY ORIENTED COORDINATE FRAME.
8      C
9      C  INPUT
10     C
11     C  GVMB  PRODUCT OF UNIVERSAL GRAVITATION CONSTANT AND THE VIRTUAL
12     C  MASS MAGNITUDE
13     C  RRCBVM  THREE COMPONENTS OF THE REFERENCE CONIC
14     C  POSITION RELATIVE TO THE VIRTUAL MASS
15     C  VRCBVM  REF CONIC VELOCITY RELATIVE TO THE VIRTUAL MASS
16     C  RRCAVM  REF CONIC DISTANCE FROM THE VIRTUAL MASS (MAGNITUDE OF
17     C  RRCBVM)
18     C
19     C  OUTPUT
20     C
21     C  HRCBVM  ANGULAR MOMENTUM (AREAL RATE) VECTOR OF REFERENCE CONIC
22     C  ORBIT RELATIVE TO THE VIRTUAL MASS
23     C  HRCAVM  MAGNITUDE OF ANGULAR MOMENTUM VECTOR (HRCBVM)
24     C  ERCBVM  ECCENTRICITY VECTOR OF CONIC ORBIT RELATIVE TO THE V MASS
25     C  ERCAVM  MAGNITUDE OF ERCBVM
26     C
27     C  SUBROUTINE CONIC
28     C
29     C  INCLUDE QTYPE
30     C  INCLUDE QVAR
31     C  INCLUDE QTEM
32     C  INCLUDE EVARS
33     C  INCLUDE EVARV
34     C  INCLUDE ETEM
35     C
36     C  CALL CROSS (RRCBVM, VRCBVM, HRCBVM)
37     C  HRCAVM = VALUE (HRCBVM)
38     C  CALL CROSS (HRCBVM, VRCBVM, TV1)
39     C  DO 10 I = 1, 3
40     C 10  ERCBVM(I) = -RRCBVM(I) / RRCAVM - TV1(I) / GVMB
41     C  ERCAVM = VALUE (ERCBVM)
42     C
43     C  RETURN
44     C  END

```

Program: CONICT

Purpose: Compute the reference conic time of flight from initial position vector and incremental vector to desired final position on orbit.

Called by: KEPLER

Calls: CROSS, DOT

Errors and Diagnostics: The CONICT result is based on a time series calculation. If the series is found to be divergent, a subroutine log out is initiated. If found to require an excessive number of terms, a diagnostic printout is initiated.

Language: Fortran subroutine

Computations:

$$V_3 = \frac{RRCIVM}{HRCAVM} \times \frac{HRCBVM}{HRCAVM}$$

$$T_2 = 1 - ERCAVM^2$$

$$T_3 = T_2 \cdot GVMB \cdot HRCAVM^{-2}$$

$$T_4 = V_3 \cdot (ERCBVM + T_3 \cdot RRCBVM)$$

$$T_5 = |T_2| (T_4 \cdot GVMB \div HRCAVM)^2$$

$$T_{6,0} = GVMB \cdot T_4^3$$

$$T_{7,0} = 1$$

$$T_{9,0} = T_2$$

BELLCOMM, INC.

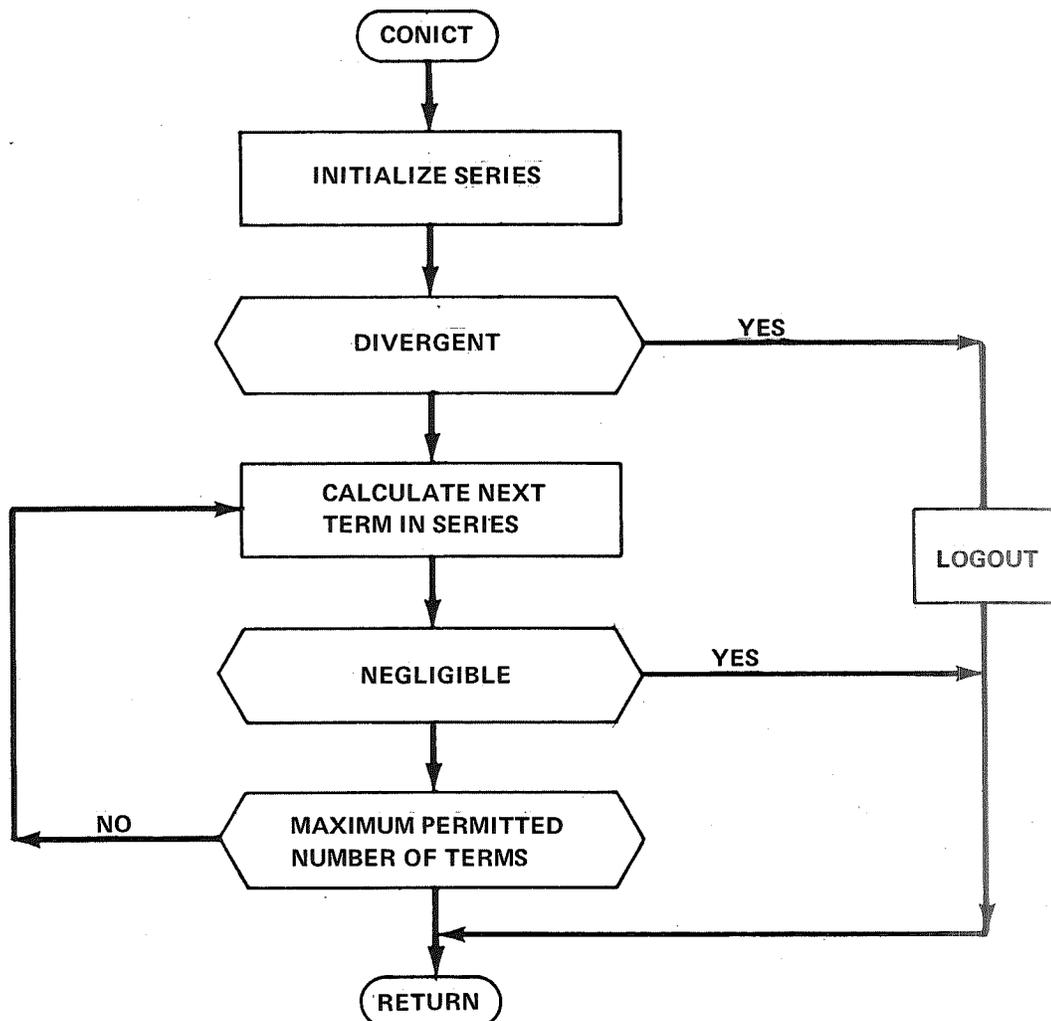
$$TRCI_0 = V_3 \circ RRCBVM$$

$$T_{7,n} = T_{7,n-1} - T_{7,n-1} \div 2n$$

$$T_8 = T_6 \cdot T_{7,n} \div (2n+1)$$

$$T_{9,n} = T_{9,n-1} \cdot T_2$$

$$TRCI_n = TRCI_{n-1} + \text{SIGN}(T_8, T_9)$$



## VMASS\*RFJK.CONICT

```

1      C
2      C  PURPOSE
3      C
4      C  THIS PROGRAM COMPUTES THE REFERENCE CONIC TIME OF FLIGHT FROM
5      C  INITIAL POSITION VECTOR AND INCREMENTAL VECTOR TO DESIRED FINAL
6      C  POSITION ON ORBIT. THE REFERENCE CONIC ORBIT IS SPECIFIED BY THE
7      C  VECTOR ORBITAL ELEMENTS.
8      C
9      C  INPUTS
10     C
11     C  HRCBVM  REFERENCE CONIC ANGULAR MOMENTUM (AREAL RATE)
12     C  HRCAVM  MAGNITUDE OF THE ANGULAR MOMENTUM
13     C  ERCBVM  REFERENCE CONIC ECCENTRICITY VECTOR
14     C  ERCAVM  MAGNITUDE OF ECCENTRICITY
15     C  RRCBVM  REFERENCE CONIC INITIAL POSITION VECTOR
16     C  RRCIVM  REFERENCE CONIC POSITION VECTOR INCREMENT
17     C  NMAX    MAXIMUM NUMBER OF TERMS IN SERIES EXPANSION FOR EVAL TIME
18     C  GVMB    PRODUCT OF UNIVERSAL GRAVITATION CONST AND VIRT MASS MAG
19     C  RTO     TOLERANCE DEFINING ACCURACY OF COMPUTED POS INCREMENT
20     C
21     C  OUTPUT
22     C
23     C  TRCI    REFERENCE CONIC INCREMENTAL TIME
24     C
25     C  SUBROUTINE CONICT
26     C
27     C  INCLUDE QTYPE
28     C  INCLUDE QVAR
29     C  INCLUDE QTEM
30     C  INCLUDE QFAC
31     C  INCLUDE EVARS
32     C  INCLUDE EVARV
33     C  INCLUDE EVARIO
34     C  INCLUDE EVARI
35     C  INCLUDE EVARL
36     C
37     C  DO 1 I = 1, 3
38         TV1(I) = RRCIVM(I) / HRCAVM
39     1   TV2(I) = HRCBVM(I) / HRCAVM
40     CALL CROSS (TV1, TV2, TV3)
41     TRCI = DOT (TV3, RRCBVM)
42     C
43     C  T2 = F1 - ERCAVM ** 2
44     C  T3 = GVMB * T2 / HRCAVM ** 2
45     C  DO 29 I = 1, 3
46     29  TV5(I) = ERCBVM(I) + RRCBVM(I) * T3
47     T4 = DOT (TV5, TV3)
48     T5 = ABS (T2) * (GVMB * T4 / HRCAVM) ** 2
49     C
50     C  CHECK FOR DIVERGENT TIME SERIES
51     C
52     C  IF (T5 .LE. F1) GO TO 5
53     C  WRITE (IOMES, 2)
54     2   FORMAT (' CONICT ** TIME SERIES DIVERGENT ** ARG .GT. 1')
55     C  LOGOUT = .TRUE.
56     C  RETURN

```

```
57      5      T6 = GVMB * T4 ** 3
58          T7 = F1
59          T9 = T2
60          DO 20 N = 1, NMAX
61              T7 = T7 - T7 / (N + N)
62              T8 = T6 * T7 / (N + N + 1)
63              IF (ABS (T8) .LE. TT0) RETURN
64              IF (ABS (T8) .LE. FINTE5 * TRCI) RETURN
65              T9 = T9 * T2
66              TRCI = TRCI + SIGN (T8, T9)
67              T6 = T6 * T5
68          C
69          C      TIME SERIES REQUIRES MORE THAN NMAX TERMS FOR CONVERGENCE
70          C
71          WRITE (IOMES, 30) NMAX
72      30      FORMAT (' CONICT *** TIME SERIES TERMS EXCEEDING', I6)
73          RETURN
74          END
```

Program: CROSS (a, b, c)

Purpose: Calculate vector cross product.

Called by: CONIC, CONICT, KEPLER

Calls: None

Errors and Diagnostics: None

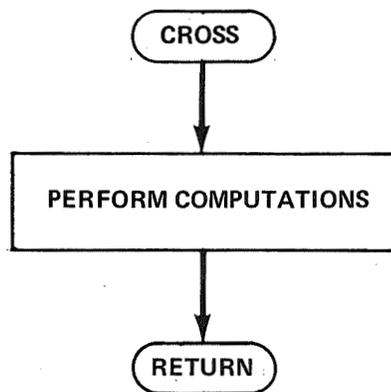
Language: Fortran subroutine

Computations:

$$C_1 = a_2 b_3 - a_3 b_2$$

$$C_2 = a_3 b_1 - a_1 b_3$$

$$C_3 = a_1 b_2 - a_2 b_1$$



VMASS\*RFJK.CROSS

```
1      C
2      C  PURPOSE
3      C
4      C    CALCULATE VECTOR C THE CROSS PRODUCT OF VECTORS A AND B (3 COMP)
5      C
6      C    SUBROUTINE CROSS (A, B, C)
7      C
8      C    INCLUDE QTYPE
9      C    DIMENSION A(3), B(3), C(3)
10     C
11     C    DO 10 I = 1, 3
12     C        J = 1 + MOD (I, 3)
13     C        K = 1 + MOD (I + 1, 3)
14     C    10  C(I) = A(J) * B(K) - A(K) * B(J)
15     C
16     C    RETURN
17     C    END
```

Program: DDREAD

Purpose: Read input data deck via Fortran Name List

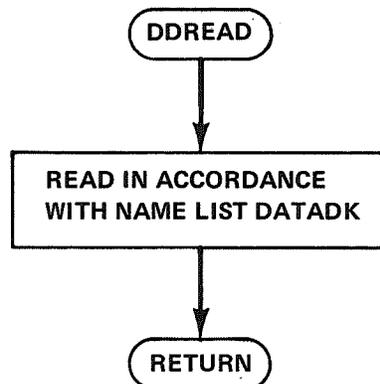
Called by: SETUP

Calls: None

Errors and Diagnostics: Log out initiation from the data deck is acknowledged.

Language: Fortran subroutine

Computations: None



VMAS\*RFJK.DDREAD

```
1      C
2      C  PURPOSE
3      C
4      C    READ DATA DECK - SOME LOCATIONS BY NAME - OTHERS BY COMMON BLOCK
5      C
6      C    SUBROUTINE DDREAD
7      C
8      C    INCLUDE QTYPE
9      C    INCLUDE QVAR
10     C    INCLUDE QEPH
11     C    INCLUDE QPHC
12     C    INCLUDE QFAC
13     C    INCLUDE QREAD
14     C    INCLUDE EVARI
15     C    INCLUDE EVARIO
16     C    INCLUDE EVARL
17     C    INCLUDE EVARS
18     C    INCLUDE EVARV
19     C
20     C
21     C    NAME LIST / DATADK / EPH, FAC, IOU, ISW, LSW, PHC, VAC, VAR,
22     1  DTAI,  DTAIDN, DTAIMN, DTAIUP, IDTRAJ, IEVREP, IEVRUN, IEVSTP,
23     2  IEVSTR, IPLATM, IPLOTM, IPNOWM, LPRLAT, LPRNOW, LSTDDK,
24     3  RTOPRF, TEPOJD, TEPOJS, ITEPOG, AU, RE, TPD, EMRAT, ICENTR,
25     4  IOEPH, IOMES, IOPRL, IOROL, LOGOUT, LINIT,
26     5  RSCDDK, VSCDDK, LPRNOW, JK, JKN, RSCEIN, VSCEIN,
27     6  IOPCH, IOPRT, IOSCR,
28     C
29     C    READ (IOCDR, DATADK)
30     C    IF (LOGOUT) WRITE (IOMES, 10)
31     10  FORMAT (' DDREAD *** DATA DECK LOG OUT')
32     C
33     C    RETURN
34     C    END
```

Program: x = DEVAR (m, n)

Purpose: Obtain difference between two variables for which are given subscripts relative to some common symbol.

Called by: EVNACC, EVNDT, INDEX

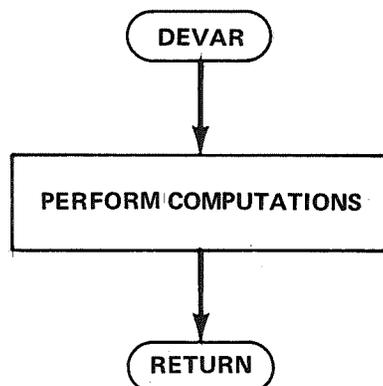
Calls: None

Errors and Diagnostics: None

Language: Fortran function

Computations:

$$x = V_m - V_n$$



## VMASS SUBROUTINE DEVAR

VMASS\*RFJK.DEVAR

```
1      C
2      C  PURPOSE
3      C
4      C  OBTAIN DIFFERENCE BETWEEN TWO DOUBLE PRECISION VAR LOCATIONS
5      C  FOR WHICH SINGLE PRECISION INDICES HAVE BEEN COMPUTED
6      C
7      C  FUNCTION DEVAR (M, N)
8      C
9      C  WRITE
10     C
11     C      X = DEVAR (M, N)
12     C
13     C  RATHER THAN
14     C
15     C      X = VAR(M) - VAR(N)
16     C
17     C  WHICH AT OBJECT CODE LEVEL BECOMES
18     C
19     C      X = VAR(2*M) - VAR(2*N)
20     C
21     C  INCLUDE QTYPE
22     C  INCLUDE QVAR
23     C  INCLUDE QTEM
24     C  EQUIVALENCE (T1, IT1), (T3, IT2)
25     C  DIMENSION IT1(2), IT2(2)
26     C  IT1(1) = KVAR(M)
27     C  IT1(2) = KVAR(M+1)
28     C  IT2(1) = KVAR(N)
29     C  IT2(2) = KVAR(N+1)
30     C  DEVAR = T1 - T3
31     C  RETURN
32     C  END
```

Program:  $x = \text{DOT} (\bar{y}, \bar{z})$

Purpose: Calculate vector dot product

Called by: CONICT, KEPLER, PREPAR, STEPIN, VMXACT

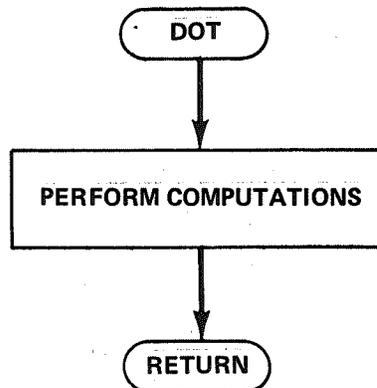
Calls: None

Errors and Diagnostics: None

Language: Fortran function

Computations:

$$x = y_1 z_1 + y_2 z_2 + y_3 z_3$$



VMASS\*RFJK.DOT

```
1      C
2      C  PURPOSE
3      C
4      C    CALCULATE DOT PRODUCT OF TWO 3 COMPONENT VECTORS
5      C
6      C    FUNCTION DOT (A, B)
7      C
8      C    INCLUDE QTYPE
9      C    DIMENSION A(3), B(3)
10     C
11     C    DOT = A(1) * B(1) + A(2) * B(2) + A(3) * B(3)
12     C
13     C    RETURN
14     C    END
```

Program: DTJTOG ( $J_d$ ,  $J_t$ , G)

DTGTOS (G,  $J_d$ ,  $J_t$ )

Purpose: Convert date-time group from Julian to Gregorian form and vice versa.

Called by: PRTIME

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: (See for reference Fliegel and Van Flanders, Communications of the ACM, October 1968)

$$G_1 = 100 (n - 49) + i + j/11$$

$$G_2 = j + 2 - 12 (j/11)$$

$$G_3 = m - 2447 j/80$$

where

$$i = 4000 (m_1 + 1) / 1461001$$

$$j = 80 m / 2447$$

$$m = m_1 + 31 - 1461 i / 4$$

$$m_1 = m_2 - (146097 n + 3) / 4$$

$$m_2 = J_d + 68569$$

$$n = 4 m_2 / 146097$$

and

$$G_4 = (3200 + J_t) / 3600$$

$$t = J_t - G_4 \cdot 3600$$

$$G_5 = (43200 + t) / 60$$

$$G_6 = 43200 + t - G_5 \cdot 60$$

On the reverse calculation for Gregorian to Julian

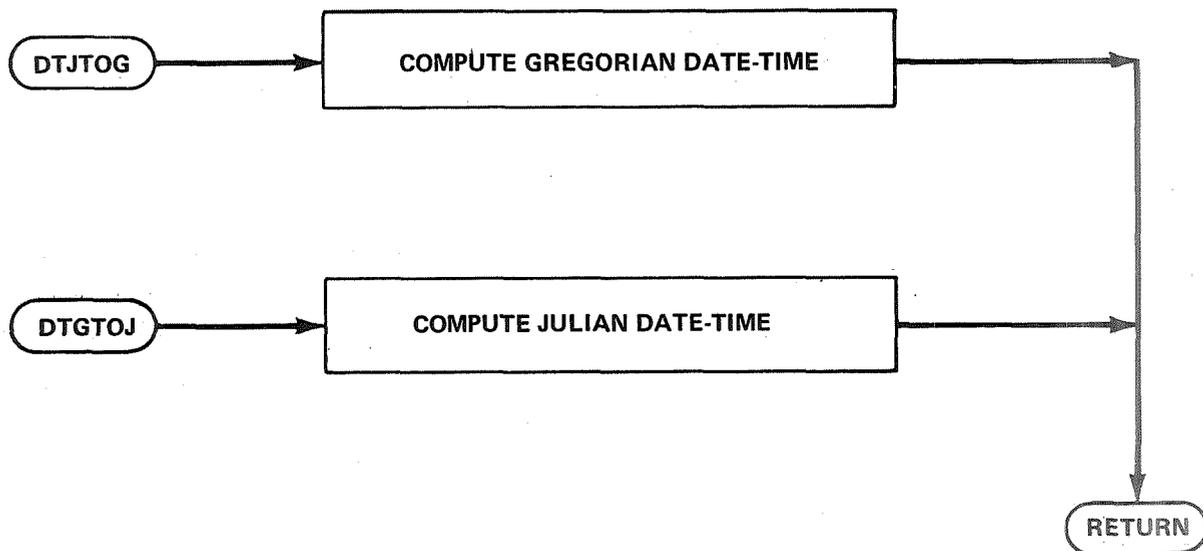
$$\begin{aligned} J_d &= .5 + k - 32075 \\ &\quad + 1561 (i + 4800 + (j - 14) / 12) / 4 \\ &\quad + 367 (j - 2 - (j - 14) / 12) / 12 \\ &\quad - 4 ((i + 4900 + (j - 14) / 12) / 100) / 4 \\ J_t &= 3600 G_4 + 60 G_5 + G_6 \end{aligned}$$

where

$$i = G_1$$

$$j = G_2$$

$$k = G_3$$



VMASS\*RFJK.DTJTOG

