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.

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

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 subroutines. 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:

Figure	Description
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:

Figure	Description
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

2011-RFJ-vh

Attachments

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

```
@ ELT,DLI RUN100
ELT PROCESSOR LEVEL      3
       @ HDG     VMASS EARTH TO MARS / R JESSUP
       @ USE     LAUPLT,AUPLOT*RFJI
       @ PDP,FIL PRB100
       @EVENT PROC
       C  REFERENCE - DOUBLE PRECISION TRAJECTORY PROGRAM DPTRAJ, JPL, JULY 69
                CALL EVNACC (' 0', F0,   ' 0', TEPEJ,  F0,  $210)
       C        CONVERT EARTH CENTERED INPUT (RSCDDK, VSCDDK) TO INERTIAL
                   CALL EPHEM
                   RUNIT = AUTOKM
                   VUNIT = AUTOKM / F86400
                   DO 120 I = 1, 3
                      RSCEIN(I) = ROOEIN(KEARTH-1+I) + RSCDDK(I) / RUNIT
         120          VSCEIN(I) = VOOEIN(KEARTH-1+I) + VSCDDK(I) / VUNIT
       C        INITIALIZE SIMULATOR
                   LPRNOW = .TRUE.
                   CALL VMXACT
                   IEVREP = IEVCUR
                   GO TO 9915
         210    CALL EVNACC (' 1', ' 0', ' 1', A1, C1, $220)
                   GO TO 9915
         220    CALL EVNACC (' 2', ' 1', ' 2', A2, C2, $230)
                   GO TO 9915
         230    CALL EVNACC (' 3', ' 2', ' 3', A3, C3, $240)
                DIMENSION RSCX04(3) / .25560119770607 53D5, -.11190645066314 82D6,
              1    -.56731591824453 49D5 /
                DIMENSION VSCX04(3) / -.45166954190225 51D1, -.11447672859677 80D1,
              1    .30564843652263 3800 /
                   DO 232 I = 1, 3
                      TV1(I) = RSCX04(I) - RUNIT * RSCE00(KMARS-1+I)
         232          TV2(I) = VSCX04(I) - VUNIT * VSCE00(KMARS-1+I)
                   T1 = VALUE (TV1)
                   T2 = VALUE (TV2)
                   WRITE (IOPRT, 234) TV1, T1, TV2, T2
         234       FORMAT (//' FINAL STATE VECTOR DISPLACEMENT FROM DPTRAJ'/
              1      ' POSITION', 4G30.18/ ' VELOCITY', 4G30.18)
                   LPRNOW = .FALSE.
                   IEVSTP = IEVCUR
                   GO TO 9915
         240    CONTINUE
         END
       @AUXEQ PROC
                TEPEJ = TEPEJD + TEPEJS / F86400
                IF (IEVCUR .NE. ' 0')  GO TO 510
                   A1 = VALUE (RVMEIN)
                   RETURN
         510    IF (IEVCUR .NE. ' 1') GO TO 520
                   A2 = VALUE (RSCE00(KMARS))
                   RETURN
         520    CONTINUE
         END
       @STORE PROC
                EQUIVALENCE              (VAC(1),TEPEJ,A3),   (VAC(2), A1    ),
              1    (VAC(3), A2     ),    (VAC(4), RUNIT ),    (VAC(5), VUNIT )
                DATA C1 /  .100 / C2 / .100 / C3 / .24389566302580 42D7 /
         END
```

```
000    QPRINT PROC
000          CALL SCALE (RUNIT, RSCEOO(KEARTH), TV1)
000          CALL SCALE (VUNIT, VSCEOO(KEARTH), TV2)
000          CALL SCALE (RUNIT, RSCEOO(KSUN  ), TV3)
000          CALL SCALE (VUNIT, VSCEOO(KSUN  ), TV4)
000          CALL SCALE (RUNIT, RSCEOO(KMARS ), TV5)
000          CALL SCALE (VUNIT, VSCEOO(KMARS ), TV6)
000          WRITE (IOPRT, 710) (TV1(I), I = 1, 18)
000     710     FORMAT ( // ' RSCE03', 3G30.18/ ' VSCE03', 3G30.18/
000       1                 ' RSCE10', 3G30.18/ ' VSCE10', 3G30.18/
000       2                 ' RSCE04', 3G30.18/ ' VSCE04', 3G30.18)
000             PRINT 720, TEPEJ, INCNTR
000     720     FORMAT (' TEPEJ ', G30.18, 18X, 'INCNTR', I6)
000     END
000    GEOCEQ PROC
000     END
000    @ COPY,S LVMASS.COMMON,COMMON
000    @ COPY,S LVMASS.SIMXXX,SIMXXX
000    @ COPY,S LVMASS.PRTXXX,PRTXXX
000    @ FOR,W  SIMXXX,SIMXXX
000    @ FOR,W  PRTXXX,PRTXXX
000    -22,23
000          INCLUDE QFAC
000          INCLUDE QTEM
000          INCLUDE QEPH
000    @ MAP    LVMASS.MAP,MAP100
000    @ ASG,T  12,T,1277
000    @ XQT    MAP100
000     $DATADK
000         RSCDDK = -.15521917D7,        .11758164D7,          .5465127006,
000         VSCDDK = -.24260004D1,        .13151313D1,          .8673262200,
000         TEPOJD = .2438735D7,      TEPOJS = 14965.D0,
000          RTOPRF = 1.D-19,
000     $END
000     $DATADK
000          RTOPRF = 1.D-18,
000     $END
000     $DATADK
000          LSTDDK = .TRUE.,
000          RTOPRF = 1.D-17,
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      -.155219169999999998+007      .117561639999999988+007      .546512699999999935+006
VSCE03      -.242600039999999997+001      .181513129999999999+001      .867326219999999994+000
RSCE10       .406216922642625586+008      .130753700775912621+009      .567375381543019336+008
VSCE10      -.314683763039406437+002      .953919393201119771+001      .421809095464935946+001
RSCE04       .189795871644716737+009     -.457915839192475830+008     -.283052261211136422+008
VSCE04      -.131440551501514505+002      .212038419512550728+002      .908028458992784089+001
TEPEJ        .243673517320601851+007             INCNTR     0

EVNDT   *** 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      .136530500635702950+009      .593048292702085816+008
VSCE10      -.323262788911615265+002      .533869080248532350+001      .239373857162319780+001
RSCE04       .178798719769476169+009     -.304960078488106155+008     -.217396542812426426+008
VSCE04      -.150738611420174361+002      .181832023437669104+002      .782629215593127879+001
TEPEJ        .243874415873242562+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      .136530500635702950+009      .593048292702085816+008
VSCE10      -.323262788911615265+002      .533869080248532350+001      .239373857162319780+001
RSCE04       .178798719769476169+009     -.304960078488106155+008     -.217396542812426426+008
VSCE04      -.150738611420174361+002      .181832023437669104+002      .782629215593127879+001
TEPEJ        .243874415873242562+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     -.655776418835555163+001
RSCE04       .102440721957332507+009      .154000657887324199+008     -.136530801515766954+007
VSCE04      -.148698219333432744+002      .299207102396274033+001      .155030323455757030+001
TEPEJ        .243879837163260961+007             INCNTR   500

EVNDT   *** 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    -.1572444787803644407+009    .3181177447537781155+008    .1413315819803070050+008
VSCE03    -.2025076917804821155+002    -.1192039285411690008+002   -.5206090339077762245+001
RSCE10    -.1946470300552542268+009    -.1031424144836444423+009   -.4438960919169088894+008
VSCE10     .8126085449714074444+001    -.1877428900187782291+002   -.8178235929629311182+001
RSCE04     .1434377367509151125+008    .4183247606214601142+007    -.7428944226779592277+006
VSCE04    -.4552698292274840057+001    -.1495555523895709971+001    .9341059436617583311-001
TEPEJ      .2436918323240023225+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    -.1572444787803644407+009    .3181177447537781155+008    .1413315819803070050+008
VSCE03    -.2025076917804821155+002    -.1192039285411690008+002   -.5206090339077762245+001
RSCE10    -.1946470300552542268+009    -.1031424144836444423+009   -.4438960919169088894+008
VSCE10     .8126085449714074444+001    -.1877428900187782291+002   -.8178235929629311182+001
RSCE04     .1434377367509151125+008    .4183247606214601142+007    -.7428944226779592277+006
VSCE04    -.4552698292274840057+001    -.1495555523895709971+001    .9341059436617583311-001
TEPEJ      .2436918323240023225+007              INCNTR   753
```


```
RSCE03    -.2113066318439352291+009    -.2339592602514815596+008   -.9936208173227514492+007
VSCE03    -.1326836644610722241+002    -.2350275624598738856+002   -.1023749363866663371+002
RSCE10    -.1596085956614149963+009    -.1546668826588509964+009   -.6686213953335256690+008
VSCE10     .1424211322085488874+002    -.1434533825956324111+002   -.6265398672170794430+001
RSCE04     .9580266381001726615+006    .1488014506187698870+006    -.1165115018961182243+006
VSCE04    -.4238563330255384467+001    -.1194186772060201171+001    .2674190737436872283+000
TEPEJ      .2436954095097523317+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    -.2141327983509096612+009    -.2861937585566105588+008   -.1221140874588069924+008
VSCE03    -.1255233325335450032+002    -.2416436782751887782+002   -.1050788437736036445+002
RSCE10    -.1566527692110111667+009    -.1577588862791418807+009   -.6821633295058021188+008
VSCE10     .1455006494811680093+002    -.1393557867477000086+002   -.6071598064118025536+001
RSCE04     .2566072339562988866+005    -.1119065390943291124+006   -.5673171562766650034+005
VSCE04    -.4316694669266830034+001    -.1144767717075340006+001    .3056480640129062282+000
TEPEJ      .2436956650258041199+007              INCNTR  1413

FINAL STATE VECTOR DISPLACEMENT FROM DPTRAJ
POSITION      -.6036250222356652607+000     .8843118092454460057-001     .1238032130135025000+000     .6225033951130271136+000
VELOCITY      -.7497557206539995536-006     .4311075600584718839-006     .3725097275177925572-006     .9416745011352633381-006
```

```
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


RSCE03      -.155219169999999996+007        .117581639999999988+007        .546512699999999935+006
VSCE03      -.242600039999999997+001        .181513129999999999+001        .867326219999999994+000
RSCE10       .406216922642625586+008        .130753700775912621+009        .567375381543019336+008
VSCE10      -.314683763039406437+002        .953919393201119771+001        .421809095464935946+001
RSCE04       .189795871644716737+009       -.457915839192475830+008       -.283052261211136422+008
VSCE04      -.131440551501514505+002        .212058419512550728+002        .908028458992784089+001
TEPEJ        .243673517320601851+007                  INCNTR     0

EVNDT  *** DIVERGENCE
  1      1        603     1161      14        0      .826917422237503680-003      -.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


RSCE03      -.339041922600075388+007        .260354788804821297+007        .122749937520827759+007
VSCE03      -.231024373417344776+001        .187957769907291665+001        .894328929948513133+000
RSCE10       .156090592659951850+008        .136550500635263443+009        .593048292700221587+008
VSCE10      -.323262788918988624+002        .553869080102586728+001        .239373857098966016+001
RSCE04       .178798719768182806+009       -.304960078485132024+008       -.217396542811173873+008
VSCE04      -.150738611428370849+002        .181852023423918046+002        .782629215533866969+001
TEPEJ        .243674415873242629+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


RSCE03      -.339041922600075388+007        .260354788604821297+007        .122749937520827759+007
VSCE03      -.231024373417344776+001        .187957769907291665+001        .894328929948513133+000
RSCE10       .156090592659951850+008        .136550500635263443+009        .593048292700221587+008
VSCE10      -.323262788918988624+002        .553869080102586728+001        .239373857098966016+001
RSCE04       .178798719768182806+009       -.304960078485132024+008       -.217396542811173873+008
VSCE04      -.150738611428370849+002        .181852023423918046+002        .782629215533866969+001
TEPEJ        .243674415873242629+007                  INCNTR   115

RSCE03      -.134677783244204517+009        .424812841601604145+008        .187966166934305132+008
VSCE03      -.206633142668761930+002       -.771306512790405946+001       -.337692229881432601+001
RSCE10      -.202305233088469065+009       -.820272284745426211+008       -.351954341214974320+008
VSCE10       .553704449068721748+001       -.199799828058093284+002       -.869645761581737779+001
RSCE04       .194704679448169350+008        .591113529868727054+007       -.785708375735651996+006
VSCE04      -.489048485662048765+001       -.168062237544438310+001       -.168690369263072990-001
TEPEJ        .243689057226121168+007                  INCNTR   500

EVNDT  *** DIVERGENCE
  2      2        605     1163      14        0      .534923430377383662-003      -.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    -.149555552367720080+001     .934105944667749943-001
TEPEJ       .243891832324002665+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    -.149555552367720080+001     .934105944667749943-001
TEPEJ       .243891832324002665+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     -.125523332080620359+002    -.241643677882598541+002    -.105078843621804213+002
RSCE10     -.156652769195972928+009    -.157768886284283421+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     .935737825346905083-001     .126017383246221470+000     .638265685218974857+000
VELOCITY   -.795048482098159592-006     .391848782267320561-006     .357329891189760857-006     .955684157129674259-006
```