```

1      C      DTJTOG   R JESSUP - BELLCOMM - 21 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      CONVERT DATE - TIME FROM JULIAN TO GREGORY (DTJTOG) OR
6      C      VICE VERSA (DTGTOJ)
7      C
8      C      REFERENCE
9      C
10     C      FLIEGEL AND VAN FLANDERS COMMUNICATIONS ACM OCTOBER 68
11     C
12     C      CALLING SEQUENCE
13     C
14     C      SUBROUTINE DTJTOG (DJ, TJ, IG)
15     C
16     C      INPUTS
17     C
18     C      DJ          JULIAN DATE
19     C      TJ          TIME IN SECONDS PAST JULIAN DATE
20     C
21     C      OUTPUT
22     C
23     C      IG          GREGORIAN DAY, MONTH, YEAR, HOUR, MINUTE, SECOND
24     C      (24 HOUR CLOCK)
25     C
26     C      INCLUDE QTYPE
27     C      INCLUDE QFAC
28     C      INCLUDE QTEM
29     C
30     C      INTEGER IG(6)
31     C
32     C      NORMALIZE DJ + TJ TO READ D1.5 + T1 (T1 POSITIVE FRACTION OF DAY)
33     C
34     C      D1 = IFIX (FPT5 + DJ + TJ / F86400)
35     C      T1 = TJ + F86400 * (DJ - D1)
36     C
37     C      COMPUTE GREGORY DATE
38     C
39     C      70  M = D1 + 68569
40     C      N = 4 * M / 146097
41     C      M = M - (146097 * N + 3) / 4
42     C      I = 4000 * (M + 1) / 1461001
43     C      M = M - 1461 * I / 4 + 31
44     C      J = 80 * M / 2447
45     C      K = M - 2447 * J / 80
46     C      M = J / 11
47     C      J = J + 2 - 12 * M
48     C      I = 100 * (N - 49) + I + M
49     C      IG(1) = I
50     C      IG(2) = J
51     C      IG(3) = K
52     C      IG(4) = (F43200 + T1) / F3600
53     C      T1 = T1 - IG(4) * F3600
54     C      IG(5) = (F43200 + T1) / F60
55     C      T1 = T1 - IG(5) * F60
56     C      IG(6) = (F43200 + T1)

```

```
57          RETURN
58          C
59          C
60          ENTRY DTGTOJ (IG1, DJ1, TJ1)
61          C
62          C
63          C      COMPUTED JULIAN DATE GIVEN AS DJ.5 + TJ
64          C
65          I = IG1(1)
66          J = IG1(2)
67          K = IG1(3)
68          DJ1 = 0.5 + FLOAT (K - 32075
69          1    + 1461 * (I + 4800 + (J - 14) / 12) / 4
70          2    + 367 * (J - 2 - (J - 14) / 12 * 12) / 12
71          3    - 3 * ((I + 4900 + (J - 14) / 12) / 100) / 4)
72          TJ1 = IG1(4) * F3600 + IG1(5) * F60 + IG1(6)
73          RETURN
74          END
```

Program: EPHEM (element EPHEM)

Purpose: Determine positions and velocities of gravitating bodies

Called by: STEPIN

Errors and Diagnostics: Error returns from the subroutine READE are indicated. A log out is initiated.

Language: Fortran subroutine

Computations: READE is called to obtain sun-centered planetary state vectors. The following type transformation is then made to obtain inertial coordinates.

$$ROEIN_k = TABOUT_{i,j} - V_1$$

$$VOEIN_k = TABOUT_{i+3,j} - V_2$$

where

$TABOUT_{i,j}$  = position vector, jth planet relative to sun

$TABOUT_{i+3,j}$  = velocity vector, jth planet relative to sun

and  $V_i$  is sun vector relative to the inertial frame of reference and is given by

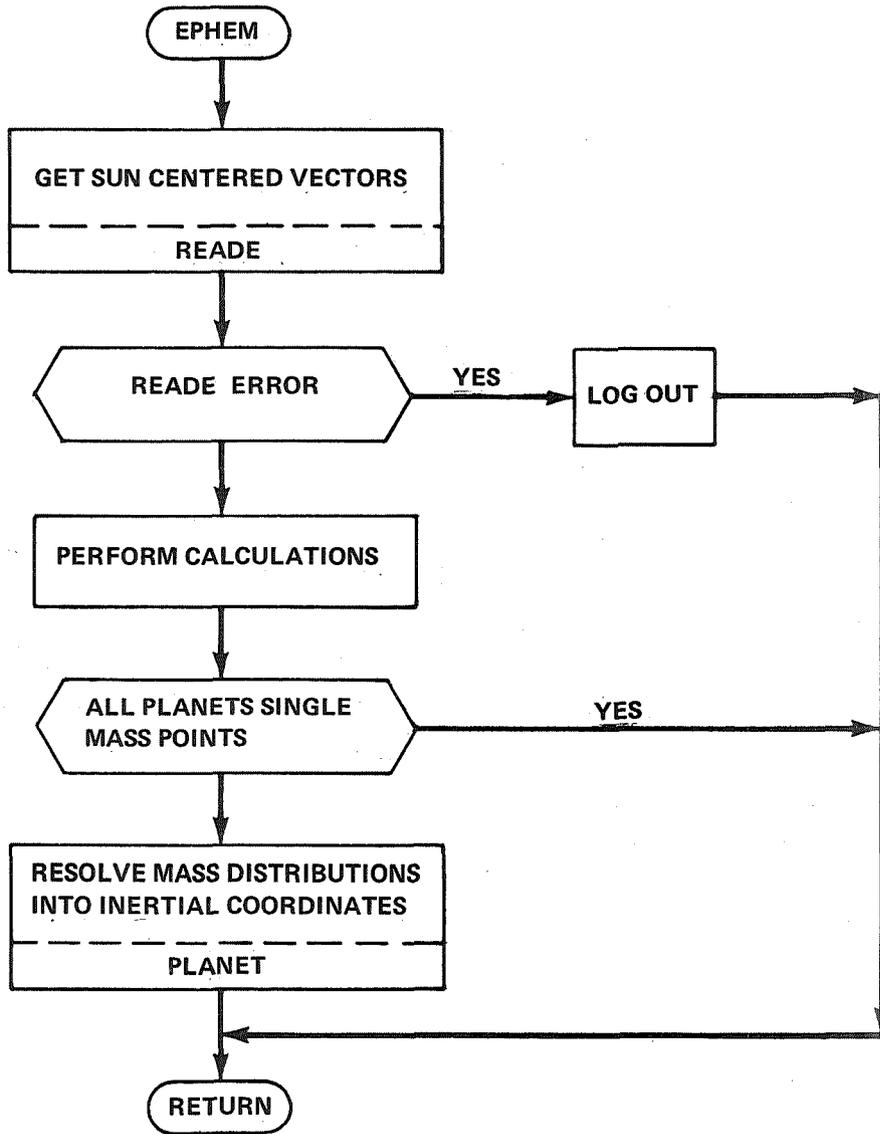
$$V_1 = \sum_j GOOR_k TABOUT_{i,j}$$

$$V_2 = \sum_j GOOR_k TABOUT_{i+3,j}$$

where

$GOOR_k$  = ratio of mass of planet j to sum of masses of solar system.

k = JKLOC (j)



VMASS\*RFJK\*EPHEM

```

1      C      EPHEM      R JESSUP - BELLCOMM - 09 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      DETERMINES FROM JPL EPHEMERIS FILES THE POSITIONS AND VELOCITIES
6      C      OF THE GRAVITATING BODIES
7      C
8      C      INPUTS
9      C
10     C      TTRE      ELAPSED TRAJECTORY TIME FROM FIRST EVENT
11     C      TEPOJD    JULIAN EPH DATE AT FIRST EVENT
12     C      TEPOJS    JULIAN EPH TIME AT FIRST EVENT
13     C      TEPEJD    CURRENT JULIAN EPHEMERIS DATE (DAYS)
14     C      TEPEJS    CURRENT JULIAN EPHEMERIS TIME (SECONDS)
15     C      TUTODA    MULTIPLIER FOR CONVERTING TRAJECTORY TIME TO DAYS
16     C
17     C      OUTPUTS
18     C
19     C      JK        LIST OF CURRENT GRAVITATING MASS POINTS
20     C      JKN        NUMBER OF ENTRIES IN JK
21     C      RJKEIN    CURRENT INERTIAL POS VECTOR OF MASS PT JK
22     C      VJKEIN    CURRENT INERTIAL VEL VECTOR OF MASS PT JK
23     C      GJKR      RATIO OF MASS OF GRAV PT JK TO MASS OF ALL THE GRAV PTS
24     C
25     C
26     C      SUBROUTINE EPHEM
27     C
28     C      INCLUDE QTYPE
29     C      INCLUDE QVAR
30     C      INCLUDE QPHC
31     C      INCLUDE QTEM
32     C      INCLUDE QEPH
33     C      INCLUDE QFAC
34     C      INCLUDE QREADE
35     C      INCLUDE EVARIO
36     C      INCLUDE EVARS
37     C      INCLUDE EVARI
38     C
39     C      DIMENSION      IEPER1(2), IEPER2(4,5)
40     C      DATA IEPER1 / ' READE *** ' /
41     C      1      IEPER2 / 'DATE TOO EARLY FOR EPH ' ,
42     C      2      'DATE TOO LATE FOR EPH ' ,
43     C      3      'IMPROPER OUTPUT REQUEST' ,
44     C      4      'INVALID CENTRAL BODY ' ,
45     C      5      'ICW OUT OF RANGE ' /
46     C
47     C
48     C      INITIALIZE
49     C
50     C
51     C      READ SUN-CENTERED EPHEMERITIES VIA JPL READE
52     C
53     C      80      T1      = TTRE * TUTODA
54     C      I          = T1
55     C      T2          = I
56     C      TEPEJD    = TEPOJD + T2

```

```

57      TEPEJS = TEPOJS + (T1 - T2) * F86400
58      I      = TEPEJS / F86400
59      T3     = I
60      TEPEJS = TEPEJS - F86400 * T3
61      TEPEJD = TEPEJD + T3
62      CALL READE (TEPEJD, TEPEJS, I)
63      IF (I .EQ. 0) GO TO 104
64      WRITE (IOPRT, 82) IEPER1, (IEPER2(J,I), J = 1, 4)
65      WRITE (IOMES, 82) IEPER1, (IEPER2(J,I), J = 1, 4)
66      82    FORMAT (12A6)
67      WRITE (IOMES, 102)
68      102   FORMAT (' READE CALLED BY EPHEM')
69      LOGOUT = .TRUE.
70      RETURN
71      C
72      C      CONVERT TO INERTIAL COORDINATES (CENTER AT SOLAR BARY-CENTER)
73      C
74      104   DO 110 I = 1, 3
75          TV1(I) = F0
76      110   DO 130 J = 1, 12
77          DO 130 J = 1, 12
78          IF (IREQ(J) .NE. 2) GO TO 130
79          K = JKLOC (J)
80          DO 120 I = 1, 3
81          TV1(I) = TV1(I) + GOOR(K) * TABOUT(I,J)
82          TV2(I) = TV2(I) + GOOR(K) * TABOUT(I+3,J)
83      120   CONTINUE
84      130   DO 150 J = 1, 12
85          IF (IREQ(J) .NE. 2) GO TO 150
86          K = JKLOC (J)
87          DO 140 I = 1, 3
88          ROOEIN(I+K-1) = TABOUT(I,J) - TV1(I)
89          VOOEIN(I+K-1) = TABOUT(I+3,J) - TV2(I)
90      140   CONTINUE
91      150   CONTINUE
92      C
93      C      RESOLVE PLANETS INTO INERTIALLY BASED COMPONENTS PER LIST JKPTS
94      C
95      IF (JKPTSN .NE. 0) CALL PLANET
96      C
97      RETURN
98      END

```

## BELLCOMM. INC.

Program: EPHEM (element EPH2CO)

Purpose: Determine position and velocity of two gravitating bodies in circular orbit about each other.

Called by: STEPIN

Calls: SIN, COS

Errors and Diagnostics: None

Language: Fortran subroutine

Computations:

$$RO1EIN_x = - GO2 \cos (TTRE - TEE)$$

$$RO1EIN_y = - GO2 \sin (TTRE - TEE)$$

$$VO1EIN_x = - RO1EIN_y$$

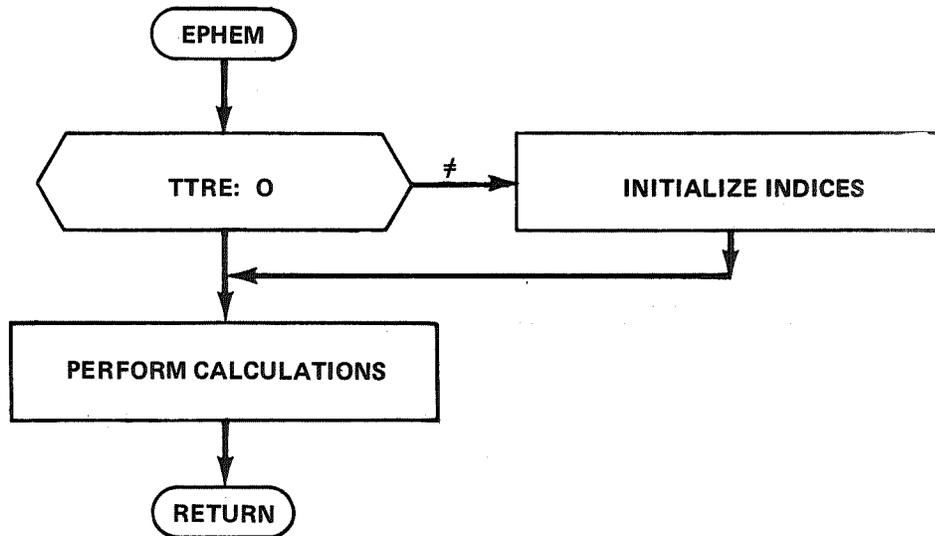
$$VO1EIN_y = RO1EIN_x$$

$$RO2EIN_x = GO1 \cos (TTRE - TEE)$$

$$RO2EIN_y = GO1 \sin (TTRE - TEE)$$

$$VO2EIN_x = - RO2EIN_y$$

$$VO2EIN_y = RO2EIN_x$$



VMASS\*RFJK.EPH2C0

```

1      C      EPHEM      D NOVAK AND R JESSUP      BELLCOMM      01 APR 69
2      C
3      C      PURPOSE
4      C
5      C      DETERMINE POSITIONS AND VELOCITIES OF TWO GRAVITATING POINTS IN
6      C      CIRCULAR ORBIT ABOUT EACH OTHER.  NORMALIZED SYSTEM WITH TOTAL
7      C      MASS 1 AND ANGULAR ROTATION 1 IN INERTIAL COORDINATES.
8      C
9      C      INPUTS
10     C
11     C      TTRE      CURRENT VALUE TRAJECTORY TIME (NORMALIZED ELAPSED EPH T)
12     C      TEE      INITIAL VALUE NORMALIZED EPHEMERIS TIME
13     C      GOO      TABLE OF GRAVITATIONAL CONSTANTS
14     C
15     C      OUTPUTS
16     C
17     C      ROOEIN    POSITION ITH POINT INERTIAL
18     C      VOOEIN    VELOCITY ITH POINT INERTIAL
19     C
20     C      SUBROUTINE EPHEM
21     C
22     C      INCLUDE QTYPE
23     C      INCLUDE QVAR
24     C      INCLUDE QTEM
25     C      INCLUDE QEPH
26     C      INCLUDE QSTORE
27     C      INCLUDE EVARI
28     C      INCLUDE EVARS
29     C      INCLUDE QFAC
30     C
31     C      IF (TTRE .NE. F0) GO TO 10
32     C          JK(1) = 1
33     C          JK(2) = 2
34     C          JKN   = 2
35     C          K01   = JKLOC (JK(1))
36     C          K02   = JKLOC (JK(2))
37     10  T1 = TTRE - TEE
38     C      T2 = COS (T1)
39     C      T3 = SIN (T1)
40     C      T4 = T2 * GOO(K01)
41     C      T5 = T3 * GOO(K01)
42     C      T6 = - T2 * GOO(K02)
43     C      T7 = - T3 * GOO(K02)
44     C
45     C      ROOEIN(K02 ) = T4
46     C      ROOEIN(K02+1) = T5
47     C      ROOEIN(K01 ) = T6
48     C      ROOEIN(K01+1) = T7
49     C      VOOEIN(K02 ) = - T5
50     C      VOOEIN(K02+1) = T4
51     C      VOOEIN(K01 ) = - T7
52     C      VOOEIN(K01+1) = T6
53     C      RETURN
54     C      END

```

Program: EVNACC (i, j, k, a, c, m)

Purpose: Build and maintain table of trajectory events.

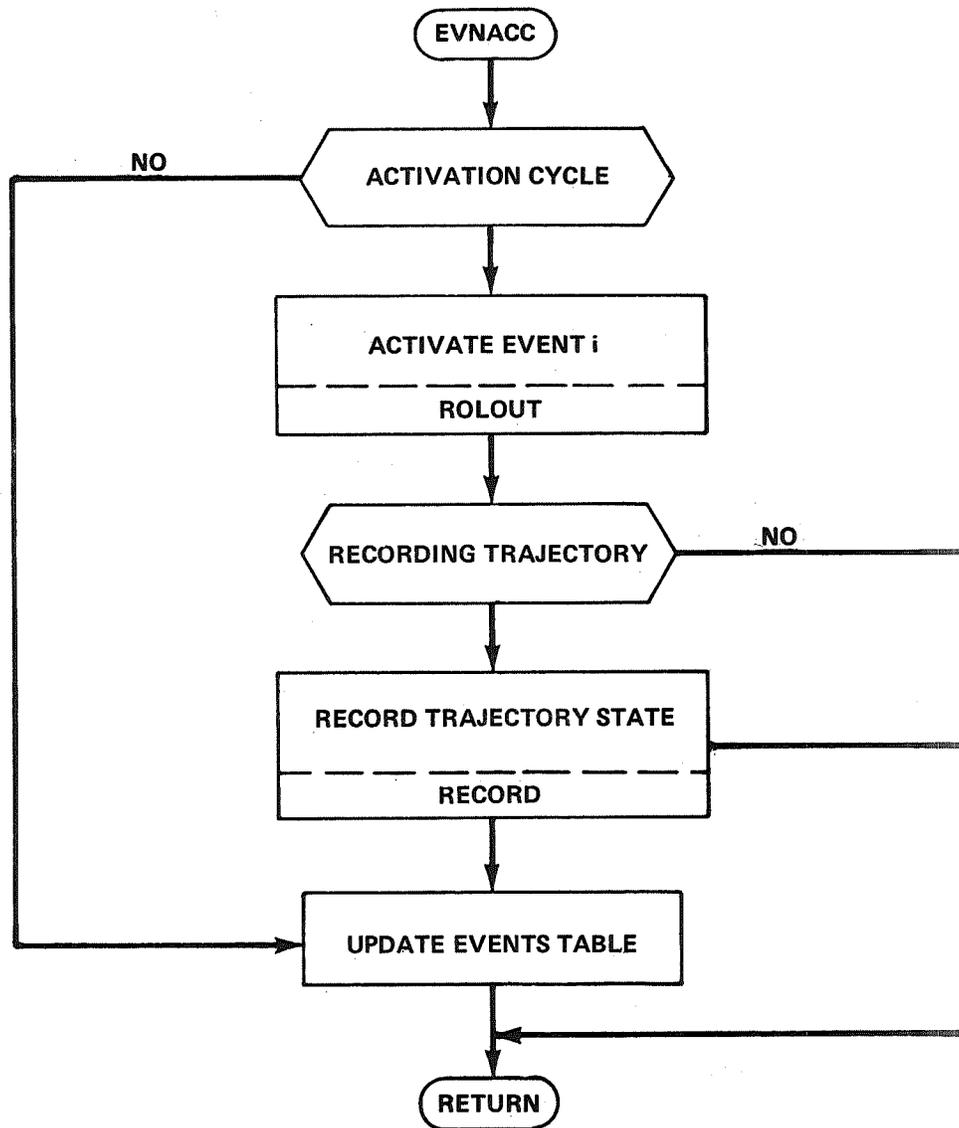
Called by: SIMXXX

Calls: DEVAR, INDRS, RECORD, ROLOUT

Errors and Diagnostics: An overflow of the events table, IEV, causes an error indication and program log out.

Language: Fortran subroutine

Computations: None



VMASS\*RFJK.EVNACC

```

1      C      EVNACC   R JESSUP - BELLCOMM - 06 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      VMASS USER CODING IN THE SIMULATOR (SIMXXX) CONSISTS OF SUCCESSIVE
6      C      CALLS TO EVNACC, ONE CALL FOR EACH TRAJECTORY EVENT.  ON EVENT
7      C      ACTIVATION (IEVFLG I OR 0) EVNACC RETURNS CONTROL TO ONE TIME
8      C      EVENT CALCULATIONS FOLLOWING THE CALL.  ON TABLE UPDATES (IEVFLG 1
9      C      AS SET BY THE SIMULATOR AFTER EVENT ACTIVATION AND BEFORE MOVING
10     C      ON TOWARD NEXT EVENT) EVNACC RETURNS CONTROL TO NEXT CALL OR TO
11     C      THE SIMULATOR AFTER ALL CALLS HAVE BEEN MADE.
12     C
13     C
14     C      CALLING SEQUENCE
15     C
16     C      CALL EVNACC (I, IACT, IDACT, CRIT, AIMPT, $MISS)
17     C
18     C      INPUTS
19     C
20     C      I          ANY EVENT DESCRIBED IN THE SIMULATOR (EVENTS LIST)
21     C      IACT      ''ACTIVATING'' EVENT FOR I (IACT MUST ''OCCUR'' BEFORE I)
22     C      IDACT     ''DE-ACTIVATING'' EV FOR I (THE ''OCCURRENCE'' OF IDACT
23     C              PRE-EMPTS FURTHER CONSIDERATION OF I)
24     C      CRIT      EVENT CRITERION
25     C      AIMPT     AIM-POINT OF THE CRITERION
26     C      MISS      CALLING PRG STATEMENT NUM TO RECEIVE CONTROL ON EV MISS
27     C
28     C      TRCI      REFERENCE CONIC TIME INCREMENT FOR NORMAL COMPUTING STEP
29     C      IEVFLG    NAME OF EVENT EXPECTED TO BE DECLARED (AS SET BY 'EVNDT')
30     C              OR FLAG SET TO 0 BY THE SIMULATOR FOR TABLE UPDATES
31     C
32     C      OUTPUTS
33     C
34     C      TEVI      TIME REMAINING FOR OCCURRENCE OF APPARENT NEAREST EVENT
35     C      IEVCUR    LATEST OR CURRENT DECLARED EVENT
36     C
37     C      EVENTS TABLE LAYOUT
38     C
39     C      EVN(1,I) CRITERION DEVIATION FROM AIMPT IN LAST COMPUTING INTERVAL
40     C      IEV(3,I) EVENT I
41     C      IEV(4,I) DE-ACTIVATING EVENT FOR I
42     C      IEV(5,I) VAR INDEX POINTER TO CRITERION FOR I
43     C      IEV(6,I) VAR INDEX POINTER TO AIM-POINT FOR I
44     C      IEV(7,I) STEPS TAKEN IN ATTEMPT TO CONVERGE TO THIS EVENT
45     C      IEV(8,I) FLAG FOR EVENT INITATION THROUGH PROGRAM CONTROL
46     C
47     C      SUBROUTINE EVNACC (I, IACT, IDACT, CRIT, AIMPT, MISS)
48     C
49     C      INCLUDE QTYPE
50     C      INCLUDE QVAR
51     C      INCLUDE QTEM
52     C      INCLUDE QFAC
53     C      INCLUDE QIEV
54     C      INCLUDE EVARIO
55     C      INCLUDE EVARS
56     C      INCLUDE EVARI

```

```

57      INCLUDE EVARL
58      C
59      IF (IEVFLG .EQ. 1) GO TO 30
60      C
61      C      EVENT ACTIVATION (IEVFLG I OR 0). NOTE EVENT AND MAKE RECORD
62      C      OF PRESENT PROGRAM STATE.
63      C
64      IF (IEVFLG .NE. I .AND. IEVFLG .NE. 0) RETURN 6
65      LXCEV = .TRUE.
66      IEVCUR = I
67      CALL ROLOUT
68      IF (.NOT. LRESKP) CALL RECORD
69      RETURN
70      C
71      C      TABLE UPDATING (IEVFLG 1). REMOVE INHIBITED (DEACTIVATED) ENTRIES
72      C
73      30  IF (IEVCUR .NE. IDACT) GO TO 50
74          DO 40 K = 1, IEVN
75              IF (IEVCUR .NE. IEV(4,K)) GO TO 40
76                  DO 32 M = 1, 8
77                      32  IEV(M,K) = 0
78                          DO 34 M = IEVM, 0, -1
79                              IEVN = M
80                                  34  IF (IEV(3,M) .NE. 0) RETURN 6
81                                  40  CONTINUE
82      RETURN 6
83      50  IF (I .NE. IACT .AND. IEVCUR .NE. IACT) RETURN 6
84      C
85      C      INSERT ENABLED EVENTS (WHICH CAN LOGICALLY FOLLOW I IN TIME FLOW)
86      C
87      DO 60 K = 1, IEVN
88          60  IF (I .EQ. IEV(3,K)) RETURN 6
89      DO 70 K = 1, IEVM
90          IF (IEV(3,K) .NE. 0) GO TO 70
91          IEVN = MAX (IEVN, K)
92          IEV(8,K) = 0
93          IEV(7,K) = 0
94          IEV(6,K) = INDRS (AIMPT)
95          IEV(5,K) = INDRS (CRIT )
96          IEV(4,K) = IDACT
97          IEV(3,K) = I
98          EVN(1,K) = DEVAR (IEV(5,K), IEV(6,K))
99          RETURN 6
100     70  CONTINUE
101     C
102     LOGOUT = .TRUE.
103     WRITE (IOMES, 90)
104     90  FORMAT (' EVNACC *** EVENTS TABLE OVERFLOW')
105     RETURN 6
106     C
107     END

```

Program: EVNDT

Purpose: Apply linear interpolation to estimate time differences between present trajectory state and each event in the events table. If the predicted time for an event is less than the normal computing interval, iteration to the event is initiated. An adjusted computing interval is taken in an attempt to reach predicted time accurately. On reaching predicted time, event activation is initiated.

Called by: SIMXXX

Calls: DEVAR

Errors and Diagnostics: A diagnostic is given whenever it appears that an event has been passed, or whenever iteration to reach an event accurately is not successful. Activation of either such event is initiated.

Language: Fortran subroutine

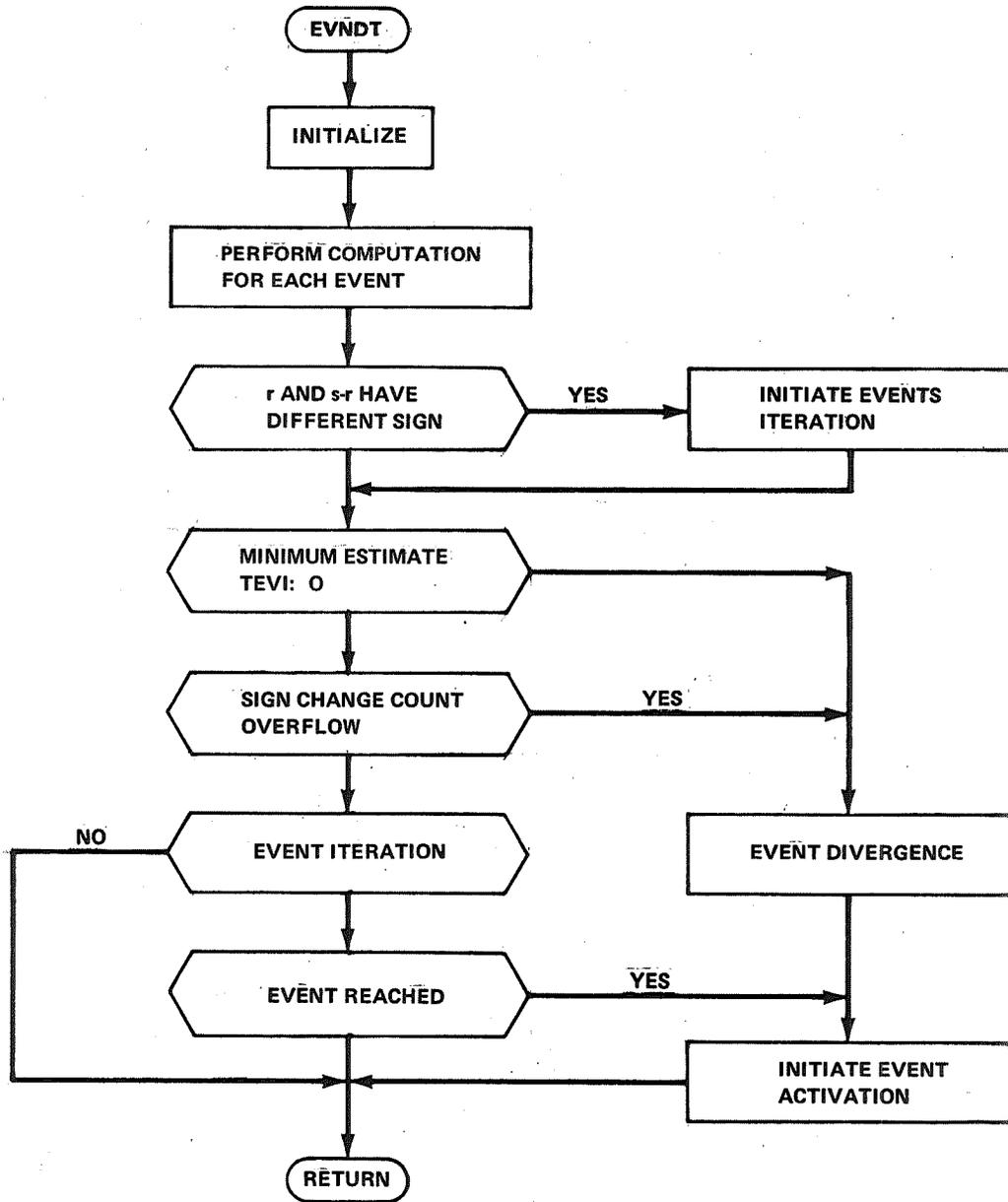
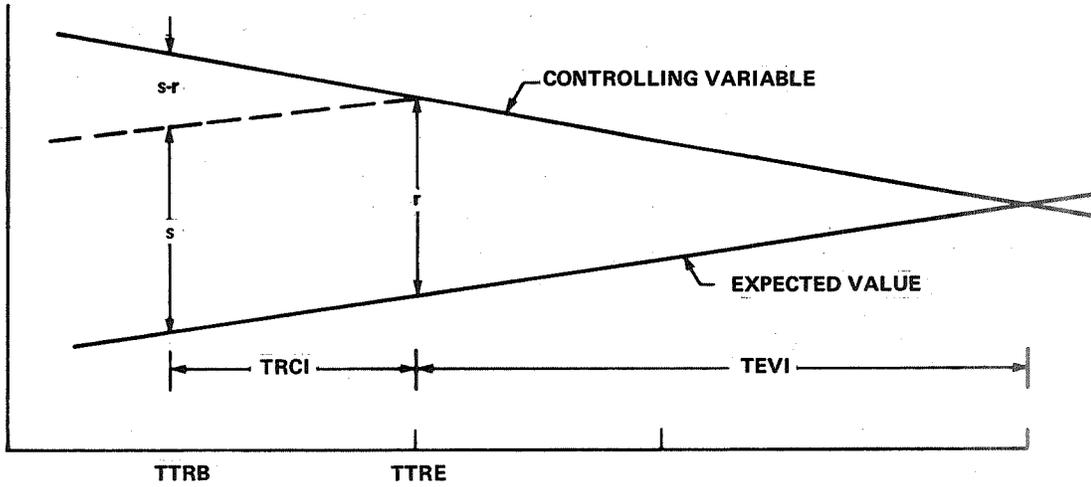
Computation: The predicted time, TEVI, to an event is given by the linear extrapolation

$$\frac{\text{TEVI}}{\text{TRCI}} = \frac{r}{s-r}$$

where

TRCI = normal reference conic computing interval  
increment

r, s = present and previous differences, respectively,  
between the event controlling variable and its  
expected value, sometimes called aim point or  
criterion.



VMAS\*RFJK.EVNDT

```

1      C
2      C  INPUTS
3      C
4      C      TRCI      REFERENCE CONIC TIME INCREMENT OVER COMPUTING INTERVAL
5      C      LINTDN    LENGTH OF COMP INTERVAL IS DOWN FROM EARLIER VALUE
6      C
7      C  OUTPUTS
8      C
9      C      TEVI      TIME TO GO TO NEAREST EVENT
10     C      IEVFLG    IF SOME EVENT WILL OCCUR IN CUR COMP INT THIS IS ITS NAME
11     C      LIMPEV    FLAG FOR INDICATING IMPENDING EVENT OCCURRENCE
12     C      LXCEV     LOGICAL EVENT EXECUTION FLAG - WE ARE THERE
13     C
14     C  PURPOSE
15     C
16     C      FIND SMALLEST TIME TO GO FOR 'NEAREST' EVENT AS BASED ON EARLIER
17     C      AND PRESENT CRITERION - AIMPOINT DEVIATIONS
18     C
19     C  SUBROUTINE EVNDT
20     C
21     C      INCLUDE QTYPE
22     C      INCLUDE QVAR
23     C      INCLUDE QTEM
24     C      INCLUDE QFAC
25     C      INCLUDE QIEV
26     C      INCLUDE EVARI
27     C      INCLUDE EVARL
28     C      INCLUDE EVARS
29     C      INCLUDE EVARIO
30     C
31     C      TEVI = FINITY
32     C      IF (IEVN .EQ. 0) RETURN
33     C      IF (.NOT. LINTDN) GO TO 205
34     C      LIMPEV = .FALSE.
35     C      LINTDN = .FALSE.
36     205 DO 280 K = 1, IEVN
37     C      IF (IEV(3,K) .EQ. 0) GO TO 280
38     C      IF (IEV(8,K) .EQ. 0) GO TO 210
39     C      TEVI = TRCI
40     C      GO TO 230
41     210 T1 = DEVAR (IEV(5,K), IEV(6,K))
42     C      IF (SIGN (F1, EVN(1,K)) .NE. SIGN (F1, T1)) IEV(7,K)=IEV(7,K)+1
43     C      T2 = EVN(1,K) - T1
44     C      IF (T2 .NE. F0) TEVI = MIN (TEVI, TRCI * EVN(1,K) / T2)
45     C      IF (TEVI .GT. F0) GO TO 240
46     220 WRITE (IOMES, 222)
47     222 FORMAT (// ' EVNDT *** DIVERGENCE')
48     C      DO 224 I = 1, IEVN
49     C      T1 = DEVAR (IEV(5,K), IEV(6,I))
50     224 WRITE (IOMES, 226) I, (IEV(J,I), J=3,8), EVN(1,I), T1
51     226 FORMAT (' IEV(' , I3, ') ' , 2A9, 4I9, 2G30.8)
52     230 IEVFLG = IEV(3,K)
53     C      LXCEV = .TRUE.
54     C      LIMPEV = .FALSE.
55     C      RETURN
56     240 IF (IEV(7,K) .EQ. 0) GO TO 250

```

```
57             LIMPEV = .TRUE.
58             IEV(7,K) = IEV(7,K) + 1
59             IF (IEV(7,K) .GT. 12) GO TO 220
60             250   IF (.NOT. LIMPEV) GO TO 280
61             T3 = ABS (TEVI - TRCI)
62             IF (T3 .LE. T10) GO TO 230
63             IF (T3 .LE. FINRES * (TEVI + TRCI)) GO TO 230
64             280   CONTINUE
65             RETURN
66             END
```

Program: GETTAP

Purpose: Get current record from ephemeris tape

Called by: READE

Calls: None

Errors and Diagnostics: Initiates READE error return on faulty ICW flag setting or on current date out of range of assigned tape.

Language: Fortran subroutine

Computations: None

Flowchart: (Omitted: This program was furnished by JPL)

Program: INDEX

Purpose: Update back values of certain program and trajectory state variables.

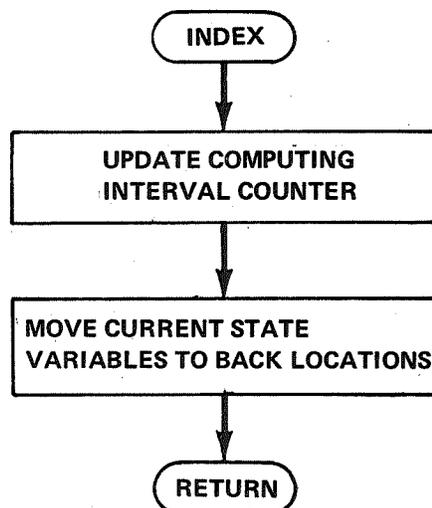
Called by: STEPIN

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



## VMASS\*RFJK.INDEX

```

1      C
2      C  PURPOSE
3      C
4      C    THIS PROGRAM INDEXES THE APPROPRIATE VALUES AT THE END OF A
5      C    COMPLETED COMPUTING INTERVAL TO COMPRISE THE BEGINNING VALUES FOR
6      C    THE NEXT INTERVAL.
7      C
8      C  INPUTS
9      C
10     C    RSCEIN
11     C    VSCEIN
12     C    RRCEVM  REFERENCE CONIC FINAL POSITION REL TO THE VIRT MASS
13     C    VRCEVM    "          VELOCITY          "
14     C    RVMEIN  VIRTUAL MASS FINAL POSITION RELATIVE TO INERTIAL FRAME
15     C    VVMEIN    "          VELOCITY          "
16     C    GVME    G TIMES FINAL VIRTUAL MASS MAGNITUDE
17     C    QVME    "          MAGNITUDE RATE
18     C    TEPE    FINAL EPHEMERIS TIME (AT END OF OLD INTERVAL)
19     C    RRCDVM  REFERENCE CONIC FINAL DISTANCE FROM VIRTUAL MASS
20     C    RSCDJK  DIST FROM SPACECRAFT TO MASS PT JK AT END OF COMP INT
21     C
22     C  OUTPUTS
23     C
24     C    RSCBIN
25     C    VSCBIN
26     C    RRCBVM  REFERENCE CONIC INITIAL POSITION REL TO THE VIRT MASS
27     C    VRCBVM    "          VELOCITY          "
28     C    RVMBIN  VIRTUAL MASS INITIAL POSITION REL TO THE INERTIAL FRAME
29     C    VVMBIN    "          VELOCITY          "
30     C    GVMB    G TIMES INITIAL VIRTUAL MASS MAGNITUDE
31     C    QVMB    "          MAGNITUDE RATE
32     C    TEPB    INITIAL EPHEMERIS TIME (AT START OF NEW INTERVAL)
33     C    RRCAVM  REFERENCE CONIC INITIAL DISTANCE FROM VIRTUAL MASS
34     C    RSCAJK  DIST FROM SPACECRAFT TO MASS PT JK AT BEG OF COMP INT
35     C
36     C  SUBROUTINE INDEX
37     C
38     C    INCLUDE QTYPE
39     C    INCLUDE QVAR
40     C    INCLUDE QEPH
41     C    INCLUDE QIEV
42     C    INCLUDE EVARI
43     C    INCLUDE EVARS
44     C    INCLUDE EVARV
45     C
46     C    INCNTR = INCNTR + 1
47     C    DO 10 I = 1, JKN
48     C        J = JKLOC (JK(I))
49     C    10    RSCA00(J) = RSCD00(J)
50     C    DO 20 I = 1, 3
51     C        RSCBIN(I) = RSCEIN(I)
52     C        VSCBIN(I) = VSCEIN(I)
53     C        RRCBVM(I) = RRCEVM(I)
54     C        VRCBVM(I) = VRCEVM(I)
55     C        RVMBIN(I) = RVMEIN(I)
56     C    20    VVMBIN(I) = VVMEIN(I)

```

## VMASS SUBROUTINE INDEX

```
57      DO 30 I = 1, IEVN
58          30      IF (IEV(3,I) .NE. 0) EVN(1,I) = DEVAR (IEV(5,I), IEV(6,I))
59      RRCAVM = RSCDVM
60      GVMB   = GVME
61      QVMB   = QVME
62      TEPB   = TEPE
63      RETURN
64      END
```

Program: INITAL

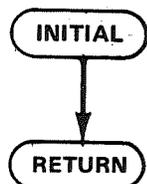
Purpose: Perform user coded calculations just after the data deck is read into the computer.

Called by: SETUP

Calls: None

Language: Dummy Fortran subroutine

Computations: None



VMASS SUBROUTINE INITAL

VMASS\*RFJK.INITAL

```
1      C
2      C  PURPOSE
3      C
4      C  PROVIDE FOR USER CALCULATIONS ON DATA DECK JUST AFTER READ IN
5      C
6      C  SUBROUTINE INITAL
7      C
8      C
9      C
10     C  RETURN
11     C  END
```

Program: KEPLER

Purpose: Given the initial position and velocity vectors and the vector orbital elements relative to the virtual mass, this program computes the final relative position and velocity vectors for an arc of the reference conic section.

Called by: STEPIN

Calls: CONICT, CROSS, DOT, UNITV, VALUE

Errors and Diagnostics: Indicates log out from CONICT. Initiates log out on non-uniform convergence of the computation series or on failure to converge after several terms.

Language: Fortran subroutine

Computations:

$$S_{2,k} = CT2_{k-1} \cdot TTRI^2 + TTRI$$

$$V_{1,k} = VRCBVM \cdot S_{2,k}$$

$$V_{2,k} = RRCBVM + V_{1,k}$$

$$S_{4,k} = HRCAVM^2 \div GVMB$$

$$S_{5,k} = S_{4,k} \div (ERCBVM \cdot V_{2,k} + |V_{2,k}|)$$

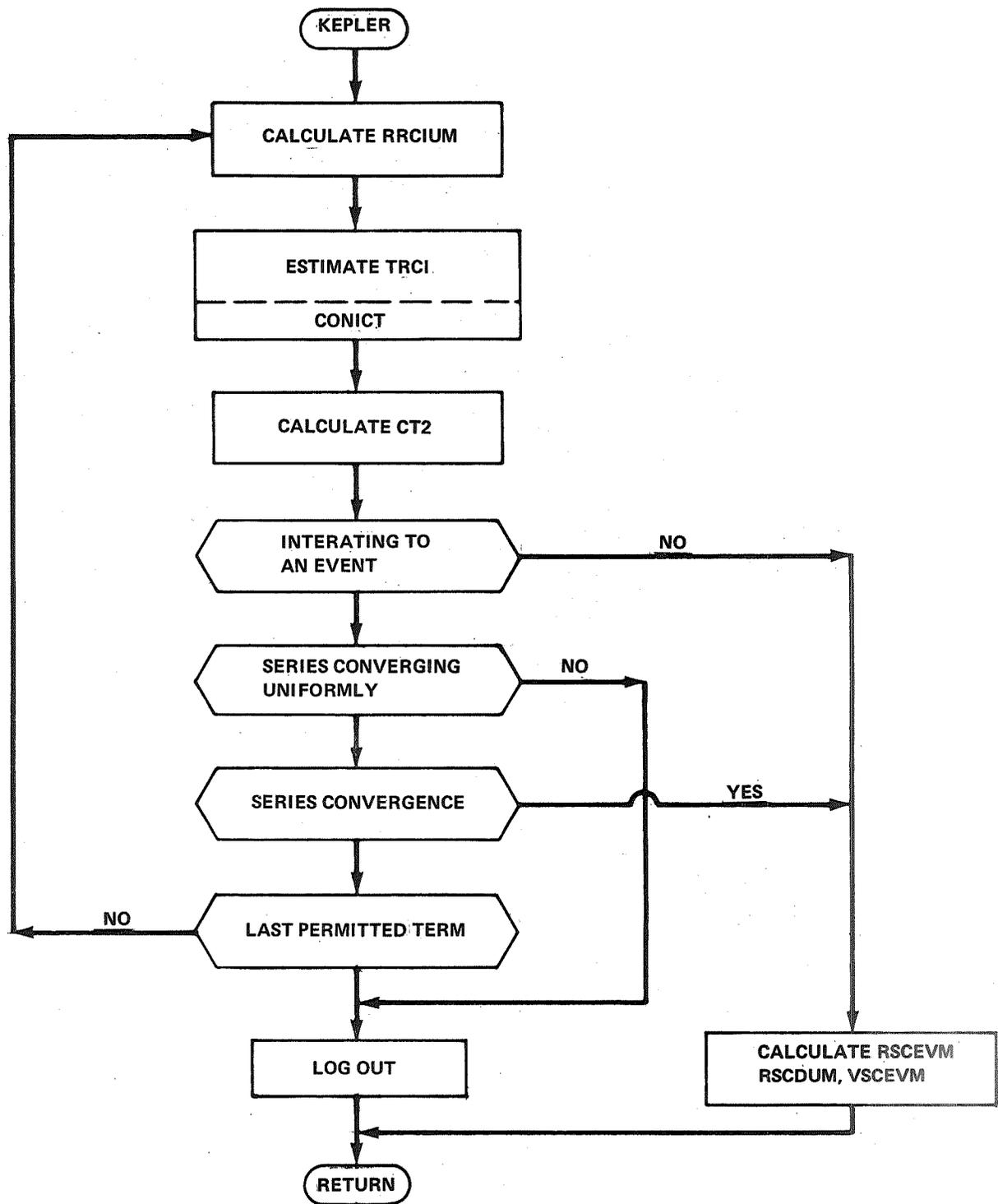
$$RRCIVM_k = V_1 + V_2 \cdot (S_{5,k} - 1)$$

$$CT2_k = S_{2,k} \cdot TRCI^{-2} - TRCI^{-1}$$

$$RRCEVM = V_{2,k} \cdot S_{5,k}$$

$$RRCDVM = |RRCEVM|$$

$$VRCEVM = HRCBVM \times \left( ERCBVM + \frac{RRCEVM}{|RRCEVM|} \right) \div S_{4,k}$$



VMASS\*RFJK.KEPLER