Figure 2 - Page 5

```
VIRTUAL MASS TRAJECTORY 08 JUN 70 15:41:06

 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 15:41:12


RSCE03      -.155219169999999996+007     .117561639999999998+007     .546512699999999935+006
VSCE03      -.242600039999999997+001     .181513129999999999+001     .867326219999999994+000
RSCE10       .406216922642625586+008     .130733700775912621+009     .567375381543019336+008
VSCE10      -.314683763039406437+002     .953919393201119771+001     .421809095464935946+001
RSCE04       .189795871644716737+009    -.437915839192475830+006    -.283052261211136422+008
VSCE04      -.131440551501514505+002     .212038419512550728+002     .908028458992784089+001
TEPEJ        .2438735517320601651+007          INCNTR       0

EVNDT  *** DIVERGENCE
  1      1           603       1161        14          0        .501572642973191875-003       -.552943107967607261-017

 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 15:41:16


RSCE03      -.339041923075764544+007     .260304788491411447+007     .122749937387137144+007
VSCE03      -.231024373741807747+001     .187957769354158845+001     .894328927547240147+000
RSCE10       .158690592541148346+008     .136550500633180752+009     .593048292691410024+008
VSCE10      -.323262788953763557+002     .533869079380586536+001     .239373856785516361+001
RSCE04       .178798719761546376+009    -.304950078466917684+008    -.217396542803473035+008
VSCE04      -.150738611467506635+002     .181852023356186830+002     .782629215242100860+001
TEPEJ        .243874415873242981+007           INCNTR      80

 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 15:41:16


RSCE03      -.339041923075764544+007     .260304788491411447+007     .122749937387137144+007
VSCE03      -.231024373741807747+001     .187957769354158845+001     .894328927547240147+000
RSCE10       .158690592541148346+008     .136550500633180752+009     .593048292691410024+008
VSCE10      -.323262788953763557+002     .533869079380586536+001     .239373856785516361+001
RSCE04       .178798719761546376+009    -.304950078466917684+008    -.217396542803473035+008
VSCE04      -.150738611467506635+002     .181852023356186830+002     .782629215242100860+001
TEPEJ        .243874415873242981+007           INCNTR      80

EVNDT  *** DIVERGENCE
  2      2           605       1163        14          0        .212599659115527762-002       -.607153216591882483-017

 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 15:41:28


RSCE03      -.137244478796029287+009     .318117743531921416+008     .141331581447467910+008
VSCE03      -.202507691713121663+002    -.119203928633456936+002    -.520609034310783734+001
RSCE10      -.194647029996319427+009    -.103142414623848258+009    -.443896092527882568+008
```