```

1      C
2      C   PURPOSE
3      C
4      C   GIVEN THE INITIAL POSITION AND VELOCITY VECTORS AND THE VECTOR
5      C   ORBITAL ELEMENTS RELATIVE TO THE VIRTUAL MASS, THIS PROGRAM COM-
6      C   PUTES THE FINAL RELATIVE POSITION AND VELOCITY VECTORS FOR AN ARC
7      C   OF THE REFERENCE CONIC SECTION.  DEPENDING UPON A FLAG SETTING THE
8      C   PROGRAM OPTIONALLY WILL ITERATE ACCURATELY TO A DESIRED FINAL TIME
9      C   OR, IN A SINGLE PASS, WILL OBTAIN FINAL CONDITIONS WHICH ONLY
10     C   APPROXIMATE THE SPECIFIED FINAL TIME.
11     C
12     C   INPUTS
13     C
14     C   LOOP      FLAG SET .TRUE. IF FINAL TIME TO BE ITERATED ACCURATELY
15     C             TO THE PRESCRIBED VALUE
16     C   RRCBVM   INITIAL POSITION ON REF CONIC RELATIVE TO VIRT MASS
17     C   VRCBVM   INITIAL VELOCITY      ''
18     C   HRCBVM   ANGULAR MOMENTUM VECTOR OF REF CONIC REL TO THE VIRT MASS
19     C   ERCBVM   ECCENTRICITY          ''
20     C   HRCAVM   MAGNITUDE OF THE VECTOR ANGULAR MOMENTUM
21     C   ERCAVM   ''                    ECCENTRICITY
22     C   GVMB     PRODUCT OF UNIV GRAVIT CONST AND VIRT MASS MAG
23     C   TEPI     DESIRED (EPHEMERIS) TIME INCREMENT
24     C   TTO     TOLERANCE ON ACCURACY OF COMPUTED TIME INCREMENT
25     C   CT2     COEFFICIENT OF SQUARED TERM IN BIASED TIME EXPANSION.
26     C             THIS IS COMPUTED INTERNALLY IN THIS SUBROUTINE, BUT AN
27     C             INITIAL VALUE (OF 0) MUST BE PROVIDED FOR THE STARTING
28     C             COMPUTATION INTERVAL.  AFTER THIS THE LAST VALUE OF THE
29     C             PRECEEDING STEP IS USED AS A CURRENT FIRST GUESS.
30     C   KEPMAX  MAX ALLOWABLE NUM OF ITER IN THIS PROG FOR FINAL COND
31     C
32     C   OUTPUT
33     C
34     C   RRCEVM   REF CONIC FINAL POSITION REL TO VIRT MASS
35     C   VRCEVM   REF CONIC FINAL VELOCITY REL TO VIRT MASS
36     C   RRCDVM   REF CONIC FINAL DISTANCE FROM VIRT MASS
37     C   RRCIVM   REFERENCE CONIC POSITION INCREMENT RELATIVE TO V MASS
38     C   CT2
39     C
40     C   SUBROUTINE KEPLER
41     C
42     C   INCLUDE QTYPE
43     C   INCLUDE QVAR
44     C   INCLUDE QFAC
45     C   INCLUDE QTEM
46     C   INCLUDE EVARI
47     C   INCLUDE EVARIO
48     C   INCLUDE EVARL
49     C   INCLUDE EVARS
50     C   INCLUDE EVARV
51     C
52     C   S1 = FINITY
53     C   DO 50 K = 1, KEPMAX
54     C     S2 = (CT2 * TEPI + F1) * TEPI
55     C     DO 10 I = 1, 3
56     C       SV1(I) = VRCBVM(I) * S2

```

```

57      10      SV2(I) = RRCBVM(I) + SV1(I)
58          S3 = VALUE (SV2)
59          S4 = (HRCAVM / GVMB) * HRCAVM
60          S5 = S4 / (DOT (ERCBVM, SV2) + S3)
61          S6 = S5 - F1
62      C
63          DO 20 I = 1, 3
64      20      RRCIVM(I) = SV1(I) + SV2(I) * S6
65      C
66          CALL CONICT
67          IF (.NOT. LOGOUT) GO TO 30
68          WRITE (IOMES, 25)
69      25      FORMAT (' CONICT CALLED BY KEPLER')
70          RETURN
71      30      CT2 = (S2 / TRCI - F1) / TRCI
72          IF (.NOT. LIMPEV) GO TO 70
73          S7 = ABS (TTRI - TRCI)
74          IF (S7 .LE. S1) GO TO 40
75          WRITE (IOMES, 110)
76          LOGOUT = .TRUE.
77          RETURN
78      40      IF (S7 .LE. TTO) GO TO 70
79          IF (S7 .LE. FINRES * (TTRI + TRCI)) GO TO 70
80      50      S1 = S7
81          WRITE (IOMES, 120), KEPMAX
82          RETURN
83      C
84      70      DO 75 I = 1, 3
85      75      RRCEVM(I) = SV2(I) * S5
86          RRCDVM = VALUE (RRCEVM)
87          CALL UNITV (RRCEVM, SV4)
88          DO 80 I = 1, 3
89      80      SV6(I) = (ERCBVM(I) + SV4(I)) / S4
90          CALL CROSS (HRCBVM, SV6, VRCEVM)
91      C
92          RETURN
93      C
94      110     FORMAT (' KEPLER *** CONVERGENCE NOT UNIFORM')
95      120     FORMAT (' KEPLER *** TERMS IN SERIES EXCEEDING', I4)
96      C
97          END

```

Program: LODCOM

UPDCOM

Purpose: Clear working COMMON storage to zero and load standard constants and calculations.

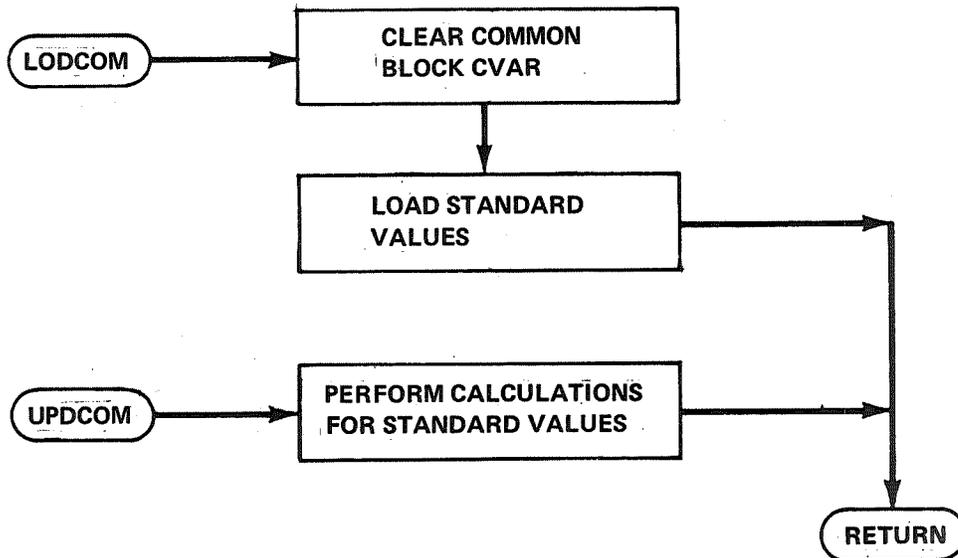
Called by: SIMXXX, SETUP

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: As shown in program



VMASS\*RFJK.LODCOM

```
1      C
2      C  PURPOSE
3      C
4      C    IN THIS PROGRAM 'LODCOM' CLEARS ALL VARIABLE LOCATIONS TO ZERO AND
5      C    LOADS VALUES SHOWN FOR LOCATIONS NORMALLY REGARDED AS CONSTANT
6      C
7      C    SUBROUTINE LODCOM
8      C
9      C    INCLUDE QTYPE
10     C    INCLUDE QVAR
11     C    INCLUDE QEPH
12     C    INCLUDE QFAC
13     C    INCLUDE QPHC
14     C    INCLUDE QTEM
15     C    INCLUDE QGRADE
16     C
17     C    INCLUDE EVARI
18     C    INCLUDE EVARS
19     C    INCLUDE EVARIO
20     C
21     C    DO 10 I = 1, 1160
22     C      KVAR(I) = 0
23     C      IEVM   = 20
24     C      MVAR   = 1160
25     C      KEPMAX = 6
26     C      NMAX   = 30
27     C
28     C      IOCDR = 5
29     C      IOEPH = 12
30     C      IOMES = 6
31     C      IOPCH = 7
32     C      IOPRL = 9
33     C      IOPRT = 6
34     C      IOROL = 8
35     C      IOSCR = 11
36     C      IPLOTM = 1
37     C      IPLATM = 10000
38     C      JKSPEC = 21
39     C      IPNOWM = 500
40     C
41     C
42     C      F0      = .000
43     C      F001    = .001D0
44     C      F00001  = .00001D0
45     C      FPT5    = .5D0
46     C      F1PT1   = 1.1D0
47     C      FPT15   = .15D0
48     C      FPT4    = .4D0
49     C      F1      = 1.0
50     C      F1PT5   = 1.5
51     C      F2      = 2.0
52     C      F2PT5   = 2.5
53     C      F3      = 3.0
54     C      F4      = 4.0
55     C      F5      = 5.0
56     C      F6      = 6.0
```

```

57      F7      = 7.0
58      F8      = 8.0
59      F9      = 9.0
60      F10     = 10.0
61      F11     = 11.0
62      F12     = 12.0
63      F13     = 13.0
64      F14     = 14.0
65      F15     = 15.0
66      F20     = 20.00
67      F24     = 24.000
68      F30     = 30.00
69      F60     = 60.000
70      F1440   = 1440.000
71      F3600   = 3600.000
72      F43200  = 43200.00
73      F86400  = 86400.000
74      FINITY  = 1.0E37
75      FINTES  = 1.D-18
76      FTT0    = FPTS
77      FRT0    = F5
78      C
79      RTOD    = 57.29577951308232D0
80      FTTOKM  = .3048D-3
81      RUTOAU  = F1
82      TUTODA  = F1
83      C
84      RTOPRF  = 1.D-15
85      DTAIDN  = FPTS
86      DTAIUP  = F1PT1
87      LTAIMN  = F1
88      C
89      C      FOR JPL READE AND FOR GRAVITATING BODIES REFERENCING
90      C
91      ICENT   = 10
92      ICW     = 1
93      TPD     = F1
94      AU      = F1
95      DO 210 I = 1, 11
96          IREQ(I) = 2
97      210    JK(I) = I
98      JKN     = 11
99      KMERC   = JKLOC (JK(1))
100     KVENUS  = JKLOC (JK(2))
101     KEARTH   = JKLOC (JK(3))
102     KMARS    = JKLOC (JK(4))
103     KJUP     = JKLOC (JK(5))
104     KSAT     = JKLOC (JK(6))
105     KURAN    = JKLOC (JK(7))
106     KNEP     = JKLOC (JK(8))
107     KPLUTO   = JKLOC (JK(9))
108     KSUN     = JKLOC (JK(10))
109     KMOON    = JKLOC (JK(11))
110     KEMB     = JKLOC (JK(12))
111     C
112     C      FOLLOWING DATA FROM JPL COMPUTER PROGRAM DPTRAJ
113     AUTOKM  = 149597893.D0

```

```

114      RETOKM = .63781492D4
115      RE      = RETOKM / AUTOKM
116      EMRAT  = 81.301D0
117      T1     = F86400 ** 2 / AUTOKM ** 3
118      GOO(KMERC ) = T1 * .221815976D5
119      GOO(KVENUS) = T1 * .324860103D6
120      GOO(KEARTH) = T1 * .398601200D6
121      GOO(KMARS ) = T1 * .428284438D5
122      GOO(KJUP  ) = T1 * .126707718D9
123      GOO(KSAT  ) = T1 * .379265257D8
124      GOO(KURAN ) = T1 * .578772346D7
125      GOO(KNEP  ) = T1 * .689057627D7
126      GOO(KPLUTO) = T1 * .732408934D5
127      GOO(KSUN  ) = T1 * .132712499D12
128      GOO(KMOON ) = T1 * .490278348D4
129      GOO(KEMB  ) = GOO(KEARTH) + GOO(KMOON )
130      RETURN
131      C
132      C
133      ENTRY UPDCOM
134      C
135      C      THIS SECTION FOR COMPUTED VALUES OF COMMON.  THE ABOVE CONSTANTS CAN
136      C      BE REPLACED VIA DDREAD IF DESIRED.
137      C
138      DTAI  = F1 / RTOPRF
139      DTOR  = F1 / RTOD
140      T1 = F0
141      DO 402 I = 1, JKN
142          IF (I .GT. 11) GO TO 404
143          K = JKLOC (JK(I))
144          T1 = T1 + GOO(K)
145      402 DO 412 I = 1, JKN
146          IF (I .GT. 11) GO TO 414
147          K = JKLOC (JK(I))
148          412 GOOR(K) = GOO(K) / T1
149      414 CONTINUE
150      RETURN
151      END

```

Program: PLANET

Purpose: Transform to inertial coordinates the planetographic state vectors of aspherical mass point representations of designated planets.

Called by: EPHEM

Calls: SIN, COS

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: Each mass point has inertial state vector

$$ROOEIN_k = ROOEIN_q + V_{1,k}$$

$$VOOEIN_k = VOOEIN_q + V_{2,k}$$

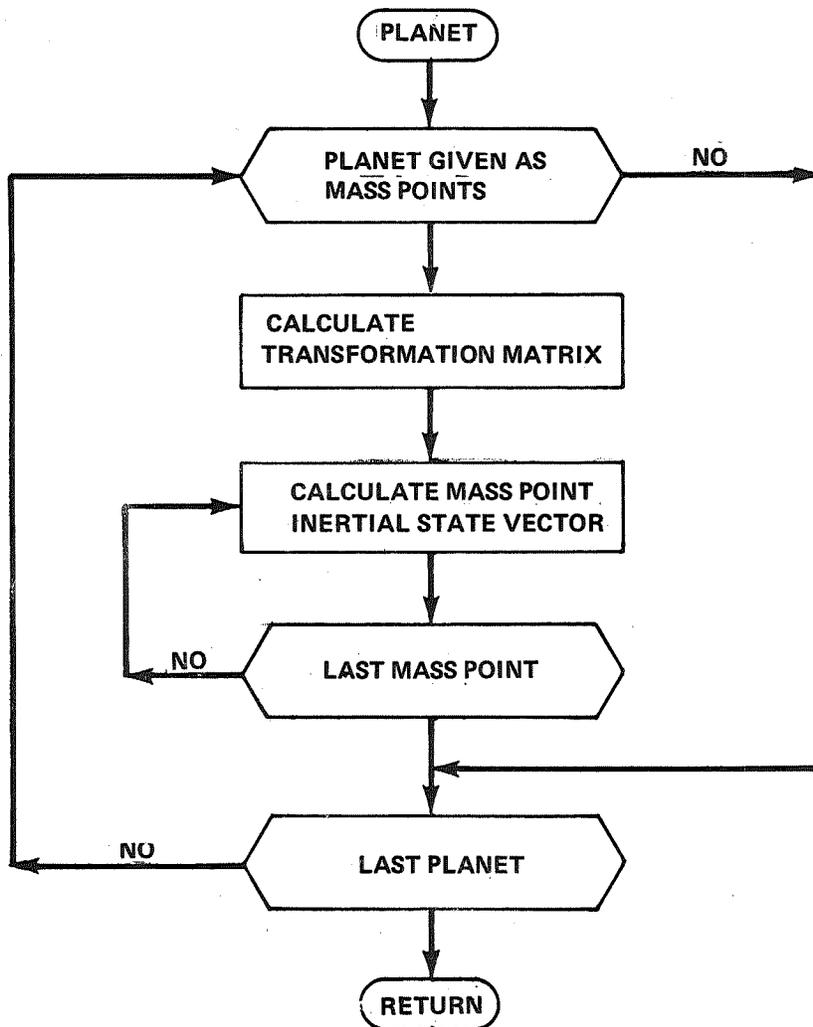
where  $ROOEIN_q$  and  $VOOEIN_q$  give the inertial state vector of the the planet itself.

$$V_{1,k} = W \cdot ROOEIN_q$$

$$V_{2,k,i} = SPLEOQ_q (W_{i,2} \cdot ROOEIN_{q,2} - W_{i,1} \cdot ROOEIN_{q,1})$$

$$i = 1, 2, 3$$

W = transformation matrix as given in program listing.  
See array X1.



VMASS\*RFJK.PLANET

```

1 C PLANET D NOVAK AND R JESSUP BELLCOMM 17 MAR 69
2 C
3 C PURPOSE
4 C
5 C TO TRANSFORM TO THE REFERENCE INERTIAL COORDINATE SYSTEM THE
6 C PLANETOGRAPHIC STATES (POS AND VEL) OF COLLECTIONS OF MASS POINTS
7 C REPRESENTING ASPHERICAL POTENTIALS OF DESIGNATED PLANETS
8 C
9 C INPUTS
10 C
11 C TEPOJD REFERENCE EPOCH OF EPHEMERIS TIME (JULIAN DAY COMPONENT)
12 C TEPOJS REFERENCE EPOCH OF EPHEMERIS TIME (JULIAN SEC COMPONENT)
13 C TEPEJD CURRENT DAY
14 C TEPEJS CURRENT DAY FRACTION IN SEC
15 C
16 C JKPTSN COUNT OF ALL POINT MASSES IN USE
17 C JKPTS PLANETARY MASS POINT LOCATORS
18 C
19 C JKPTS(1,I) FOR PLANET I SUBSCRIPT IN RMNOJK OF FIRST POINT MASS
20 C JKPTS(2,I) FOR PLANET I COUNT OF POINT MASSES
21 C
22 C FOR EA PLANET JK (JK = 01,...,11) AT SETUP OR RECONFIGURATION TIME
23 C
24 C RMNOJK PLANETOGRAPHIC POS VECTOR OF MASS PT MN OF PLANET JK
25 C
26 C DANEJK RIGHT ASCENSION OF ASCENDING NODE OF EQUATORIAL PLANE
27 C MEASURED FROM X REF DIRECTION IN BASE INERTIAL SYS XY PL
28 C DOBEJK OBLIQUITY OF EQ PL TO BASE INERTIAL XY PLANE
29 C DLNOJK LONGITUDE OF NODE AT REFERENCE TIME - (TEPOJD, TEPOJS)
30 C SPLEJK ANGULAR ROTATION RATE (SIDEREAL)
31 C
32 C FOR EA PLANET AT CURRENT TIME
33 C
34 C RJKEIN INERTIAL POS VECTOR OF CENTER OF MASS (JK = 01, ..., 11)
35 C VJKEIN INERTIAL VEL VECTOR OF CENTER OF MASS (JK = 01, ..., 11)
36 C
37 C OUTPUTS
38 C
39 C RJKEIN INERTIAL POS VECTOR OF EA MASS PT (JK = 11 + JKPTS + ..)
40 C VJKEIN INERTIAL VEL VECTOR OF EA MASS PT (JK = 11 + JKPTS + ..)
41 C
42 C
43 C EPHEMERITIES STORAGE LAYOUT (SEE ALSO JPL READE)
44 C
45 C R01EIN CURRENT INERTIAL POS 01 MERCURY 05 JUP 09 PLUTO
46 C R02EIN 02 VENUS 06 SAT 10 SUN
47 C ... 03 EARTH 07 URANUS 11 MOON
48 C R12EIN 04 MARS 08 NEP 12 EM B CEN
49 C
50 C RJKEIN RMNOJK AS LOCATED VIA LIST JKPTS (JK = 11 + ...)
51 C ...
52 C
53 C R..EIN CURRENT INERTIAL POS PLANETARY MASS PTS
54 C
55 C SUBROUTINE PLANET
56 C

```



Program: PREPAR

Purpose: Make first estimate of final state of the Virtual Mass and initialize perturbation equation calculation.

Called by: STEPIN

Calls: DOT

Errors and Diagnostics: None

Language: Fortran subroutine

Calculations:

$$T_1 = \text{TRCI} \cdot \text{VRCEVM} \cdot \text{RRCEVM} \cdot \text{RRCDVM}^{-2}$$

$$T_4 = \text{GVMB} \cdot \text{RRCDVM}^{-3}$$

$$T_5 = \text{QVMB} \cdot \text{RRCVAVM}^{-3}$$

$$T_6 = \text{TRCI}^2 \cdot T_5 \div 12$$

$$T_7 = \text{TRCI} (T_1 + 2) T_4 \div 4$$

$$T_8 = 1 - \text{TRCI}^2 \cdot T_4 \div 12$$

$$T_9 = \text{TRCI}^3 \cdot T_5 \div 20$$

$$T_{10} = 1 + \text{TRCI}^2 \cdot T_4 \cdot (1.5 + T_1 \div 10)$$

$$T_{11} = \text{TRCI}^3 \cdot T_4 \div 30$$

$$\begin{aligned} \text{SV}_1 &= \text{RVMBIN} + \text{TRCI} \cdot \text{VMBIN} \\ &+ T_9 \cdot \text{RRCBVM} + T_{10} \cdot \text{RRCEVM} \\ &- T_{11} \cdot \text{VRCEVM} \end{aligned}$$

$$SV_2 = VVMBIN - T_6 \cdot RRCBVM + T_7 \cdot RRCEVM \\ + T_8 \cdot VRCEVM$$

$$SV_{3,0} = \infty$$

$$RSCEVM_0 = RRCEVM$$

$$VSCEVM_0 = VRCEVM$$

$$GVME_0 = GVMB + TRCI \cdot GVMB$$

$$QVME_0 = QVMB$$

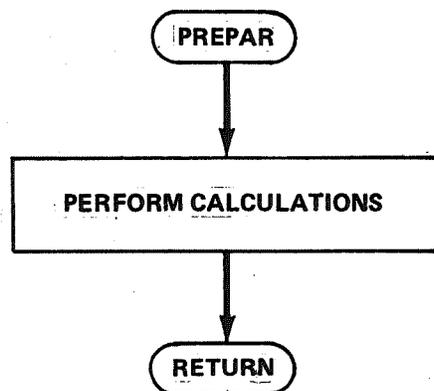
$$S_5 = TRCI^2 \div 20$$

$$S_4 = TRCI^2 \div 4$$

$$S_3 = TRCI^2 \div 12$$

$$S_2 = TRCI \cdot (.4)$$

$$S_1 = TRCI \cdot (.5)$$



VMASS\*RFJK.PREPAR

```

1      C      PREPAR      D NOVAK AND R JESSUP      BELLCOMM  04 FEB 69
2      C
3      C      PURPOSE
4      C
5      C      TO MAKE FIRST ESTIMATE OF FINAL STATE OF VIRTUAL MASS AND TO
6      C      COMPUTE FIXED QUANTITIES FOR THE ITERATIVE SOLUTION OF THE
7      C      PERTURBATION EQUATION
8      C
9      C      PREPARE FOR ITERATION OF INTERVAL COMPUTATIONS. ESTIMATE END OF
10     C      INTERVAL VIRTUAL MASS STATE. SOLVE VIRTUAL MASS PERTUBATION
11     C      EQUATIONS.
12     C
13     C      INPUT
14     C
15     C      RRCBVM      INITIAL REFERENCE CONIC POSITION RELATIVE TO THE V MASS
16     C      VRCBVM      INITIAL REFERENCE CONIC VELOCITY RELATIVE TO THE V MASS
17     C      RRCEVM      FINAL REFERENCE CONIC POSITION RELATIVE TO THE V MASS
18     C      VRCEVM      FINAL REFERENCE CONIC VELOCITY RELATIVE TO THE V MASS
19     C      RVMBIN      INITIAL VIRTUAL MASS INERTIAL POSITION
20     C      VVMBIN      INITIAL VIRTUAL MASS INERTIAL VELOCITY
21     C      GVMB        INITIAL VIRTUAL MASS MAGNITUDE
22     C      QVMB        INITIAL VIRTUAL MASS MAGNITUDE RATE
23     C      RRCAVM      INITIAL DISTANCE BETWEEN REF CONIC AND V MASS
24     C      RRCDVM      FINAL DISTANCE BETWEEN REF CONIC AND V MASS
25     C      TRCI        REF CONIC TIME INCREMENT FOR COMPUTING INTERVAL
26     C
27     C      OUTPUTS
28     C
29     C      ITRAT      ITERATION COUNTER (SET TO 0)
30     C      RVMEIN      FIRST ESTIMATE OF FINAL V MASS INERTIAL POSITION
31     C      VVMEIN      FIRST ESTIMATE OF FINAL V MASS INERTIAL VELOCITY
32     C      GVME        FINAL VIRTUAL MASS MAGNITUDE
33     C      QVME        FINAL VIRTUAL MASS MAGNITUDE RATE
34     C
35     C      SUBROUTINE PREPAR
36     C
37     C      INCLUDE QTYPE
38     C      INCLUDE QVAR
39     C      INCLUDE QFAC
40     C      INCLUDE QTEM
41     C      INCLUDE EVARS
42     C      INCLUDE EVARV
43     C
44     C      FIXED QUANTITIES FOR STEPIN HAVE S PREFIX
45     C
46     C      T1 = TRCI * DOT (VRCEVM, RRCEVM) / RRCDVM ** 2
47     C      T2 = TRCI ** 2
48     C      T3 = TRCI * T2
49     C      T4 = GVMB / RRCDVM ** 3
50     C      T5 = QVMB / RRCAVM ** 3
51     C      T6 = T2 * T5 / F12
52     C      T7 = TRCI * (F2 + T1) * T4 / F4
53     C      T8 = F1 - T2 * T4 / F12
54     C      T9 = T3 * T5 / F20
55     C      T10 = F1 + T2 * T4 * (FPT15 + T1 / F10)
56     C      T11 = T3 * T4 / F30

```

## VMASS SUBROUTINE PREPAR

```
57      DO 20 I = 1, 3
58          SV1(I) = RVMBIN(I) + TRCI * VVMBIN(I) - T9 * RRCBVM(I)
59          1      + T10 * RRCEVM(I) - T11 * VRCEVM(I)
60          SV2(I) = VVMBIN(I) - T6 * RRCBVM(I) + T7 * RRCEVM(I)
61          1      + T8 * VRCEVM(I)
62          RSCEVM(I) = RRCEVM(I)
63          VSCEVM(I) = VRCEVM(I)
64      20      SV3(I) = FINITY
65          GVME = GVMB + TRCI * QVMB
66          QVME = QVMB
67          S5 = T2 / F20
68          S4 = T2 / F4
69          S3 = T2 / F12
70          S2 = TRCI * FPT4
71          S1 = TRCI / F2
72      C
73      RETURN
74      END
```

# BELLCOMM, INC.

Program: PRTCOM

Purpose: Tabulate program COMMON by block name and subscript.

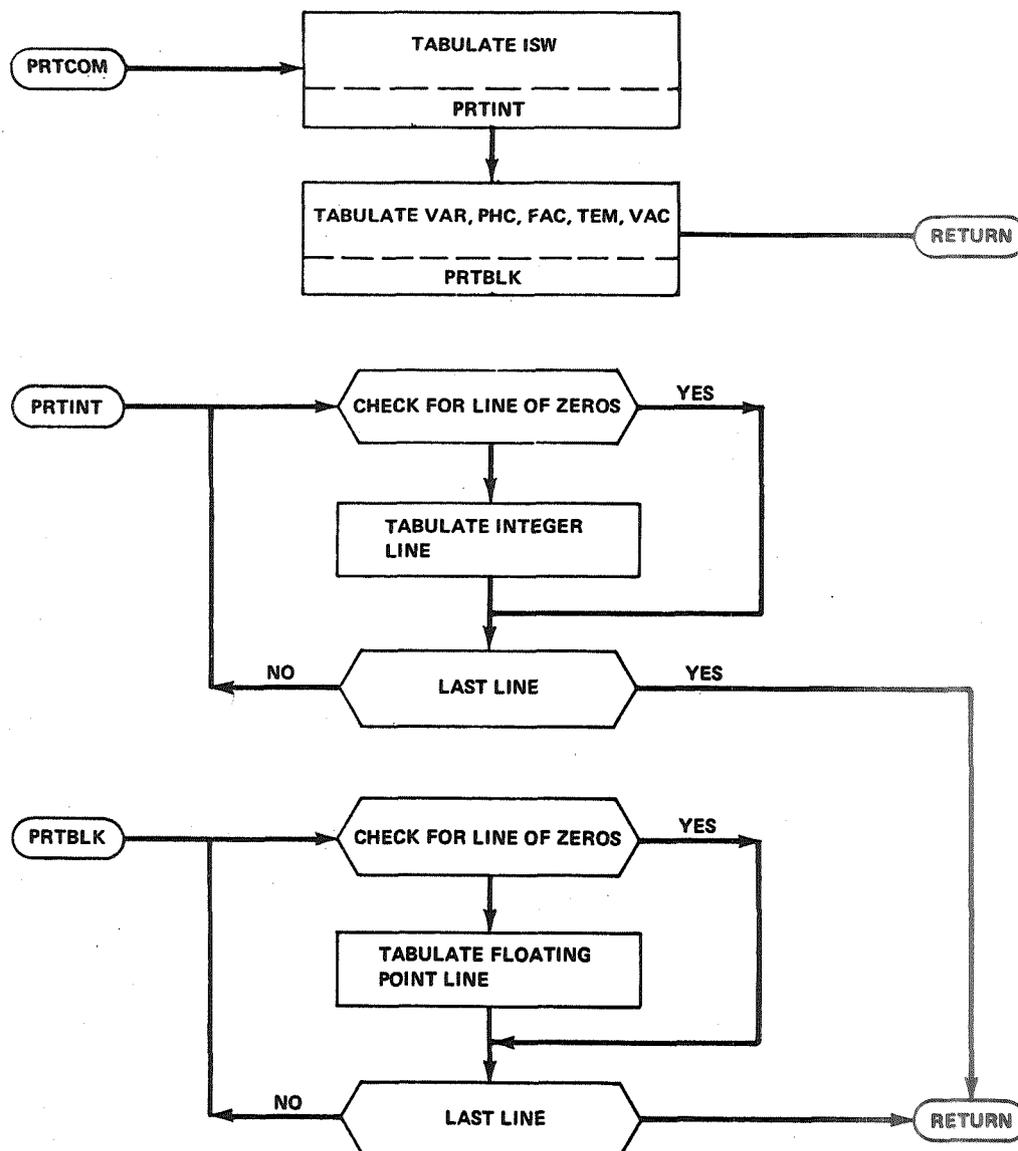
Called by: SETUP, SIMXXX

Calls: PRTBLK, PRTINT (internal subroutines)

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS\*RFJK.PRTCOM

```

1      C
2      C  PURPOSE
3      C
4      C    PROVIDE COMPACT FEATURE FOR PRINTING COMMON BLOCKS
5      C
6      C    SUBROUTINE PRTCOM
7      C
8      C  INPUT
9      C
10     C    COMMON BLOCKS (WITHOUT EQUIVALENCES)
11     C
12     C  OUTPUT
13     C
14     C    PRINT OF EACH BLOCK AS SUBSCRIPTED ARRAY
15     C
16     C    INCLUDE QTYPE
17     C    INCLUDE QVAR
18     C    INCLUDE QTEM
19     C    INCLUDE QPHC
20     C    INCLUDE QFAC
21     C    INCLUDE EVARI
22     C    INCLUDE EVARIO
23     C
24     C    WRITE (IOPRT, 100)
25     C    CALL PRTINT ('ISW', ISW, 300)
26     C    CALL PRTBLK ('VAR', VAR, 300)
27     C    CALL PRTBLK ('PHC', PHC, 50)
28     C    CALL PRTBLK ('FAC', FAC, 58)
29     C    CALL PRTBLK ('TEM', TEM, 100)
30     C    CALL PRTBLK ('VAC', VAC, 50)
31     C    RETURN
32     C
33     C 100  FORMAT (///' PRTCOM *** PRINT OF COMMON')
34     C
35     C    SUBROUTINE PRTBLK (NAME, BLOCK, NUMBER)
36     C
37     C    DOUBLE PRECISION BLOCK(NUMBER)
38     C
39     C    DO 30 I = 1, NUMBER, 4
40     C      DO 10 K = 0, 3
41     C        10  IF (BLOCK(I+K) .NE. F0) GO TO 20
42     C          GO TO 30
43     C        20  WRITE (IOPRT, 110) NAME, I, (BLOCK(I+K), K = 0, 3)
44     C        30  CONTINUE
45     C    WRITE (IOPRT, 130)
46     C    RETURN
47     C
48     C
49     C    ENTRY PRTINT (NAM, IBLOC, NUM)
50     C
51     C    INTEGER IBLOC(NUM)
52     C
53     C    DO 80 I = 1, NUM, 8
54     C      DO 60 K = 0, 7
55     C        60  IF (IBLOC(I+K) .NE. 0) GO TO 70
56     C          GO TO 80

```

```
57      70      WRITE (IOPRT, 120) NAM, I, (IBLOC(I+K), K = 0, 7),
58      1      (IBLOC(I+K), K = 0, 7), (IBLOC(I+K), K = 0, 7)
59      80      CONTINUE
60      WRITE (IOPRT, 130)
61      RETURN
62      C
63      C
64      110     FORMAT (A7, '(, I3, )', 4G30.18)
65      120     FORMAT (A7, '(, I3, )', 8O15/ 12X, 8A15/ 12X, 8I15)
66      130     FORMAT (///)
67      END
```

BELLCOMM, INC.

Program: PRTIME

Purpose: Print out various date-time groups

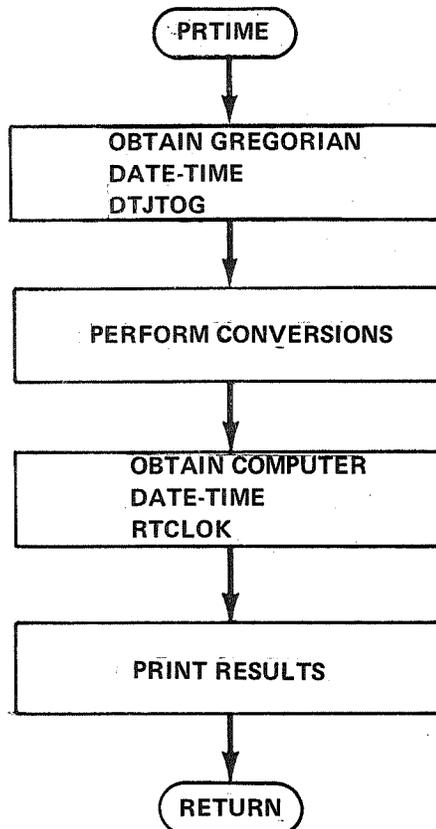
Called by: PRTXXX

Calls: DTJTOG, RTCLOK

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: Convert current trajectory time to calendar units.



VMASS\*RFJK.PRTIME

```

1      C      PRTIME      R JESSUP - BELLCOMM - 10 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      PRINT OUT DATE TIME GROUP
6      C
7      C      INPUT
8      C
9      C      TTRE      CURRENT TRAJECTORY TIME
10     C      TEPOJD    EPH DATE CURRENT AT FIRST EVENT (JULIAN DATE)
11     C      TEPOJS    EPH TIME CURRENT AT FIRST EVENT
12     C      TEPEJD    CURRENT EPH DATE
13     C      TEPEJS    CURRENT EPH TIME
14     C      IEVCUR    CURRENT EVENT
15     C
16     C      OUTPUT
17     C
18     C      IOPRT     DATE TIME STATEMENT (TRAJECTORY - EPHEMERIS - SIMULATOR)
19     C
20     C      SUBROUTINE PRTIME
21     C
22     C      INCLUDE QTYPE
23     C      INCLUDE QVAR
24     C      INCLUDE QFAC
25     C      INCLUDE QTEM
26     C      INCLUDE EVARI
27     C      INCLUDE EVARS
28     C      INCLUDE EVARIO
29     C
30     C      INCLUDE QMON
31     C      INTEGER IGREG(6), ITRAJ(4), ISTEIME(3)
32     C      EQUIVALENCE ( ITRAJ, ITRDA), (ITRAJ(2), ITRHR),
33     C      1 (ITRAJ(3), ITRMN), (ITRAJ(4), ITRSC)
34     C
35     C      T1      = TTRE * TUTODA
36     C      ITRDA = T1
37     C      T2      = ITRDA
38     C      ITRHR = (T1 - T2) * F24
39     C      T3      = ITRHR
40     C      ITRMN = (T1 - T2 - T3 / F24) * F1440
41     C      ITRSC = (T1 - T2 - T3 / F24 - ITRMN / F1440) * F86400
42     C      CALL DTJTOG (TEPEJD, TEPEJS, IGREG)
43     C      CALL RTCLOK (ISTEIME)
44     C
45     C      J = IGREG(2)
46     C      IGREG(2) = MONTH(J)
47     C      WRITE (IOPRT, 100) IEVCUR, ITRAJ, TEPEJD, TEPEJS, IGREG, ISTEIME
48     C      100 FORMAT (/ A7, ' AT', I5, ' DAYS', I3, ' HR', I3, ' MIN',
49     C      1 I3, ' SEC; JULIAN', F10.1, ' DAYS', G12.6, ' SEC (',
50     C      2 I4, 1X, A3, 2I3, 2(':', I2), '); COMPUTED ', 3A6//)
51     C      RETURN
52     C      END

```

## BELLCOMM, INC.

Program: PRTRAJ

Purpose: Process a trajectory stored on the "print later" file, IOPRL, to obtain extra output in the event interval (IEVSTR, IEVSTP).

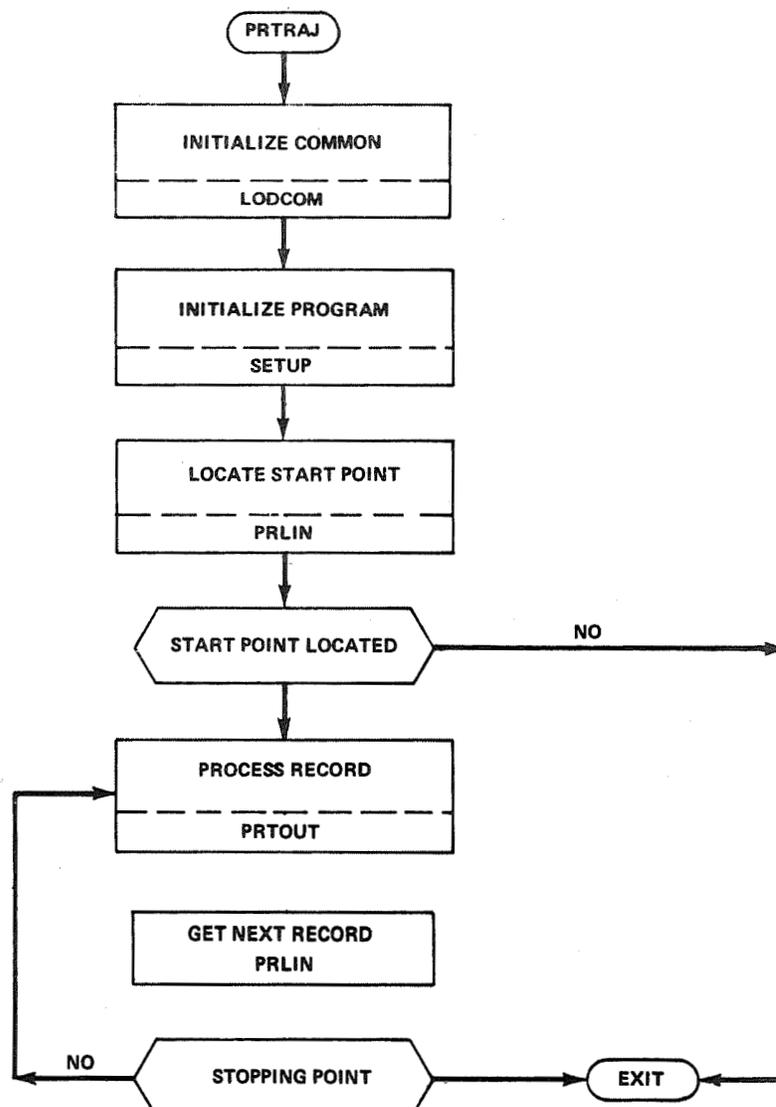
Called by: Separate computer execution

Calls: LODCOM, SETUP, PRLIN, PRTVAR, PRTOUT

Errors and Diagnostics: A diagnostic message is given whenever the event interval or the requested trajectory, as given by the trajectory identification, IDTRAJ, cannot be found on the tape.

Language: Fortran main program

Computations: None



VMASS\*RFJK.PRTRAJ

```

1      C          PRTRAJ          R JESSUP          BELLCOMM          20 JUN 69
2      C
3      C  PURPOSE
4      C
5      C          PROCESS A TRAJECTORY STORED ON A SECONDARY FILE FOR EXTRA OUTPUT
6      C          IN THE EVENT INTERVAL (IEVSTR, IEVSTP)
7      C
8      C  INPUT
9      C
10     C          IDTRAJ  IDENTIFICATION OF TRAJECTORY TO BE PROCESSED
11     C          IEVSTR  EXTRA OUTPUT STARTING EVENT
12     C          IOPRL   THE PROCESS LATER FILE
13     C          IEVSTP  EXTRA OUTPUT STOPPING EVENT
14     C
15     C  OUTPUT
16     C
17     C          AS PROGRAMMED IN THE PRINT SUBROUTINE PRTXXX
18     C
19     C          INCLUDE QTYPE
20     C          INCLUDE QVAR
21     C          INCLUDE EVARL
22     C          INCLUDE EVARIO
23     C          INCLUDE EVARI
24     C
25     C  2  LLDCOM = .TRUE.
26     C
27     C          CALL LODCOM
28     C          REWIND IOPRL
29     C  5  CALL SETUP
30     C
31     C          SEARCH FOR TRAJECTORY AND STARTING EVENT (START AT BEGINNING IF
32     C          NO STARTING EVENT IS GIVEN)
33     C
34     C          I1 = IEVSTR
35     C          I2 = IEVSTP
36     C  10  CALL PRLIN
37     C          IF (IEVCUR .EQ. -1) GO TO 50
38     C          IF (IDTRAJ(1) .EQ. 0) GO TO 30
39     C          DO 12 I = 1, 3
40     C  12          IF (IDTRAJ(I) .NE. IRTIME(I)) GO TO 14
41     C                  GO TO 22
42     C  14          WRITE (IOMES, 100)
43     C                  CALL PRTVAR
44     C                  GO TO 10
45     C  22          IF (I1 .EQ. 0) GO TO 30
46     C                  IF (I1 .EQ. IEVCUR) GO TO 30
47     C                  GO TO 5
48     C
49     C          PROCESS INFORMATION THROUGH SUBROUTINE PRTXXX
50     C
51     C  30  CALL PRTOUT
52     C          CALL PRLIN
53     C          IF (IEVCUR .EQ. -1) GO TO 50
54     C
55     C          TERMINATE AT STOPPING EVENT (OR AT END OF TRAJECTORY IF NO
56     C          STOPPING EVENT IS GIVEN)

```

```
57      C
58      IF (IEVFLG .EQ. I2 .AND. I2 .NE. IEVCUR) GO TO 5
59          GO TO 30
60      50  PRINT 130
61          GO TO 10
62      100 FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED TRAJECTORY ')
63      110 FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED START EVENT ')
64      120 FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED STOP EVENT ')
65      130 FORMAT (' PRTRAJ *** END-OF-FILE ')
66      END
```

Program: PRTXXX (element name)

PRTOUT

Purpose: Provide structural element for user coded trajectory print statements.

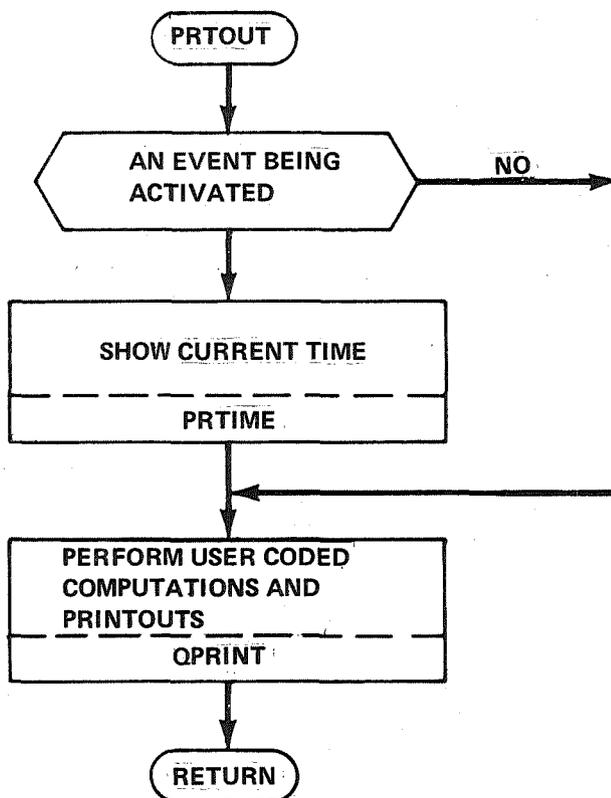
Called by: RECORD, PRTRAJ

Calls: PRTIME

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: As coded by user in element QPRINT.



VMASS\*RFJK.PRTXXX