VSCE10	.812608546010923055+001	-.187742889976421346+002	-.817823592781141615+001
RSCE04	.143437737006951847+008	.418324751143471175+007	-.742894462030051163+006
VSCE04	-.455269828721685679+001	-.149555552239800396+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	-.157244476796029287+009	.318117743531921416+008	.141331581447467910+008
VSCE03	-.202507691713121665+002	-.119203928633456936+002	-.520609034310783734+001
RSCE10	-.194647029996319427+009	-.103142414623848258+009	-.443896092527882568+008
VSCE10	.812608546010923055+001	-.187742889976421346+002	-.817823592781141615+001
RSCE04	.143437737006951847+008	.418324751143471175+007	-.742894462030051163+006
VSCE04	-.455269828721685679+001	-.149555552239800396+001	.934105950706054625-001
TEPEJ	.243691832324005367+007	INCNTR 359	

RSCE03	-.212283039626351223+009	-.251474357420816121+008	-.106999689306137767+008
VSCE03	-.150214601712035374+002	-.237492187095143681+002	-.103421940961186272+002
RSCE10	-.158746999622139850+009	-.155727145121741524+009	-.673252046748644564+008
VSCE10	.143623905604722873+002	-.142239805524749626+002	-.621200401959359494+001
RSCE04	.643280903429541219+006	.601794353014502890+005	-.966230336932313190+005
VSCE04	-.424514335649002153+001	-.119395525576912951+001	.268567669557245319+000
TEPEJ	.243895495416752635+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	-.214132798251912537+009	-.236193758825620669+008	-.122114087573768955+008
VSCE03	-.125523329596146074+002	-.241643675682281622+002	-.105078842759119999+002
RSCE10	-.156652769112013347+009	-.157796886306042346+009	-.682163329620762179+008
VSCE10	.145300652418586063+002	-.139395784154790694+002	-.607159796266956413+001
RSCE04	.256608225930837865+005	-.111976565994279399+006	-.567317271233726893+005
VSCE04	-.431669437552710940+001	-.114476745776447738+001	.305648165461334444+000
TEPEJ	.243895663025804199+007	INCNTR 667	

FINAL STATE VECTOR DISPLACEMENT FROM OPTRAJ

POSITION	-.762622476256578921+000	.115331131199013726+000	.135298919199442480+000	.724765763196019207+000
VELOCITY	-.160349544159665353-005	.171816697381796391-006	.271061299354953544-006	.109173162551067398-005