```
1      C      R JESSUP - BELLCOMM - 3 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      PRESENT THE COMPUTED TRAJECTORY VIA THE SYSTEM PRINTER
6      C
7      C      INPUTS
8      C
9      C      ALL FORMALLY NAMED PROGRAM VARIABLES
10     C
11     C      OUTPUTS
12     C
13     C              AS PROGRAMMED
14     C
15     C      SUBROUTINE PRTOUT
16     C
17     C
18     C      INCLUDE QTYPE
19     C      INCLUDE QVAR
20     C      INCLUDE EVARI
21     C      INCLUDE EVARL
22     C      INCLUDE EVARV
23     C      INCLUDE EVARS
24     C      INCLUDE EVARIO
25     C      INCLUDE QSTORE
26     C
27     C      IF (LXECEV) CALL PRTIME
28     C
29     C      INCLUDE OPRINT, LIST
30     C
31     C      RETURN
32     C      END
```

Program: PRTVAR

PRTEPH

PRTFAC

PRTPHC

PRTTEM

Purpose: Tabulate program COMMON by symbol name using the Fortran name list feature.

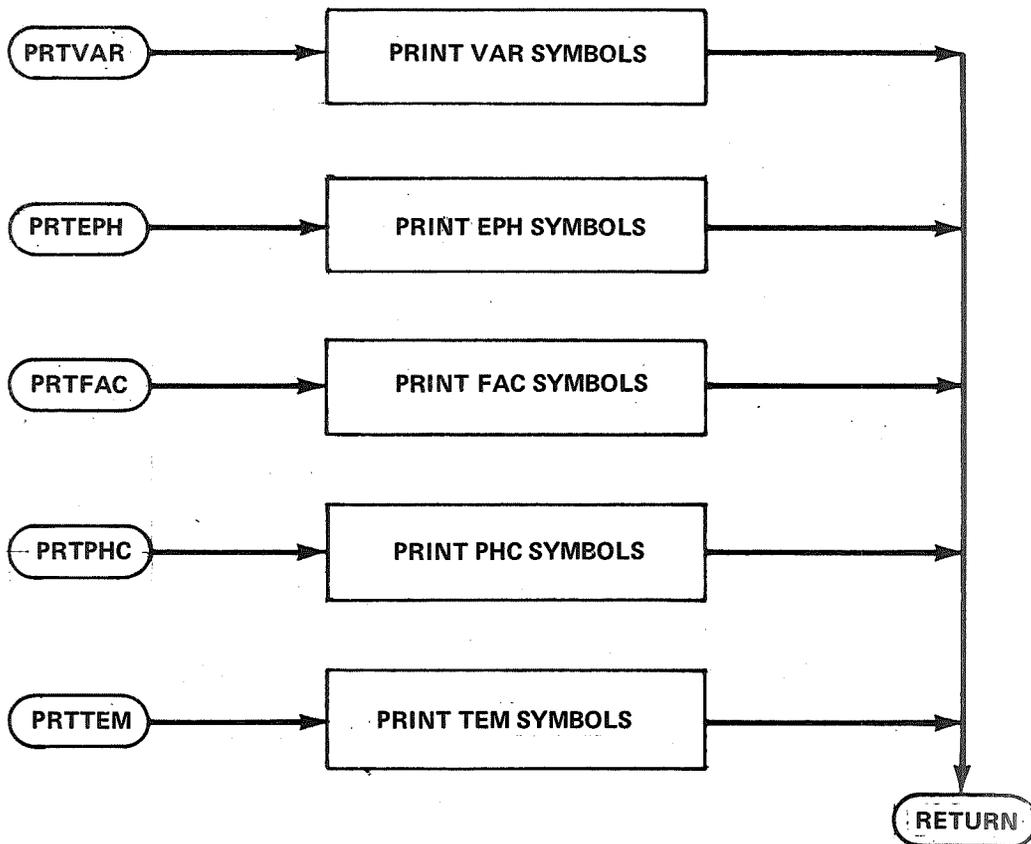
Called by: SIMXXX

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS\*RFJK.PRTVAR

```

1      C
2      C  PURPOSE
3      C
4      C    TABULATE COMMON BLOCKS BY NAME
5      C
6      C    SUBROUTINE PRTVAR
7      C
8      C    INCLUDE QVAR, LIST
9      C
10     C    INCLUDE QTYPE
11     C    INCLUDE QEPH
12     C    INCLUDE QIEV
13     C    INCLUDE QREADE
14     C    INCLUDE EEPH
15     C    INCLUDE EVARI
16     C    INCLUDE EVARL
17     C    INCLUDE EVARS
18     C    INCLUDE EVARV
19     C    INCLUDE EVARIO
20     C
21     C    NAME LIST / VARS /
22     1  TTRB,  TTRE,          TEPEJD, TEPEJS, TEPOJD, TEPOJS,
23     2          DTAI,  DTAIDN, DTAIMN, DTAIUP,          ERCAVM,
24     3          GVMB,  GVME,          HRCAVM,
25     4  QVMB , QVME ,
26     5  RRCAVM, RRCDVM, RSCAVM, RSCDVM,
27     6  RTOPCM, RTOPRF,          RUTOAU,          TTO,
28     7  TUTODA,          VUTOAD
29     C
30     C    NAME LIST / VARV /
31     1  ERCBVM, HRCBVM,
32     2  RRCIVM, RRCBVM, VRCBVM, RRCEVM, VRCEVM,
33     3  RSCBIN, VSCBIN, RSCVIN, VSCVIN, RSCDDK, VSCDDK,
34     4  RSCBVM, VSCBVM, RSCVEM, VSCVEM, RVMBIN, VVMBIN, RVMEIN, VVMEIN,
35     C
36     C    NAME LIST / VARI /
37     1  IEVN , IEVSTP, IEVSTR, INCNTR,
38     2  IOCDR, IOEPH, IOMES, IOpCH, IOPRL, IOPRT, IOROL,
39     3  IPLATM, IPLOTM, IPNOWM, ITRAT, JKN, KEPMAX,
40     4  KEARTH, KEMB , KJUP , KMARS , KMERC , KMOON , KNEP , KPLUTO,
41     5  KSAT , KSUN , KURAN , KVAR , MVAR , NMAX ,
42     C    NAME LIST / VARL /
43     5  LIMPEV, LINTDN, LLDCOM, LOGOUT, LPRBKP, LPRCOM, LPREPH,
44     6  LPRFAC, LPRLAT, LPRNOW, LPRPHC, LPRVAR, LRESKP, LSTDDK, LXECEV,
45     C
46     C    WRITE (IOPRT, VARI)
47     C    WRITE (IOPRT, VARL)
48     C    WRITE (IOPRT, VARS)
49     C    WRITE (IOPRT, VARV)
50     C    WRITE (IOPRT, 25) IDTRAJ, IRTIME, IEVREP, IEVRUN, IEVCUR,
51     1  IEVSTR, IEVSTP
52     25  FORMAT (' IDTRAJ ', 3A6, ' ', IRTIME ' ', 3A6, ' ', IEVREP', A7, '- ',
53     1  A6, ' ', IEVRUN', A7, '- ', A6, ' ',
54     2  ' IEVCUR', A7, ' ', IEVSTR', A7, ' ', IEVSTP', A7)
55     C    WRITE (IOPRT, 30) ((IEV(J,I)', J = 3, 7), EVN(1,I), I = 1, IEVN)
56     30  FORMAT (' EVENTS TABLE' / (2A8, 3I8, G26.18))

```

```
57         RETURN
58         C
59         C
60         C     PRINT BLOCK EPH BY NAME
61         C
62         C     ENTRY PRTEPH
63         C
64         C     NAME LIST / RDEPRT / AU, RE, TPD, EMRAT, ICW, ICENT, IREQ, JK
65         C
66         C     WRITE (IOPRT, RDEPRT)
67         C     WRITE (IOPRT, 152)
68         C     DO 150 I = 1, JKN
69         C         K = JK(I)
70         C     150   WRITE (IOPRT, 154) K, (EPH(J,K), J = 1, JKSPEC),
71         C           1   (EPH(J,K), J = 17, 19)
72         C     152   FORMAT ('0GRAVITATING POINT DESCRIPTIONS')
73         C     154   FORMAT (I5, ' ROOEIN', G30.18, 2G36.18/ 6X, 'V00EIN', G30.18, 2G36.18/
74         C           1   6X, 'RSCE00', G30.18, 2G36.18/ 6X, 'VSCE00', G30.18, 2G36.18/
75         C           2   6X, 'RSCA00', G30.18, ' RSCD00', G29.18, ' GOO  ', G29.18/
76         C           3   6X, 'G00R  ', G30.18, ' DANEO0', G29.18, ' DOBE00', G29.18/
77         C           4   6X, 'DLNE00', G30.18, ' SPLE00', G29.18, ' ROOD  ', G29.18/
78         C           5   6X, 'RMNE00', G30.18, 2G36.18//)
79         C     160   CONTINUE
80         C
81         C     LPREPH = .FALSE.
82         C     RETURN
83         C     END
```

VMASS\*TPF\$.PRTFAC

```

1      C
2      C  PURPOSE
3      C
4      C    TABULATE COMMON BLOCKS BY NAME
5      C
6      C    SUBROUTINE PRTFAC
7      C
8      C    INCLUDE QVAR, LIST
9      C    INCLUDE QTYPE
10     C    INCLUDE QIEV
11     C    INCLUDE QPHC
12     C    INCLUDE QTEM
13     C    INCLUDE QFAC
14     C    INCLUDE EVARI
15     C    INCLUDE EVARL
16     C    INCLUDE EVARIO
17     C    INCLUDE ETEM
18     C
19     C    NAME LIST / FACPRT /      FINITY, FINTES, FPT15 , FPT4 , FPT5 ,
20     1  FRT0 , FTT0 , FTOKM, F0 , F0001, F001 , F1 , F1PT1 ,
21     2  F1PT5 , F10 , F11 , F12 , F13 , F14 , F1440 , F15 ,
22     3  F2 , F2PT5 , F20 , F24 , F3 , F30 , F3600 , F4 ,
23     4  F43200, F5 , F6 , F60 , F7 , F8 , F86400, F9 ,
24     5  F9TNT, DTOR , RETOKM, RTOD , AUTOKM
25     C    WRITE (IOPRT, FACPRT)
26     C    LPRFAC = .TRUE.
27     C    RETURN
28     C
29     C
30     C    PRINT BLOCK PHC BY NAME
31     C
32     C    ENTRY PRTPHC
33     C
34     C
35     C    NAME LIST / PHCPRT / GS, PIMASS
36     C    WRITE (IOPRT, PHCPRT)
37     C    LPRPHC = .FALSE.
38     C    RETURN
39     C
40     C
41     C    PRINT BLOCK TEM BY NAME
42     C
43     C    ENTRY PRITEM
44     C
45     C
46     C    NAME LIST / TEMPRT /      TV1, TV2, TV3, TV4, TV5,
47     1  TV6, TV7, TV8, TV9, TV10, TV11, SV1, SV2,
48     2  SV3, SV4, SV5, SV6, XV1, XV2, XV3, XV4,
49     3  XV5, XV6, T1, T2, T3, T4, T5, T6,
50     4  T7, T8, T9, T10, T11, S1, S2, S3,
51     5  S4, S5, S6, S7, S8, S9, X1, X2,
52     6  X3, X4, X5, X6, X7, X8, X9
53     C    WRITE (IOPRT, TEMPRT)
54     C    LPRTEM = .FALSE.
55     C    RETURN
56     C    END

```

Program: READE (JED, TSEC, IERR)

Purpose: Obtain selenocentric positions and velocities of planets at current time.

Called by: EPHEM

Calls: GETTAP

Errors and Diagnostics: Gives error return on invalid central body, output code, or current time out of range of assigned ephemeris tape.

Language: Fortran subroutine

Computations: Interpolate as necessary when current time lies between ephemeris tape records.

Flowchart: (Omitted: This program was furnished by JPL)

## BELLCOMM, INC.

Program: RECORD

Purpose: Control trajectory recording on the on-line printer or print later file.

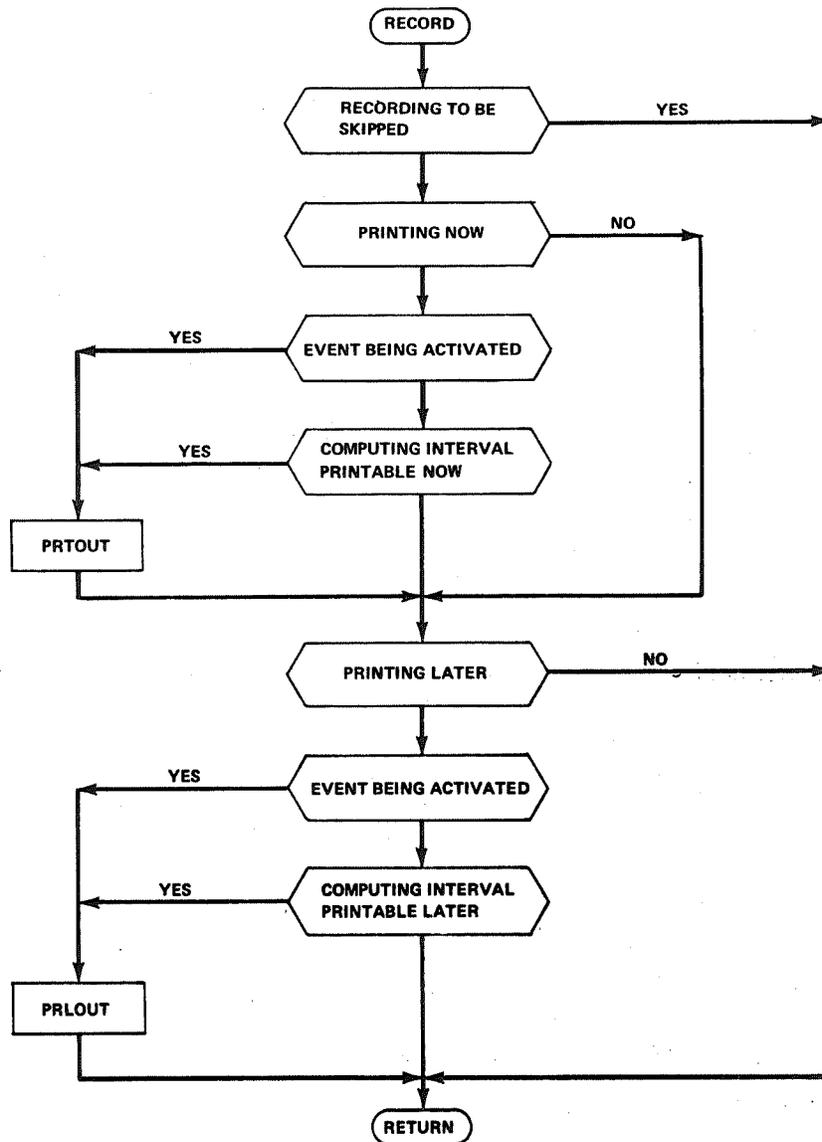
Called by: EVNACC, SIMXXX

Calls: PRLOUT, PRTOUT

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS\*RFJK.RECORD

```

1      C      RECORD   R JESSUP - BELLCOMM - 06 JAN 69
2      C
3      C      PURPOSE
4      C
5      C
6      C      MAKE RECORD OF THE RUN
7      C
8      C      INPUTS
9      C
10     C      LRESKP   RECORDING BEING SKIPPED (DURING TARGETING FOR EXAMPLE)
11     C      LPRNOW   PRINTING (VIA PRINT-LIST) OCCURRING DURING RUN
12     C      LPRLAT   PRINTING (VIA PRINT LIST) TO OCCUR LATER
13     C      IPLATM   INCREMENTAL COMPUTING STEPS PER PRINT LATER FILE WRITE
14     C      IPNOWM   INCREMENTAL COMPUTING STEPS PER CONCURRENT PRINTOUT CALL
15     C
16     C      OUTPUT
17     C
18     C      IOPRL    (PRLOUT) THE V MASS PROGRAM VARIABLES BLOCK - KVAR
19     C      IOPRT    (PRTOUT) EDITED PRINT OF THE PROGRAM RUN
20     C      INCNTR   INCREMENTAL COMPUTING STEPS COUNTER
21     C
22     C      SUBROUTINE RECORD
23     C
24     C      INCLUDE QVAR
25     C      INCLUDE EVARI
26     C      INCLUDE QTYPE
27     C      INCLUDE EVARL
28     C
29     C      IF (LRESKP) RETURN
30     C      IF ((LXECEV .OR. MOD (INCNTR, IPNOWM) .EQ. 0) .AND. LPRNOW)
31     1  CALL PRTOUT
32     C      IF ((LXECEV .OR. MOD (INCNTR, IPLATM) .EQ. 0) .AND. LPRLAT)
33     1  CALL PRLOUT
34     C
35     C
36     C      RETURN
37     C      END

```

Program: ROLLIO (name of element)

ROLOUT

PRLOUT

ROLBCK (i)

PRLIN

Purpose: Transfer program state as represented by the array KVAR from internal to secondary storage or vice versa.

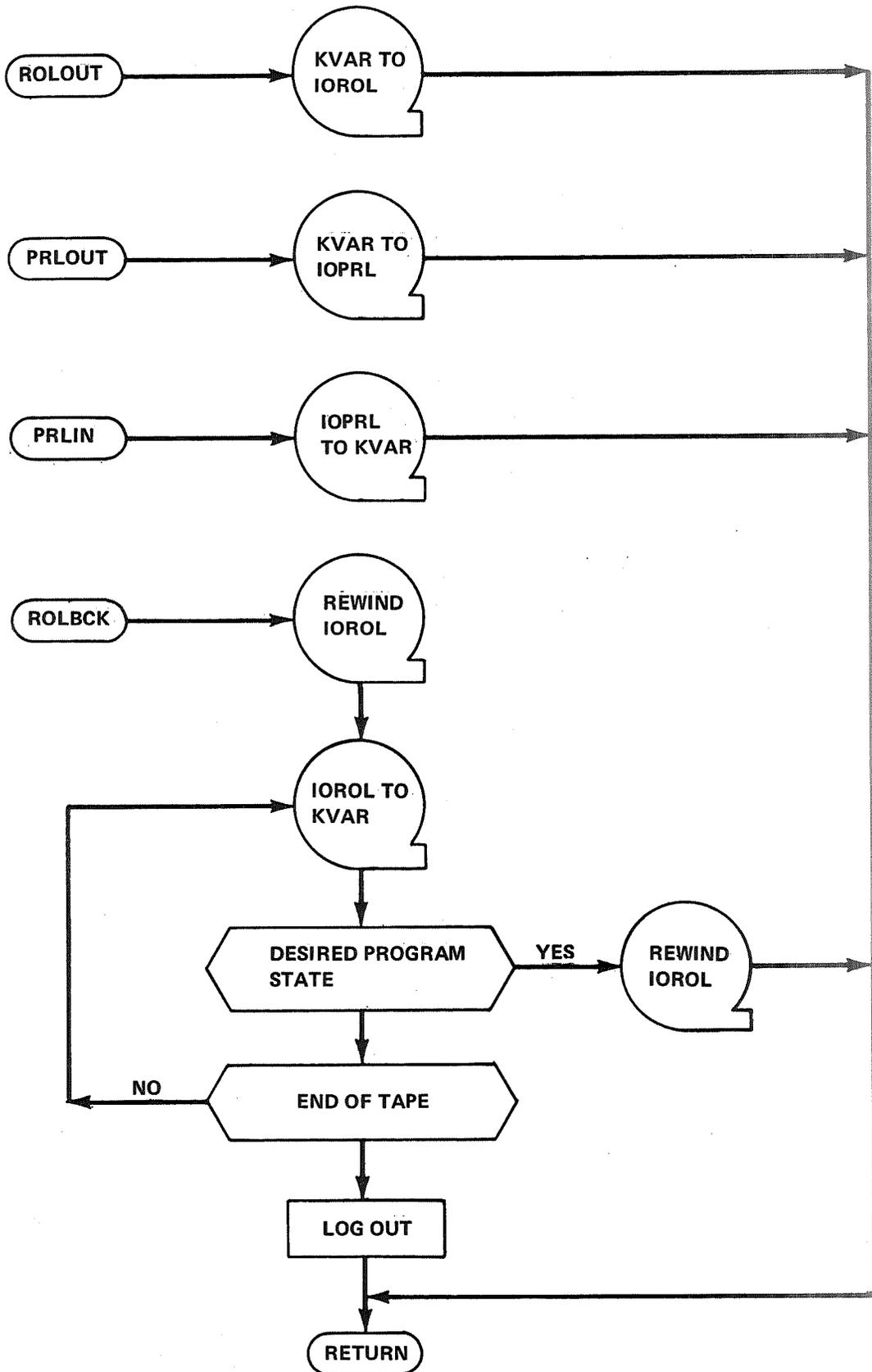
Called by: ROLOUT: EVNACC; ROLBCK; SETUP; PRLIN: PRTRAJ;  
PRLOUT: RECORD, SETUP

Calls: None

Errors and Diagnostics: Program log out is initiated on inability to find roll back event state.

Language: Fortran subroutine

Computations: None



VMASS\*RFJK.ROLLIO

```

1      C      ROLLIO   R JESSUP - BELLCOMM - 06 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      IN THIS PROGRAM 'ROLOUT' AND 'PRLOUT' PLACE CURRENT V MASS SYSTEM
6      C      STATE 'KVAR' ON SECONDARY FILES IOROL AND IOPRL, RESPECTIVELY.
7      C      'ROLBCK' AND 'PRLIN' READ SUCH A STATE BACK INTO THE COMPUTER.
8      C
9      C      INPUTS
10     C
11     C      IOPRL      (PRLIN) KVAR
12     C      IOROL      (ROLIN) KVAR
13     C
14     C      OUTPUT
15     C
16     C      IOPRL      (PRLOUT) KVAR AT EVENTS AND SELECTED COMPUTING INCREMEN
17     C                        BETWEEN EVENTS
18     C      IOROL      (ROLOUT) KVAR AT EVENTS
19     C
20     C
21     C      SUBROUTINE ROLOUT
22     C
23     C      INCLUDE QVAR
24     C      INCLUDE QTYPE
25     C      INCLUDE EVARI
26     C      INCLUDE EVARL
27     C      INCLUDE EVARIO
28     C
29     C      WRITE (IOROL) KVAR
30     C      RETURN
31     C
32     C
33     C      ENTRY PRLOUT
34     C      WRITE (IOPRL) KVAR
35     C      RETURN
36     C
37     C
38     C      ENTRY ROLBCK (I1)
39     C      I2 = I1
40     C      I3 = IEVCUR
41     C      REWIND IOROL
42     C      10  READ (IOROL) KVAR
43     C          IF (IEVCUR .NE. I2) GO TO 20
44     C          REWIND IOROL
45     C          RETURN
46     C      20  IF (IEVCUR .NE. I3) GO TO 10
47     C          WRITE (IOPRL, 30)
48     C          LOGOUT = .TRUE.
49     C          RETURN
50     C      30  FORMAT (' ROLBCK *** REQUESTED EVENT NOT ON ROLL BACK FILE')
51     C
52     C
53     C      ENTRY PRLIN
54     C      READ (IOPRL) KVAR
55     C      RETURN
56     C      END

```

Program: RTCLOCK

RTCLOCK (a)

Purpose: Obtain current value of computer real time clock

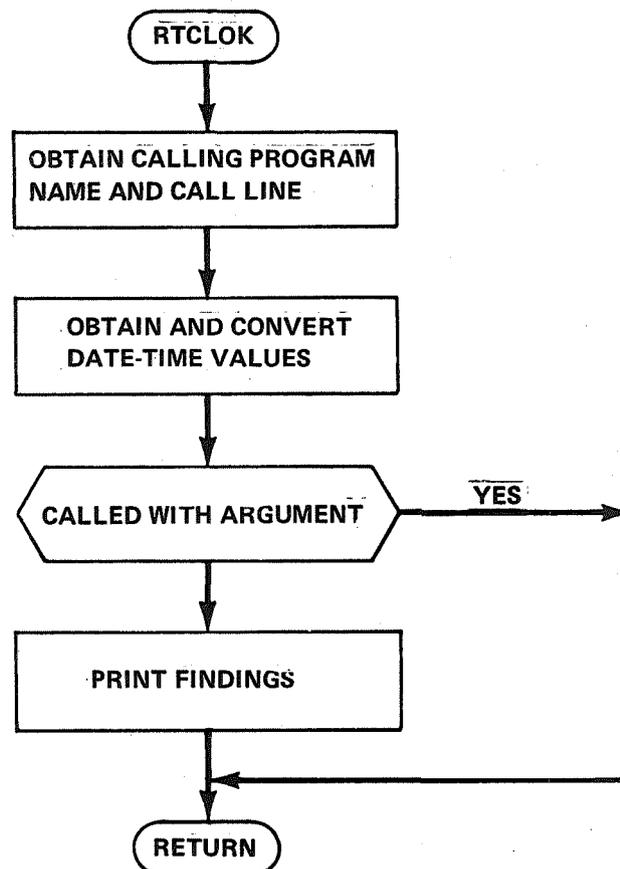
Called by: PRTIME, SETUP

Calls: None

Errors and Diagnostics: None

Language: 1108 Assembler subroutine

Computations: Convert clock readings to printable form.



VMASS\*RFJK.RTCLOK

```

1      .
2      .  PURPOSE
3      .
4      .    OBTAIN CURRENT READING OF THE COMPUTER 'REAL TIME' CLOCK
5      .
6      .  CALLING SEQUENCE
7      .
8      .  FOR PRINT-OUT INCLUDING CALLING PROGRAM NAME AND CALL LOCATION
9      .    CALL RTCLOK
10     .
11     .  FOR 18 CHARACTER RETURN VIA A IN THE FORM 'DA MON YR HR:MI:SE'
12     .    CALL RTCLOK (A)
13     .
14     $(1)  LIT
15     .      +0.  +1.  +2.  +3.  +4.  +5.  +6.  +7.  +8.  +9
16     MSG   'RTCLOK *** PRGNAM LINE NUMBER      DA MON YR HR:MI:SE.ONDS '
17     .
18     $(2)
19     RTCLOK* EQU      $
20     S      B11, ('SAVB11,')
21
22     L      A1,0,B11      .  INSERT LINE NUMBER CONVERTED TO FIELDATA
23     L,XM   A2,5
24     LSSL  A0,3
25     LDSL  A0,3
26     A,XM  A0,060
27     JGD   A2,$-3
28     S     A0,MSG+4
29     SZ    A0      .  INSERT PROGRAM NAME (000000 FOR MAIN PR)
30     SZ    MSG+2
31     TNE,H2 A0,0,B11
32     J     $+3
33     L     A0,*0,B11
34     S     A0,MSG+2
35     ER    DATE$
36     L     A2,(0530000530000)      .  :MI:SE
37     DSL   A1,12
38     SSC   A2,6
39     DSL   A1,12
40     SSC   A2,6
41     S     A2,MSG+8
42     L     A2,(0050000050000)      .  YR HR
43     DSL   A1,12
44     SSC   A2,6
45     DSL   A0,24
46     SSL   A1,12
47     DSL   A1,12
48     SSC   A2,6
49     S     A2,MSG+7
50     L     B11,(1,0)
51     TE,H2 A0,MON,*B11
52     J     $-1
53     L     A2,MON-1,B11
54     L,XM  A3,5
55     LDSC  A2,66
56     DSL   A1,12

```

```

57          S      A2,MSG+6
58          L      B11,('SAVB11,')
59          TNZ,H1  0,B11
60          J      ONE
61          P$RINT MSG,8,1      . PRINT COMPOSED MESSAGE AND RETURN
62          L      B11,('SAVB11,')
63          J      1,B11
64          .
65          ONE   L      A0,0,B11
66          L      A1,MSG+6
67          S      A1,0,A0
68          DL     A1,MSG+7
69          DS     A1,1,A0
70          J      2,B11
71          $(3)
72          CMON  INFO  2 3      . COMMON / CMON / MON(12)
73          MON  RES   12
74          .
75          A0   EQU   014
76          A1   EQU   A0+1
77          A2   EQU   A0+2
78          A3   EQU   A0+3
79          B11  EQU   013
80          C3   EQU   013
81          H1   EQU   02
82          H2   EQU   01
83          RU   EQU   0100
84          XM   EQU   017
85          .
86          END

```

Program: SCALE (s, u, v)

Purpose: Form product of a scalar and a vector

Called by: None

Calls: None

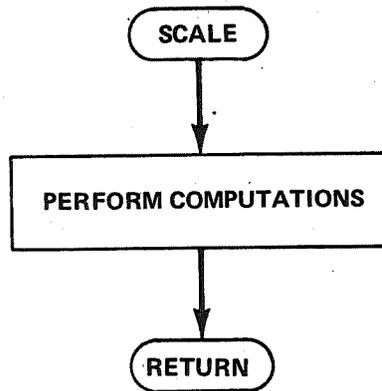
Errors and Diagnostics: None

Language: Fortran subroutine

Computations: For the vectors u and v compute

$$v_i = s u_i$$

for  $i = 1, 2, 3$ .



VMASS\*RFJK.SCALE

```
1      C
2      C  PURPOSE
3      C
4      C    FORM PRODUCT OF A SCALAR AND A VECTOR (3 - COMPONENTS)
5      C
6      C  INPUT
7      C
8      C    SCAL    SCALAR
9      C    VEC1    VECTOR INPUT
10     C
11     C  OUTPUT
12     C
13     C    VEC2    VECTOR RESULT
14     C
15     C    SUBROUTINE SCALE (SCAL, VEC1, VEC2)
16     C
17     C    INCLUDE QTYPE
18     C    DIMENSION VEC1(3), VEC2(3)
19     C
20     C    DO 10 I = 1, 3
21     10  VEC2(I) = SCAL * VEC1(I)
22     C    RETURN
23     C    END
```

Program: SETUP

Purpose: Read the data deck and perform program initialization and termination functions.

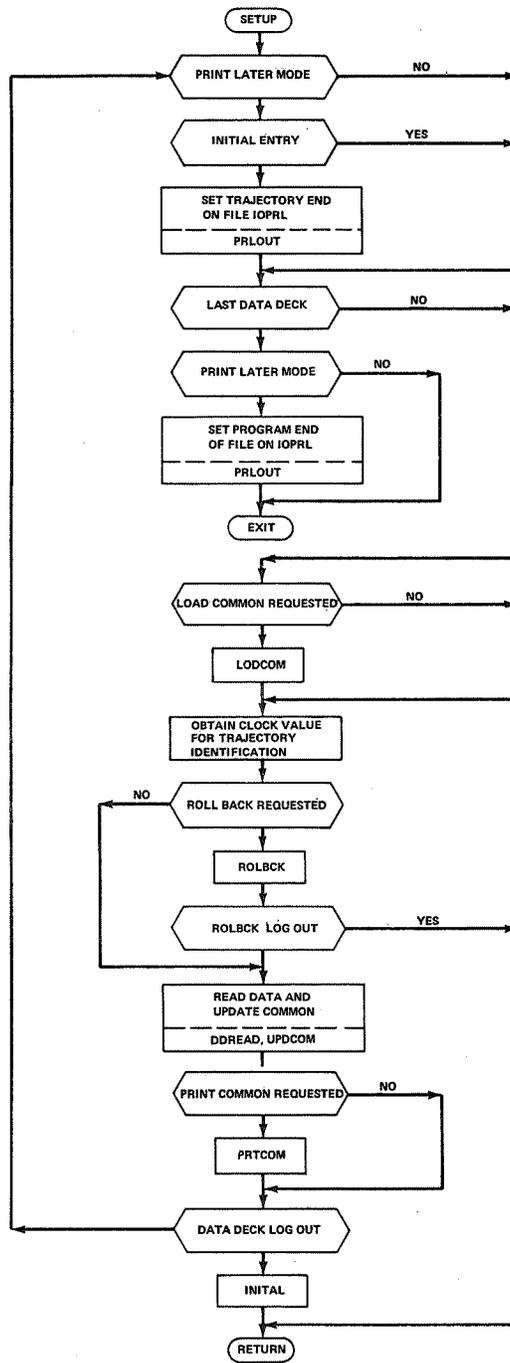
Called by: SIMXXX, PRTRAJ

Calls: LODCOM, PRLOUT, RTCLOK, DDREAD, UPDCOM, PRTCOM, INITAL, EXIT

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



## VMASS\*RFJK.SETUP

```

1      C
2      C  PURPOSE
3      C
4      C    THIS PROGRAM READS THE INPUT 'DATA DECK' AND SETS UP THE
5      C    PROGRAM FOR A COMPUTER RUN.
6      C
7      C  INPUTS
8      C
9      C    THE 'DATA DECK' VIA FORTRAN I0CDR (USUALLY THE ON-LINE
10     C    CARD READER)
11     C    IDTRAJ  IDENT FOR PREVIOUSLY DET TRAJ TO BE PRINTED BY THIS RUN
12     C
13     C  OUTPUTS
14     C
15     C    IRTIME  REAL TIME CLOCK READING AS DA MON YR HR:MI:SE FOR TRAJ ID
16     C
17     C  SUBROUTINE SETUP
18     C
19     C    INCLUDE QTYPE
20     C    INCLUDE QVAR
21     C    INCLUDE EVARI
22     C    INCLUDE EVARL
23     C    INCLUDE EVARIO
24     C
25     C    FOR THE PRINT LATER OPTION (LPRLAT TRUE) PROVIDE SOFTWARE TERMINAL
26     C    RECORD (IEVCUR 0) FOR THE LAST EVENT OF EACH TRAJECTORY SEGMENT.
27     C    MVAR 0 INDICATES TRAJECTORY BEGINNING.
28     C
29     C  5  IF (.NOT. LPRLAT) GO TO 7
30     C    IF (MVAR .EQ. 0) GO TO 7
31     C    IEVCUR = 0
32     C    CALL PRLOUT
33     C
34     C    STOP PROGRAM IF ALL DATA DECKS HAVE BEEN PROCESSED (LSTDDK TRUE)
35     C
36     C  7  IF (.NOT. LSTDDK) GO TO 9
37     C    IF (LPRLAT) CALL PRLOUT
38     C    CALL EXIT
39     C
40     C    PRELOAD COMMON FROM STORAGE (LLDCOM TRUE). THE CALLING PROGRAM,
41     C    USUALLY THE EVENTS LIST, SETS LLDCOM TRUE ON INITIAL CALL TO SETUP
42     C    LLDCOM, SET FALSE BY LODCOM, REMAINS FALSE THROUGH SUBSEQUENT
43     C    INITIALIZATIONS UNLESS CHANGED BY THE VMASS USER.
44     C
45     C  9  IF (LLDCOM) CALL LODCOM
46     C
47     C    OBTAIN COMPUTER SYSTEM DATE-TIME GROUP FOR 18 CHARACTER IDENTIFI-
48     C    CATION (IRTIME) OF THE TRAJECTORY SEGMENT TO BE CALCULATED. SEE
49     C    ALSO IDTRAJ IN THE PRINT LATER PROGRAM PRTRAJ.
50     C
51     C    CALL RTCLOK (IRTIME)
52     C    WRITE (IOPRT, 20) IRTIME
53     C  20  FORMAT ('VIRTUAL MASS TRAJECTORY ', 3A6)
54     C
55     C    ON INDICATION IEVREP NOT 0 RELOAD COMMON TO VALUES HELD ON THE
56     C    PROGRAM ROLL BACK FILE FOR THE EVENT GIVEN BY IEVREP. THE ROLL

```

```
57      C      BACK FILE (IOROL) TRAJECTORY SEGMENTS MAY BE FROM THE CURRENT OR
58      C      SOME PREVIOUS COMPUTER RUN.
59      C
60      IF (IEVREP(1) .EQ. 0) GO TO 30
61      CALL ROLBCK (IEVREP)
62      IF (.NOT. LOGOUT) GO TO 30
63      WRITE (IOMES, 22)
64      22      FORMAT (' ROLBCK CALLED BY SETUP')
65      RETURN
66      C
67      C
68      30      CALL DDREAD
69      CALL UPDCOM
70      C
71      C      LPRNOW SET TRUE IN THE DATA DECK CAUSES PRINT OF COMMON AFTER
72      C      INPUT READING. ROLL BACK OR PRELOADING OF COMMON CAN BE CONTROLLED
73      C      FROM THE DATA DECK BY THE APPROPRIATE INDICATORS PLUS LOGOUT
74      C      TRUE.
75      C
76      IF (LPRCOM) CALL PRTCOM
77      IF (.NOT. LOGOUT) GO TO 100
78      WRITE (IOMES, 40)
79      40      FORMAT (' DDREAD CALLED BY SETUP')
80      LOGOUT = .FALSE.
81      GO TO 5
82      C
83      C      PROGRAM USER CAN EXTEND SETUP BY WRITING HIS OWN INITIAL ROUTINE
84      C
85      100     CALL INITAL
86      RETURN
87      END
```

Program: SIMXXX

Purpose: This program controls spacecraft state vector propagation from a starting to a stopping event. It is designed to contain user coded events descriptions and events associated calculations.

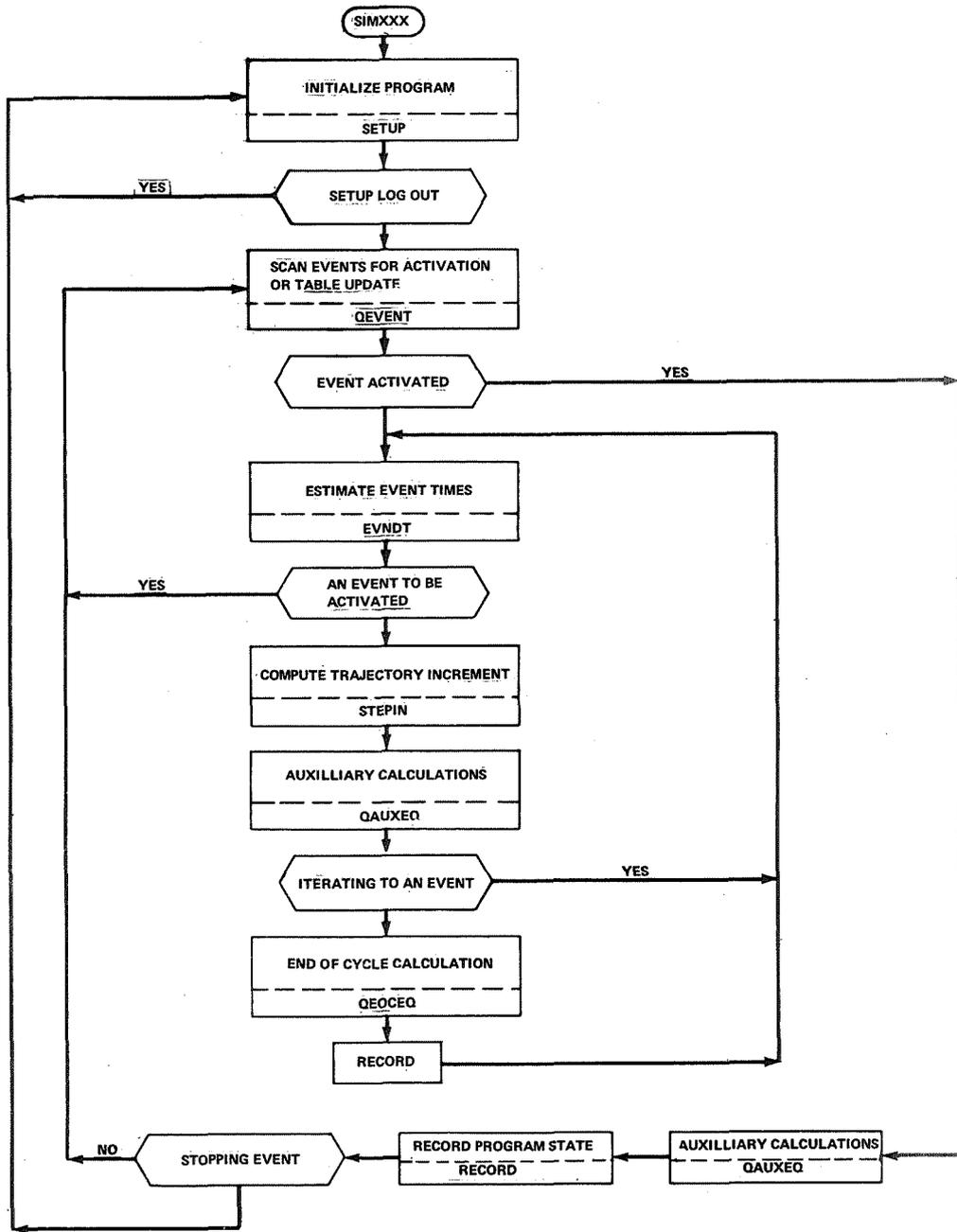
Called by: Separate computer execution.

Calls: SETUP, EVNACC, EVNDT, PRTCOM, STEPIN, PRTVAR, RECORD, internal subroutines AUXEQ and EOCEQ.

Errors and Diagnostics: Acknowledges log out from SETUP and STEPIN.

Language: Fortran main program

Computations: User coded as indicated in elements QAUXEQ and QEOCEQ.



VMASS\*TPF\$.SIMXXX

```

1      C      SIMXXX   R JESSUP - BELLCOMM - 06 JAN 69
2      C
3      C      PURPOSE
4      C
5      C      COMPUTE A VIRTUAL MASS TRAJECTORY (PROPAGATE A SPACE CRAFT INITIAL
6      C      STATE FROM A STARTING TO A STOPPING EVENT)
7      C
8      C      INPUTS
9      C
10     C      (LDDCOM) VALUES FOR FLAGS AND CONSTANTS REFLECTING PROGRAMMED
11     C      CONVENTIONS FOR TYPICAL RUNS
12     C      (SETUP) THE DATA-DECK GIVING THE INITIAL SPACE CRAFT STATE AND
13     C      THE STARTING AND STOPPING EVENTS (IEVSTR AND IEVSTP)
14     C      (STEPIN) COMPUTATIONS FOR INCREMENTAL STEPS IN SC STATE PROGATION
15     C      (EVNACC) EVENTS ACCOUNTING AND RECOGNITION
16     C      (AUXEQ) EVENT ASSOCIATED AUXILIARY COMPUTATIONS
17     C      (EOCEQ) EV ASSOC AUX EQ FOR NORMAL COMPUTATION STEPS ONLY
18     C
19     C      OUTPUTS
20     C
21     C      (RECORD) PRINTOUT - GRAPHS - 'PRINT LATER' TAPES
22     C
23     C      INCLUDE QTYPE
24     C      INCLUDE QVAR
25     C      INCLUDE QFAC
26     C      INCLUDE QEPH
27     C      INCLUDE EVARI
28     C      INCLUDE EVARL
29     C      INCLUDE EVARS
30     C      INCLUDE EVARV
31     C      INCLUDE EVARIO
32     C      INCLUDE QSTORE
33     C      INCLUDE QTEM
34     C
35     C      ON FIRST DATA DECK INITIALIZE FOR RUN WITH PRELOAD OF COMMON.
36     C      INITIALIZE IN ACCORDANCE WITH DATA DECKS AND USERS CODING THERE-
37     C      AFTER.
38     C
39     C      LDDCOM = .TRUE.
40     C      9000 CALL SETUP
41     C      IF (.NOT. LOGOUT) GO TO 9020
42     C      WRITE (IOMES, 9010)
43     C      9010  FORMAT (' SETUP CALLED BY MAIN PROGRAM')
44     C      CALL PRTCOM
45     C      GO TO 9000
46     C      9020 CONTINUE
47     C      9300 CONTINUE
48     C      INCLUDE QEVENT, LIST
49     C
50     C      AFTER UPDATE OF EVENTS TABLE CHECK FOR NEXT EVENT AND RESUME
51     C      CALCULATIONS
52     C
53     C      9911 CALL EVNDT
54     C      IF (LXECEV) GO TO 9300
55     C
56     C      OBTAIN INCREMENTAL STEP IN TRAJECTORY SEGMENT

```