Earth to Mars Trajectory with Graphics

```
VMASS*RFJK.RUN110
@ HDG     VIRTUAL MASS 110 EARTH TO MARS WITH GRAPHICS / R JESSUP
@ PDP,FLI QEVENT/RUN110
QEVENT PROC
C       EARTH TO MARS FLIGHT WITH EPHEMERIS FILE
        CALL EVNACC (' 0', F0,    ' 0', TEPEJ,  F0,  $210)
C       CONVERT EARTH CENTERED INPUT (RSCDDK, VSCDDK) TO INERTIAL
           CALL EPHEM
           IF (.NOT. LOGOUT) GO TO 110
              WRITE (IOMES, 100)
              CALL PRTVAR
              CALL EXIT
  100         FORMAT (' EPHEM  CALLED BY SIMXXX')
  110      RUNIT = AUTOKM
           VUNIT = AUTOKM / F86400
           DO 120 I = 1, 3
              RSCEIN(I) = R00EIN(KEARTH-1+I) + RSCDDK(I) / RUNIT
  120         VSCEIN(I) = V00EIN(KEARTH-1+I) + VSCDDK(I) / VUNIT
           CALL PLTITL ('SPACE CRAFT AND VIRTUAL MASS DEPARTING EARTH')
           REAL EYE(3), ORIGIN(3)
           DATA EYE /   .0, -31.9, 91.6 / ORIGIN / 3*0. /
           CALL PRJEYE (EYE, 1)
           XL = -2.3D0
           XU =    .7D0
           YL = -1.2D0
           YU =  1.8D0
C       INITIALIZE SIMULATOR
           CALL VMXACT
           LPRNOW = .TRUE.
           GO TO 9915
  210   CALL EVNACC (' 1', ' 0', ' 1', A1, C1, $220)
           CALL PLOT
           GO TO 9915
  220   CALL EVNACC (' 2', ' 1', ' 2', A2, C2, $230)
           CALL PLTITL ('TRANS MARS MOVEMENT')
           CALL PLOT
           GO TO 9915
  230   CALL EVNACC (' 3', ' 2', ' 3', A3, C3, $240)
           CALL PLTITL ('SPACE CRAFT AND VIRTUAL MASS APPROACHING MARS')
           CALL PLOT
           CALL PLTEND
           CALL EXIT
  240   CONTINUE
  END
@ PDP,FLI QAUXEQ/RUN110
QAUXEQ PROC
        TEPEJ = TEPEJD + TEPEJS / F86400
        IF (IEVCUR .NE. ' 0')  GO TO 510
           A1 = VALUE (RVMEIN)
           RETURN
  510   IF (IEVCUR .NE. ' 1') GO TO 520
           A2 = VALUE (RSCE00(KMARS))
           RETURN
  520   CONTINUE
  END
@ PDP,FLI QEOCEQ/RUN110
QEOCEQ PROC
```

```
          DO 530 I = 0, 2
              CALL COLECT ('      X' + I, SNGL (RSCEIN(I+1)))
              CALL COLECT ('RVMEIX' + I, SNGL (RVMEIN(I+1)))
              CALL COLECT (KEARTH    + I, SNGL (ROOEIN(KEARTH+I)))
              CALL COLECT (KMARS     + I, SNGL (ROOEIN(KMARS +I)))
  530         CALL COLECT (KSUN      + I, SNGL (ROOEIN(KSUN  +I)))
          IF (MOD (INCNTR, IPLOTM) .NE. 0) RETURN
          IF (INCNTR .NE. 0) GO TO 540
              CALL PLTOUT
              CALL PLTICS ('      X', SNGL (XL), SNGL (XU))
              CALL PLTICS ('      Y', SNGL (YL), SNGL (YU))
  540     CALL PLOT
      END
@ PDP,FLI QSTORE/RUN110
QSTORE PROC
          EQUIVALENCE             (VAC(1),TEPEJ,A3),  (VAC(2), A1      ),
         2   (VAC(3), A2     ),    (VAC(4), RUNIT ),   (VAC(5), VUNIT ),
         3   (VAC(7), XL     ),
         4   (VAC(8), XU     ),    (VAC(9), YL     ),   (VAC(10), YU     )
C         NOTE. THE DATA STATEMENT MAY NOT BE USED FOR ASSIGNMENTS TO COMMON
C         SEE LODCOM WHICH CLEARS COMMON TO 0 ON PROGRAM STARTUP
          DATA C1 / .10D0 / C2 / .1D0 / C3 / .24389566302580 42D7 /
          IXP (X) = IFIX (1024. * SNGL ((X - XL) / (XU - XL)))
          IYP (Y) = IFIX (1024. * SNGL ((Y - YL) / (YU - YL)))
      END
@ PDP,FLI QPRINT/RUN110
QPRINT PROC
          CALL SCALE (RUNIT, RSCEOO(KEARTH), TV1)
          CALL SCALE (VUNIT, VSCEOO(KEARTH), TV2)
          CALL SCALE (RUNIT, RSCEOO(KSUN  ), TV3)
          CALL SCALE (VUNIT, VSCEOO(KSUN  ), TV4)
          CALL SCALE (RUNIT, RSCEOO(KMARS ), TV5)
          CALL SCALE (VUNIT, VSCEOO(KMARS ), TV6)
          WRITE (IOPRT, 710) (TV1(I), I = 1, 18)
  710         FORMAT ( // ' RSCE03', 3G30.18/ ' VSCE03', 3G30.18/
         1                 ' RSCE10', 3G30.18/ ' VSCE10', 3G30.18/
         2                 ' RSCE04', 3G30.18/ ' VSCE04', 3G30.18)
              PRINT 720, TEPEJ, INCNTR
  720         FORMAT (' TEPEJ ', G30.18, 18X, 'INCNTR', I6)
              RETURN
          ENTRY PLOT
  750     CALL PLTSIM (1)
              CALL PLCHAR ('S')
              CALL Q3DPER ('      X', '      Y', '      Z')
              CALL PLCHAR ('V')
              CALL Q3DPER ('RVMEIX', 'RVMEIY', 'RVMEIZ')
              CALL PLCHAR ( 0 )
              CALL Q3DPER (KEARTH, KEARTH+1, KEARTH+2)
              CALL Q3DPER (KMARS , KMARS +1, KMARS +2)
              CALL Q3DPER (KSUN  , KSUN  +1, KSUN  +2)
              CALL AURITE (IXP (ROOEIN(KEARTH)+.1), IYP (ROOEIN(KEARTH+1)),
         1         1023, 90, 05, 'EARTH')
              CALL AURITE (IXP (ROOEIN(KMARS )-.5), IYP (ROOEIN(KMARS +1)),
         1         1023, 90, 04, 'MARS')
              CALL AURITE (IXP (ROOEIN(KSUN  )+.1), IYP (ROOEIN(KSUN  +1)),
         1         1023, 90, 03, 'SUN')
      END
```

```
@ COPY,S LVMASS.SIMXXX,SIMXXX
@ COPY,S LVMASS.PRTXXX,PRTXXX
@ COPY,S LVMASS.COMMON,COMMON
@ FOR,W   SIMXXX.SIMXXX
@ FOR,W   PRTXXX.PRTXXX
-23,23
        INCLUDE @FAC
        INCLUDE @TEM
        INCLUDE @EPH
@ MAP     LVMASS.MAP,MAP110
@ XQT     MAP110
 $DATADK
      RSCDDK = -.15521917D7,          .11758164D7,          .5465127OD6,
      VSCDDK = -.24260004D1,          .18151313D1,          .86732622DO,
      TEPOJD = .2438735D7,      TEPOJS = 14965.DO,
       LSTDDK = .TRUE.,
       IPLOTM = 16.
 $END
@ FREE    12.
@ ASG,TM PLOTFILE,T,PLOT
@ 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      -.314683760039406437+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+005 SEC (1964 DEC 14 15:48:34); COMPUTED 25 SEP 70 18:15:04


RSCE03      -.339041929273824328+007     .260364781445131577+007     .122749934381066057+007
VSCE03      -.231024381159086105+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+005 SEC (1964 DEC 14 15:48:34); COMPUTED 25 SEP 70 18:15:06


RSCE03      -.339041929273824328+007     .260364781445131577+007     .122749934381066057+007
VSCE03      -.231024381159086105+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 18:16:20


RSCE03      -.157244479098136979+009     .318117720416785605+008     .141331571367142137+008
VSCE03      -.202507690435170196+002    -.119203930392418903+002    -.620609041991881336+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     .934106083888519278-001
TEPEJ        .243891832324063290+007             INCNTR   171
```

Figure 4 - Page 2

```
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 19:45:27); COMPUTED 25 SEP 70 18:15:20

RSCE03    -.157244479098136979+009    .318117720416785605+008    .141331571367142137+008
VSCE03    -.202507690435170196+002    -.119203930392418903+002    -.620609041991881336+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    .934106083888519278-001
TEPEJ     .243891832324063290+007              INCNTR   171

3     AT  221 DAYS 10 HR 58 MIN  9 SEC; JULIAN 2438956.0 DAYS .544543+005 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    -.111906771495006057+006    -.567318070187339697+005
VSCE04    -.431668753725688159+001    -.114476081829608474+001    .305650897790385680+000
TEPEJ     .243895663025804199+007              INCNTR   317
```