```
57      C
58      9913 CALL STEPIN
59          IF (.NOT. LOGOUT) GO TO 9915
60          WRITE (IOMES, 9914)
61      9914  FORMAT (' STEPIN CALLED FROM EVENTS LIST')
62          CALL PRTVAR
63          GO TO 9000
64      C
65      C    AFTER EACH INCREMENTAL STEP, WHICH MAY INCLUDE ACTIVATION AND ONE
66      C    TIME CALCULATIONS OF AN EVENT, PERFORM USERS SUPPLIED AUXILLIARY
67      C    EQUATIONS.
68      C
69      9915 CALL AUXEQ
70          IF (.NOT. LXECEV) GO TO 9920
71      C
72      C    ON ACTIVATION OF AN EVENT RECORD TRAJECTORY STATE AND, UNTIL
73      C    STOPPING EVENT IS REACHED, UPDATE EVENTS TABLE AND CONTINUE
74      C    INCREMENTAL STEPS.
75      C
76          CALL RECORD
77          LXECEV = .FALSE.
78          IEVFLG = 1
79          IF (IEVCUR .NE. IEVSTP) GO TO 9300
80          GO TO 9000
81      C
82      C    AFTER INCREMENTAL STEPS NOT UNDER ADJUSTMENT TO REACH AN EVENT
83      C    (LIMPEV TRUE) PERFORM USERS SUPPLIED END CYCLE EQUATIONS. PROCEED
84      C    TO NEXT STEP.
85      C
86      9920 IF (LIMPEV) GO TO 9911
87          CALL EOCEQ
88          CALL RECORD
89          GO TO 9911
90      C
91      C
92          SUBROUTINE AUXEQ
93      C
94      C    PURPOSE
95      C
96      C    USERS SUPPLIED AUXILLIARY EQUATIONS
97      C
98          INCLUDE QAUXEQ, LIST
99          RETURN
100     C
101     C
102     ENTRY EOCEQ
103     C
104     C    PURPOSE
105     C
106     C    USERS SUPPLIED END OF CYCLE EQUATIONS
107     C
108     C    INCLUDE QEOCEQ, LIST
109     C
110     RETURN
111     END
```

Program: STEPIN

Purpose: Compute an increment in the spacecraft state vector.

Called by: SIMXXX

Calls: INDEX, CONIC, VALUE, KEPLER, EPHEM, DOT, VMXACT, SQRT,  
PREPAR

Errors and Diagnostics: Acknowledges log out from KEPLER and EPHEM.

Language: Fortran subroutine

Computations:

$$TTO = RTOPRF \cdot FTTO \cdot RRCAVM / VRCAVM$$

$$TTRI = \min (TEVI, TTO \cdot DTAI)$$

$$TTRE = TTRB + TRCI$$

$$RSCDVM_k = |RSCEVM_k|$$

$$T_{3,k} = \left( S_3 \cdot QVME_k - GVME_k \cdot (S_1 + S_4 \cdot RSCEVM_k \cdot VSCEVM_k \cdot RSCVDM_k^{-2}) \right) \cdot RSCDVM_k^{-3}$$

$$V_{1,k} = T_{3,k} \cdot RSCEVM_k + S_3 \cdot GVME_k \cdot RSCDVM_k^{-3} \cdot VSCEVM_k$$

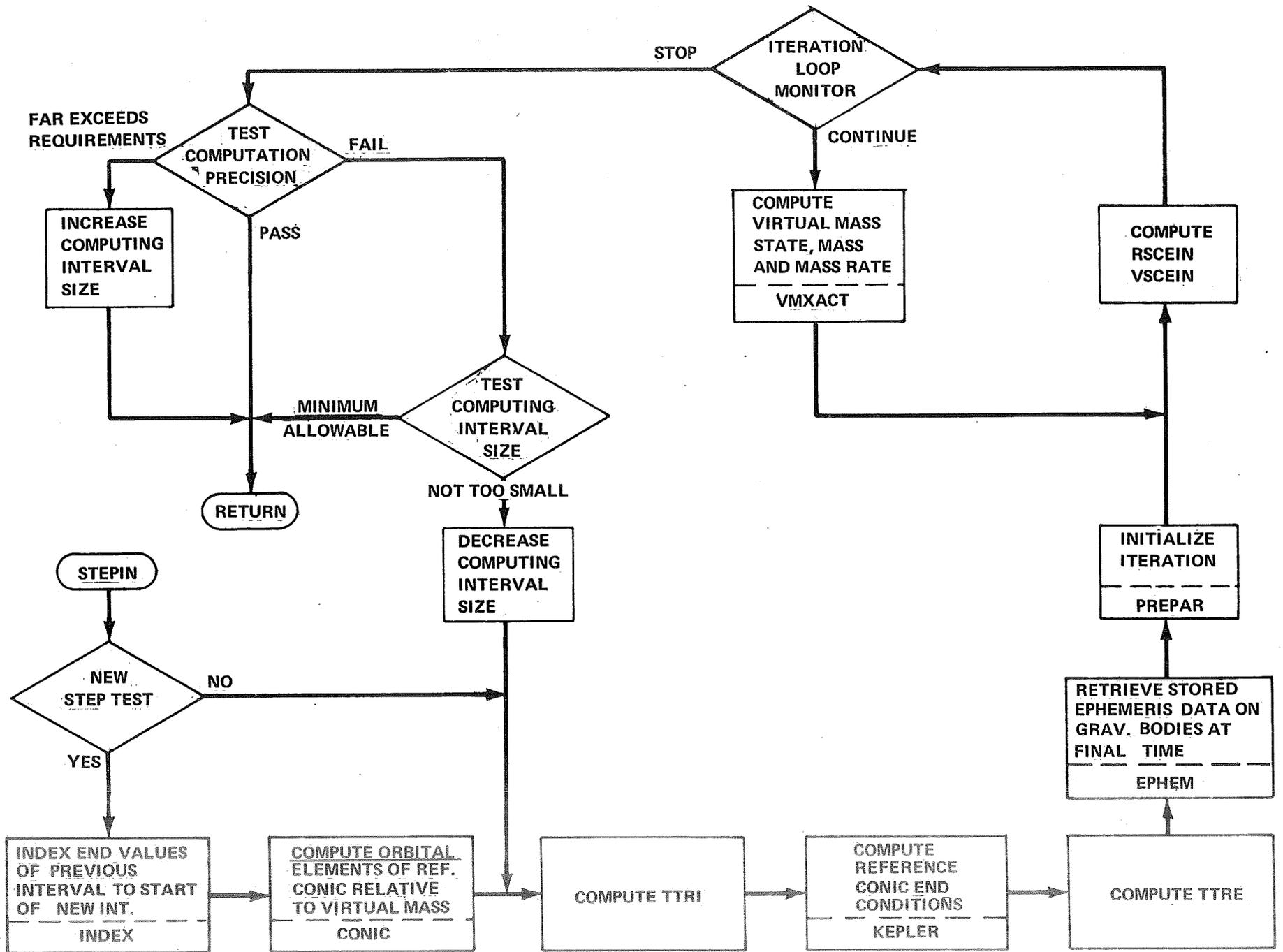
$$V_{2,k} = SV_{3,k-1}$$

$$SV_{3,k} = S_2 \cdot V_{1,k} + S_5 \cdot GVME_k \cdot RSCDVM_k^{-3} \cdot RSCEVM_k$$

$$RSCEIN_k = SV_1 + SV_{3,k}$$

$$VSCEIN_k = SV_2 + V_{1,k}$$

$$RTOPCM = |V_{2,k} - SV_{3,k}| \cdot RSCDVM_k^{-1}$$





```

57      C  INPUTS
58      C
59      C      DTAI      REFERENCE CONIC TRUE ANOMOLY ANGLE OVER INTERVAL
60      C      TEVI      ESTIMATED TIME INTERVAL TO NEXT EVENT
61      C      TTO      TOLERANCE ON ACCURACY OF COMPUTED TIME INCREMENT
62      C
63      C  OUTPUTS
64      C
65      C      TEPI      DESIRED COMPUTING TIME INTERVAL
66      C      ITRAT     COUNTER CONTROLLING THE COMPUTATION INTERVAL CHANGES
67      C
68      10  TTRI = MIN (TEVI, TTO * DTAI)
69          ITRAT = 0
70      20  CALL KEPLER
71          IF (.NOT. LOGOUT) GO TO 30
72          ICALL(1) = 'KEPLER'
73          GO TO 500
74      30  TTRE = TTRB + TRCI
75          CALL EPHEM
76          IF (.NOT. LOGOUT) GO TO 40
77          ICALL(1) = 'EPHEM'
78          GO TO 500
79      40  CALL PREPAR
80      C
81      C  UPDATE *** 5 LOCATIONS FROM PREPAR
82      C
83      60  T1      = DOT (RSCEVM, RSCEVM)
84          RSCDVM = SQRT (T1)
85          T2      = RSCDVM * T1
86          T3 = (S3 * QVME - GVME * (S1 + S4 * DOT(RSCEVM, VSCEVM) / T1)) / T2
87          T4 = S3 * GVME / T2
88          T5 = S5 * GVME / T2
89          T6 = F0
90          DO 62 I = 1, 3
91              TV1(I) = T3 * RSCEVM(I) + T4 * VSCEVM(I)
92              TV2(I) = SV3(I)
93              SV3(I) = S2 * TV1(I) + T5 * RSCEVM(I)
94              T6 = T6 + (TV2(I) - SV3(I)) ** 2
95              RSCEIN(I) = SV1(I) + SV3(I)
96          62  VSCEIN(I) = SV2(I) + TV1(I)
97          RTOPCM = SQRT (T6) / RSCDVM
98          IF (ITRAT .GE. 2) GO TO 70
99          ITRAT = ITRAT + 1
100         CALL VMXACT
101         GO TO 60
102      70  IF (RTOPCM .GT. RTOPRF) GO TO 80
103         IF (RTOPCM * FRTO .LT. RTOPRF) DTAI = DTAI * DTAIUP
104         RETURN
105      80  IF (DTAI .LE. DTAIMN) RETURN
106         DTAI = DTAI * DTAIDN
107         LINTDN = .TRUE.
108         GO TO 10
109      500 WRITE (IOPRT, 510) ICALL
110      510 FORMAT (1X, 4A6)
111         RETURN
112         END

```

Program: UNITV (a, b)

Purpose: Calculate b the unit vector of a.

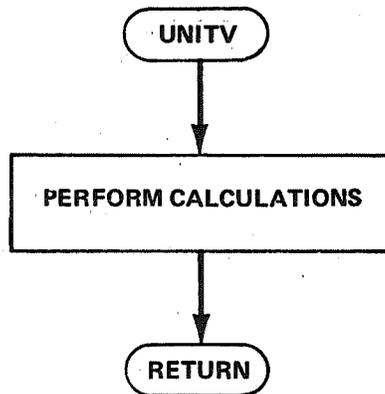
Called by: Kepler

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Calculations:  $b = \frac{a}{|a|}$



VMASS\*RFJK.UNITV

```
1      C
2      C  PURPOSE
3      C
4      C    CALCULATE B THE UNIT VECTOR OF A
5      C
6      C    SUBROUTINE UNITV (A, B)
7      C
8      C    INCLUDE QTYPE
9      C    INCLUDE QTEM
10     C    DIMENSION A(3), B(3)
11     C
12     C    T1 = VALUE (A)
13     C    DO 10 I = 1, 3
14     C      10    B(I) = A(I) / T1
15     C
16     C    RETURN
17     C    END
```

Program: s = VALUE (a)

Purpose: Calculate s the magnitude of the vector a.

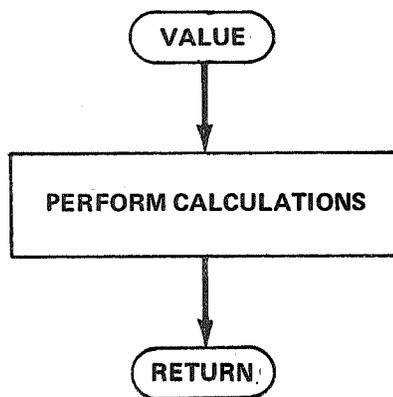
Called by: CONIC, KEPLER, STEPIN, UNITV

Calls: None

Error and Diagnostics: None

Language: Fortran function

Calculations:  $s = a_1^2 + a_2^2 + a_3^2$



### VMASS SUBROUTINE VALUE

VMASS\*RFJK.VALUE

```

1      C
2      C  PURPOSE
3      C
4      C  CALCULATE MAGNITUDE OF A 3 COMPONENT VECTOR
5      C
6      C  FUNCTION VALUE (A)
7      C
8      C  INCLUDE QTYPE
9      C  DIMENSION A(3)
10     C
11     C  VALUE = SQRT (A(1) ** 2 + A(2) ** 2 + A(3) ** 2)
12     C  RETURN
13     C  END
  
```

Program: VMXACT

Purpose: Compute the Virtual Mass inertial position, velocity, mass, and mass rate. Compute spacecraft position and velocity relative to the Virtual Mass and to each gravitating body.

Called by: STEPIN

Calls: DOT, SQRT

Errors and Diagnostics: None

Language: Fortran subroutine

Computations:

$$RSCEO_k = RSCEIN - ROOEIN_k$$

$$RSCDO_k = |RSCEO_k|$$

$$VSCEO_k = VSCEIN - VOOEIN_k$$

$$RUMEIN = \frac{1}{t_1} \sum_i \frac{ROOEIN_i \cdot GOO_i}{|RSCEO_i|^3}$$

$$VMMEIN = \frac{1}{t_1} \left( \sum_i \frac{GOO_k}{|RSCDO_i|^3} (VVOEIN_i - T_6 \cdot ROOEIN_i) \right)$$

$$RSCEVM = RSCEIN - VMMEIN$$

$$VSCEVM = VSCEIN - VMMEIN$$

$$RSCDVM = |RSCEVM|$$

$$GVME = T_1 \cdot |RSCEVM|^3$$

$$QVME = GVME \cdot 3 \cdot \frac{RSCEVM \cdot VSCEVM}{|RSCEVM|^2} + \frac{t_4}{t_1}$$

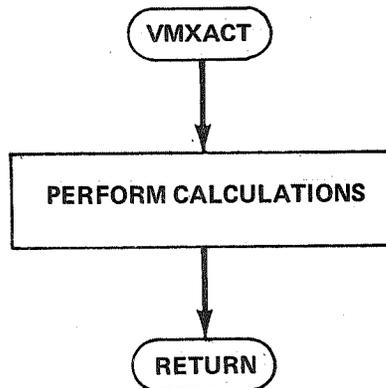
Where

$$t_1 = \sum_i \frac{GOO_i}{|RSCEOO_i|^3}$$

$$t_4 = - \sum_i \frac{GOO_i}{|RSCEOO_i|^3} \cdot t_6$$

$$t_6 = \frac{3 \cdot RSCEOO_i \cdot VSCEOO_i}{|RSCEOO_i|^2}$$

$i = 1, 2, \dots, JKN$



VMASS\*RFJK.VMXACT

```

1      C
2      C   PURPOSE
3      C
4      C   COMPUTES THE VIRTUAL MASS POSITION, MASS, VELOCITY, AND MASS RATE,
5      C   GIVEN THE INERTIAL POSITIONS AND VELOCITIES OF THE SPACE CRAFT AND
6      C   ALL THE GRAVITATING MASS POINTS.  OUTPUTS THE VIRTUAL MASS DATA IN
7      C   INERTIAL COORDINATES AND GIVES THE SPACECRAFT POSITION AND
8      C   VELOCITY RELATIVE TO THE VIRTUAL MASS.
9      C
10     C   INPUTS
11     C
12     C   JK       LIST OF POINT MASS SUBSCRIPTS
13     C   JKN     NUMBER OF SUBSCRIPTS IN LIST JK
14     C   GJK     PRODUCT OF UNIVERSAL GRAVITATION CONSTANT AND MASS OF THE
15     C           GRAVITATING POINT JK
16     C   RJKEIN  POSITION VECTOR OF MASS POINT JK RELATIVE TO AN INERTIAL
17     C           COORDINATE FRAME
18     C   VJKEIN  INERTIAL VELOCITY VECTOR OF MASS POINT JK
19     C   RSCEIN  INERTIAL POSITION VECTOR OF SPACECRAFT
20     C   VSCEIN  INERTIAL VELOCITY VECTOR OF SPACECRAFT
21     C
22     C   OUTPUT
23     C
24     C   RVMEIN   POSITION VECTOR OF THE VIRTUAL MASS RELATIVE TO AN
25     C           INERTIAL COORDINATE FRAME
26     C   VVMEIN   INERTIAL VELOCITY VECTOR OF THE VIRTUAL MASS
27     C   RSCEVM   POSITION OF SPACECRAFT RELATIVE TO THE VIRTUAL MASS
28     C   VSCEVM   VELOCITY OF SPACECRAFT RELATIVE TO THE VIRTUAL MASS
29     C   GVME     PRODUCT OF UNIVERSAL GRAVITATION CONST AND VIRTUAL MASS
30     C   QVME     PRODUCT OF UNIVERSAL GRAVITATION CONST AND RATE OF CHANGE
31     C           OF THE VIRTUAL MASS MAGNITUDE
32     C   RSCDVM   DISTANCE OF SPACECRAFT FROM VIRTUAL MASS POINT
33     C   RSCEJK   POS OF SPACECRAFT RELATIVE TO MASS POINT JK
34     C   VSCEJK   VEL
35     C   RSCDJK   DIST FROM SPACECRAFT TO MASS POINT JK
36     C
37     C   SUBROUTINE VMXACT
38     C
39     C   INCLUDE QTYPE
40     C   INCLUDE QVAR
41     C   INCLUDE QFAC
42     C   INCLUDE QPHC
43     C   INCLUDE QTEM
44     C   INCLUDE QEPH
45     C   INCLUDE EVARS
46     C   INCLUDE EVARV
47     C   INCLUDE EVARI
48     C
49     C   T1 = F0
50     C   T4 = F0
51     C   DO 2 M = 1, 3
52     C       TV1(M) = F0
53     C       TV2(M) = F0
54     C   DO 10 I = 1, JKN
55     C       K = JKLOC (JK(I))
56     C       DO 4 M = 1, 3

```

```

57          RSCE00(K+M-1) = RSCEIN(M) - R00EIN(K+M-1)
58          VSCE00(K+M-1) = VSCEIN(M) - V00EIN(K+M-1)
59          T2          = DOT (RSCE00(K), RSCE00(K))
60          RSCD00(K)  = SQRT (T2)
61          T3          = G00(K) / (T2 * RSCD00(K))
62          T1          = T1 + T3
63          T6          = F3 * DOT (RSCE00(K), VSCE00(K)) / T2
64          T4          = T4 - T6 * T3
65          DO 6 M = 1, 3
66              TV1(M)  = TV1(M) + R00EIN(K+M-1) * T3
67              TV2(M)  = TV2(M)
68              + T3 * (V00EIN(K+M-1) - R00EIN(K+M-1) * T6)
69          CONTINUE
70          DO 12 M = 1, 3
71              RVMEIN(M) = TV1(M) / T1
72              VVMEIN(M) = (TV2(M) - RVMEIN(M) * T4) / T1
73              RSCEVM(M) = RSCEIN(M) - RVMEIN(M)
74              VSCEVM(M) = VSCEIN(M) - VVMEIN(M)
75          T5          = DOT (RSCEVM, RSCEVM)
76          RSCDVM     = SQRT (T5)
77          GVME       = T1 * RSCDVM * T5
78          QVME       = GVME * (F3 * DOT (RSCEVM, VSCEVM) / T5 + T4 / T1)
79          C
80          RETURN
81          END

```

Q FIN

**BELLCOMM, INC.**

Subject: The Univac 1108 Virtual Mass  
Trajectory Simulation Program

From: R. F. Jessup

Distribution List

Complete Memorandum to

NASA Headquarters

V. N. Huff/MTE  
A. S. Lyman/MR

Ames Research Center

J. M. Deerwester/202-9

Goddard Space Flight Center

R. E. Coady/551  
C. E. Doll, Jr./552  
M. L. Mouton/553  
R. K. Squires/643

Jet Propulsion Laboratory

J. D. Mulholland  
F. M. Sturms, Jr.

Langley Research Center

C. P. Llewellyn/156  
W. H. Michael, Jr./152A  
R. H. Tolson/159

Lewis Research Center

E. A. Willis/501-2

Manned Spacecraft Center

H. D. Beck/FM5  
J. R. Elk/FM5  
J. Funk/FM8  
T. F. Gibson, Jr./FS5  
J. M. Lewallen/ED13  
R. O. Nobles/FM7

Marshall Space Flight Center

R. F. Arenstorff/S&E-COMP-T  
A. W. Deaton/S&E-AERO-GG  
J. Lindberg/S&E-AERO-M  
H. F. Thomae/PD-DO-P  
G. Wittenstein/S&E-AERO-MFT

Analytical Mechanics Associates

J. T. Findlay  
S. Pines

The Boeing Company

J. Coons

Computer Sciences Corporation

D. H. Novak

Martin-Marietta Corporation

W. J. Pragluski

MIT - Instrumentation Laboratory

R. H. Battin

The University of Texas at Austin

V. Szebehely

U. S. Naval Observatory

T. C. Van Flandern

Bellcomm, Inc.

M. V. Bullock  
J. O. Cappellari, Jr.  
L. P. Gieseler  
W. G. Heffron

BELLCOMM. INC.

Distribution List (Contd.)

Bellcomm, Inc.

K. E. Martersteck  
C. C. H. Tang  
Central File  
Department 1024 File  
Library

Memorandum without Appendix to

A. P. Boysen, Jr.  
J. P. Downs  
D. R. Hagner  
D. P. Ling  
J. Z. Menard  
J. W. Timko  
R. L. Wagner  
M. P. Wilson

Abstract Only to

G. M. Anderson  
R. A. Bass  
H. B. Bosch  
K. M. Carlson  
K. R. Carpenter  
D. A. Corey  
D. A. De Graaf  
F. El-Baz  
A. J. Ferrari  
E. M. Grenning  
N. W. Hanners  
T. B. Hoekstra  
A. N. Kontaratos  
S. L. Levie, Jr.  
M. Liwshitz  
H. S. London  
D. Macchia  
J. L. Marshall, Jr.  
W. I. McLaughlin  
G. T. Orrok  
P. E. Reynolds  
P. S. Schaenman  
F. N. Schmidt  
R. J. Stern  
A. R. Vernon  
G. D. Wolske  
T. L. Yang