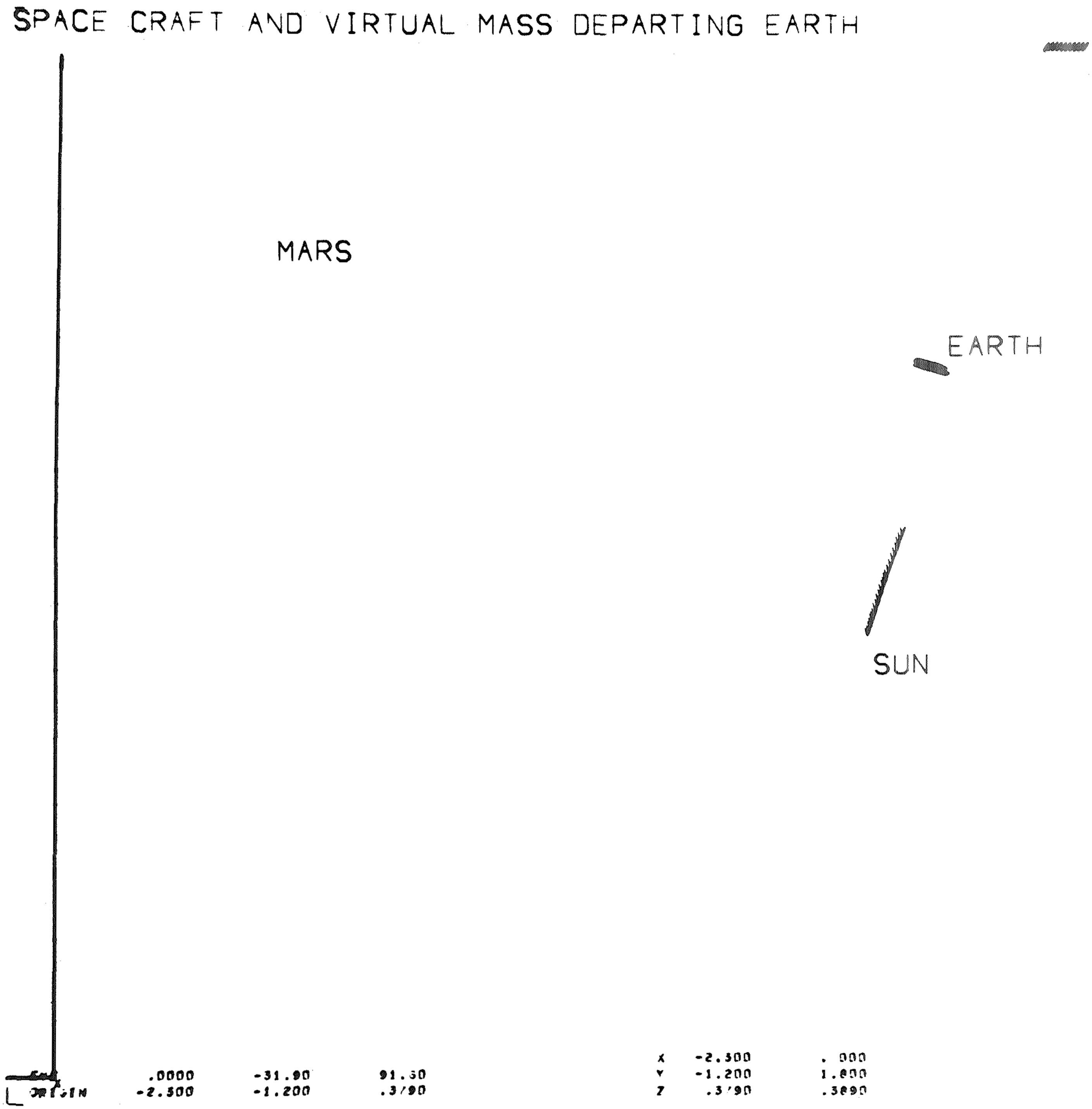


FIGURE 5 - EARTH TO MARS TRAJECTORY: VIEW 1

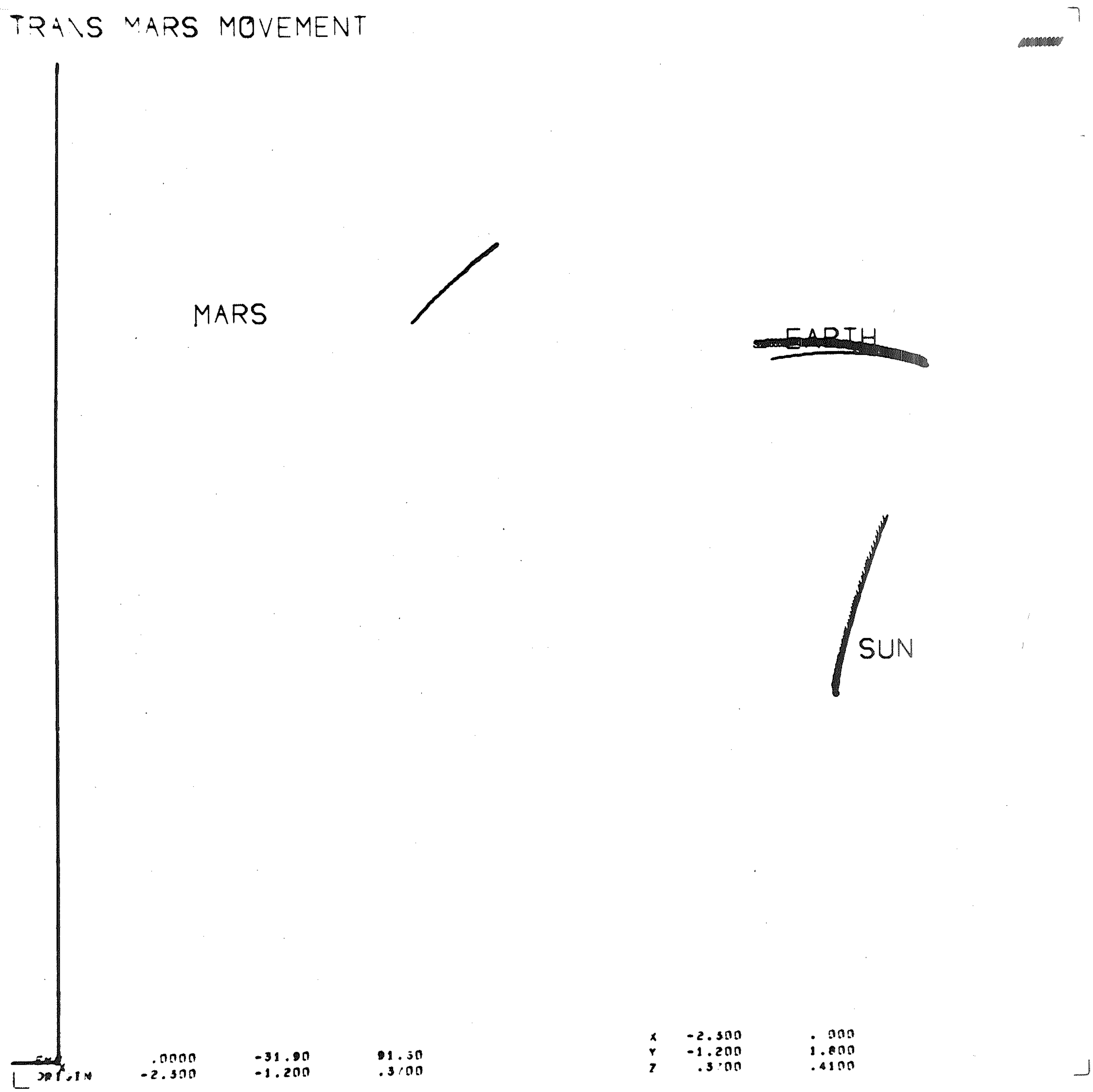


FIGURE 6 - EARTH TO MARS TRAJECTORY: VIEW 2

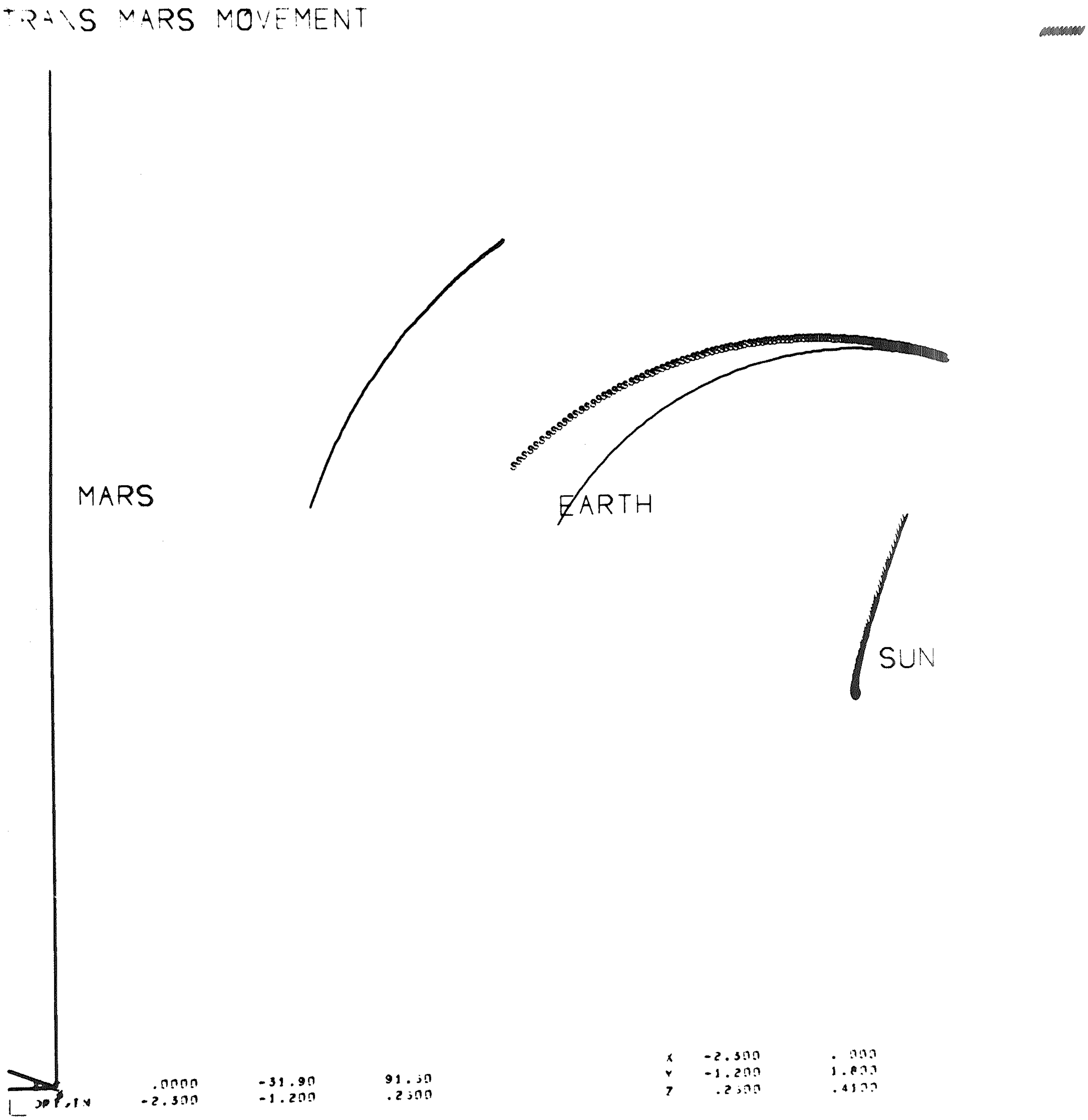


FIGURE 7 - EARTH TO MARS TRAJECTORY: VIEW 3

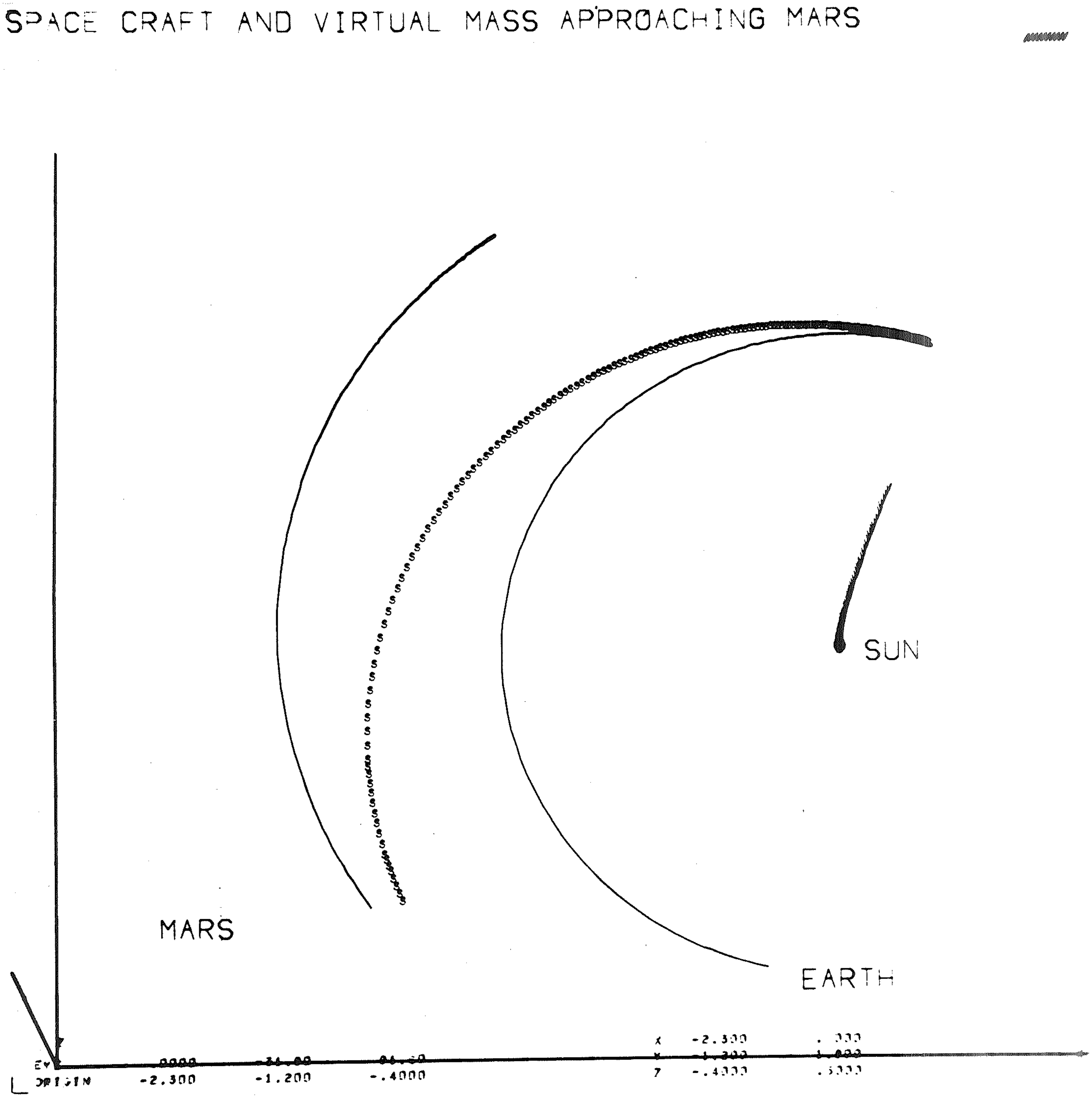


FIGURE 8 - EARTH TO MARS TRJAECTORY: VIEW 4

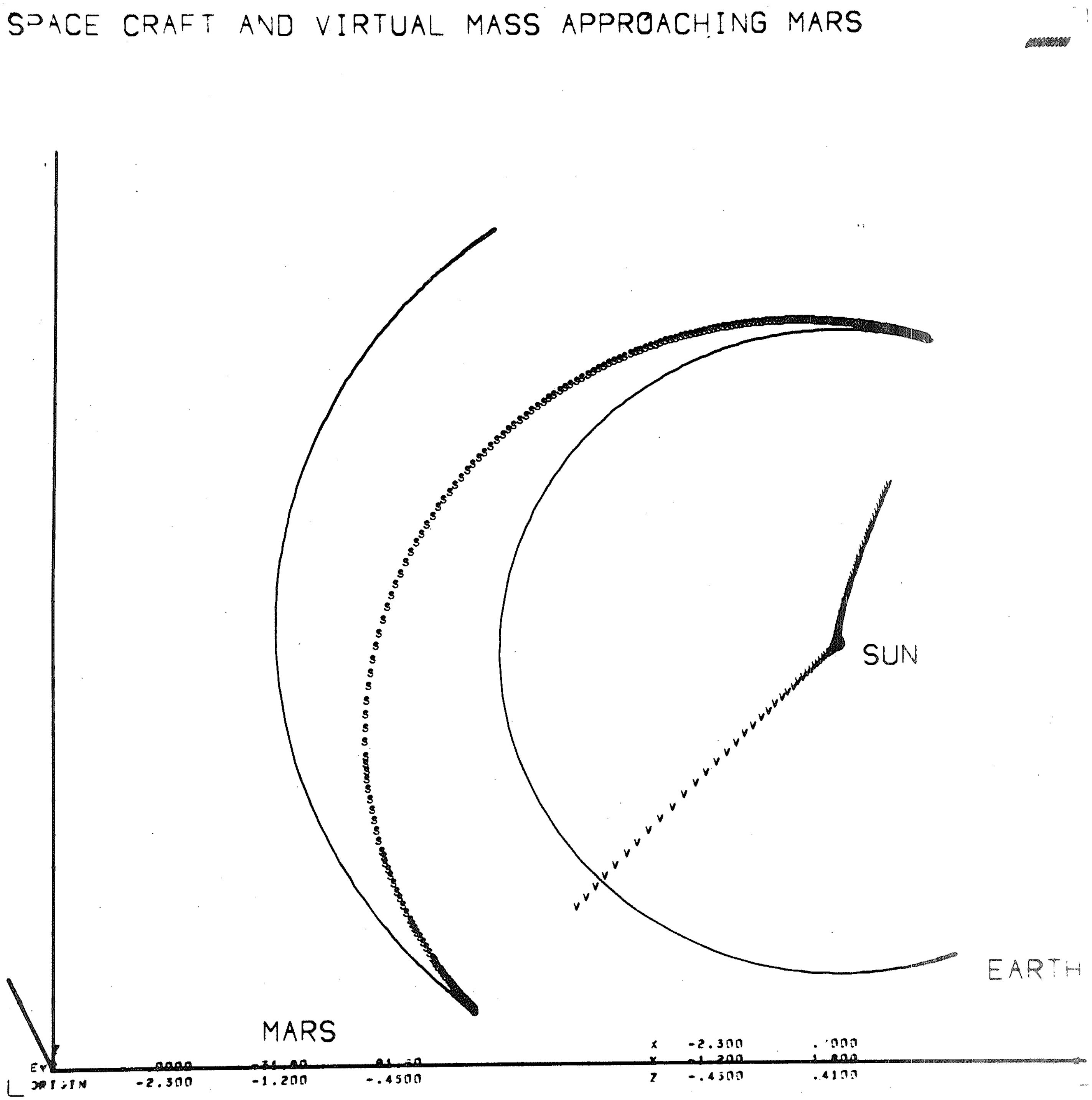


FIGURE 9 - EARTH TO MARS TRAJECTORY: VIEW 5

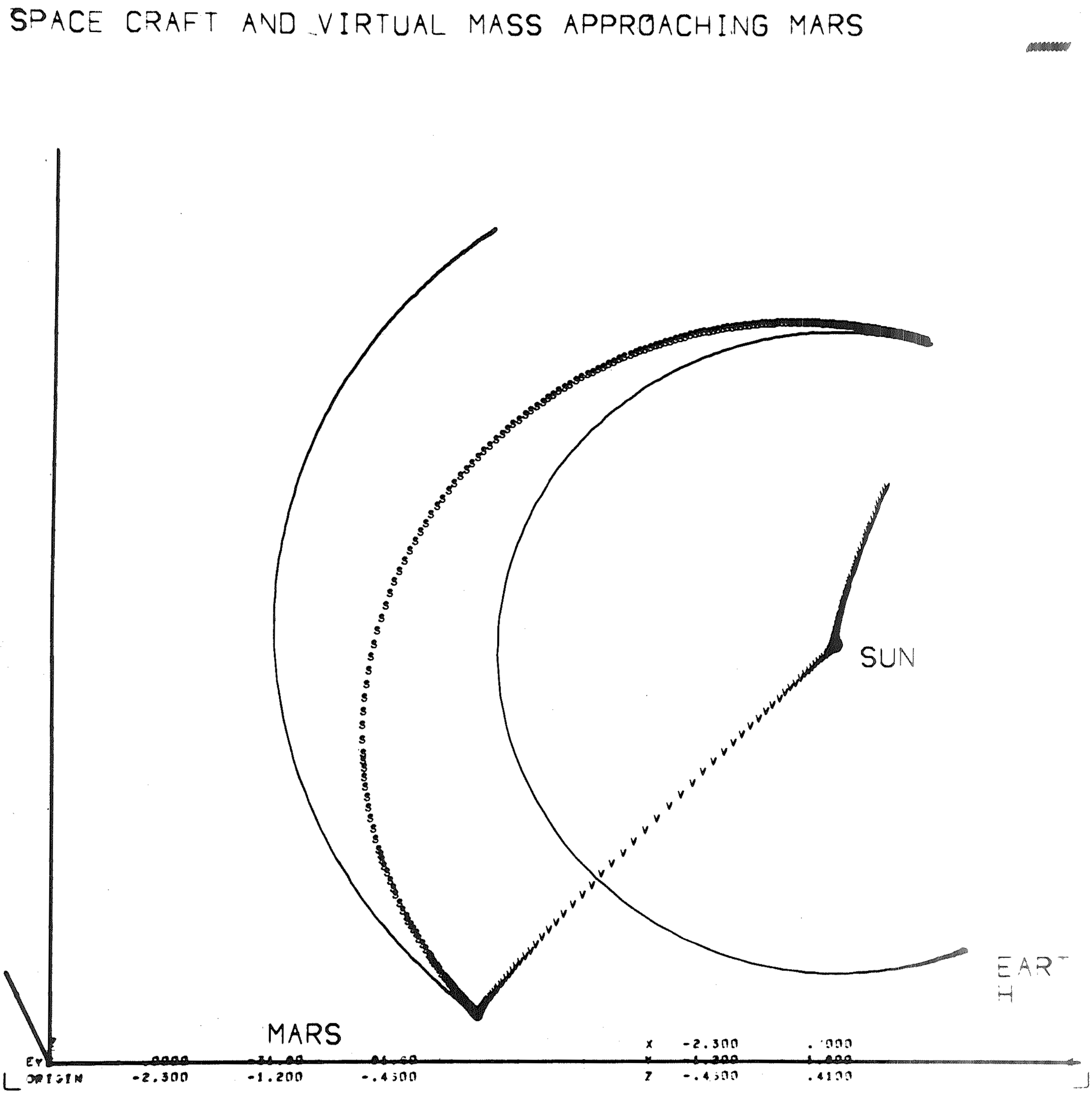


FIGURE 10 - EARTH TO MARS TRAJECTORY: VIEW 6

Figure 11

Restricted 3-Body Trajectory Instructions

```
VMASS*RFJK.RUN300
      @ HDG    V MASS EARTH-MOON WITH EPHPCD / R JESSUP
      @ PDP,FLI @EVENT/RUN300
      @EVENT PROC
      C       REFERENCE:  D H NOVAK, ''VIRTUAL MASS TECHNIQUE FOR COMPUTING
      C       SPACE TRAJECTORIES,'' ER 14045, CONTRACT NAS 9-4370,
      C       MARTIN COMPANY, JANUARY 1966.
              CALL EVNACC ('START', F0, 'START', TTRE, F0, $310)
      C       EVENT START ONE TIME CALCULATIONS.  CONVERT INPUT TO DIMENSIONLESS
      C       UNITS FOR COMPATABILITY WITH 2-BODY EPHEMERIS
                  G02 = .01214328900
                  G01 = F1 - G02
                  RUNIT = 207747.2D0
                  TUNIT = RTOD / .5490149300
                  VUNIT = RUNIT / TUNIT
                  CALL SCALE (F1 / RUNIT, RSCDDK, RSCEIN)
                  CALL SCALE (F1 / VUNIT, VSCDDK, VSCEIN)
      C       INITIALIZE SIMULATOR
                  TEE = - 93.591177D0 / TUNIT
                  TUTODA = TUNIT / F24
                  CALL EPHEM
                  CALL VMXACT
                  LPRNOW = .TRUE.
                  GO TO 9915
        310   CALL EVNACC ('MDCORS', 'START', 'MDCORS', 70.D0, TTREHR, $320)
                  GO TO 9915
        320   CALL EVNACC ('CUTOFF', 'MDCORS', 'CUTOFF', A2, F0, $340)
                  IEVSTP = IEVCUR
                  LPRNOW = .FALSE.
                  GO TO 9915
        340    CONTINUE
        END
      @ PDP,FLI @AUXEQ/RUN300
      @AUXEQ PROC
               TTREHR = TTRE * TUNIT
               ENERGY = (F2 * (G01 / RSCD01 + G02 / RSCD02)
             1         - F2 * (RSCEIN(2) * VSCEIN(1)  -  RSCEIN(1) * VSCEIN(2))
             2          - DOT (VSCEIN, VSCEIN)) * VUNIT * VUNIT
      C        AUXILLIARY EQUATIONS FOR CALCULATIONS CONTROLLING EVENTS MDCORS
      C        AND CUTOFF
               IF (IEVCUR .EQ. 'START') RETURN
               IF (IEVCUR .EQ. 'MDCORS') A2 = DOT (RSCE02, VSCE02)
        END
      @ PDP,FLI @EOCEQ/RUN300
      @EOCEQ PROC
        END
      @ PDP,FLI @PRINT/RUN300
      @PRINT PROC
      C        PRINT SPACE CRAFT STATE IN INPUT UNITS
               NAME LIST / PRT001 / INCNTR, TTREHR, ENERGY, RSCXIN, VSCXIN
               CALL SCALE (RUNIT, RSCEIN, RSCXIN)
               CALL SCALE (VUNIT, VSCEIN, VSCXIN)
               WRITE (IOPRT, PRT001)
        END
      @ PDP,FLI @STORE/RUN300
      @STORE PROC
               EQUIVALENCE            (VAC(1), TTREHR),   (VAC(2), A1     ),
```

```
      1  (VAC(3), A2     ),   (VAC(4), RUNIT ),   (VAC(5), TUNIT ),
      2  (VAC(6), VUNIT ),    (VAC(7), TEE   ),   (VAC(8), ENERGY),
      3  (VAC(9), RSCXIN),    (VAC(10), VSCXIN),
      4  (G00   , G01   ),    (G00(20)  , G02   ),
      5  (RSCE00, RSCE01),    (RSCE00(22), RSCE02),
      6  (VSCE00, VSCE01),    (VSCE00(22), VSCE02),
      7  (RSCD00, RSCD01),    (RSCD00(22), RSCD02)
       DIMENSION RSCXIN(3), VSCXIN(3)
   END
  @ COPY,S LVMASS.COMMON,COMMON
  @ COPY,S LVMASS.SIMXXX,SIMXXX
  @ COPY,S LVMASS.PRTXXX,PRTXXX
  @ COPY,S LVMASS.EPH2C0,EPH2C0
  @ FOR,W  SIMXXX,SIMXXX
  @ FOR,W  PRTXXX,PRTXXX
  @ FOR,W  EPH2C0,EPH2C0
  @ MAP    LVMASS.MAP,MAP300
  @ XQT    MAP300
   $DATADK
    RSCDDK = -1126.088D0, -5433.0951D0,   195.9727D0,
    VSCDDK = 18364.879D0,  3152.5321D0, 10624.889D0,
    LSTDDK = .TRUE.,
   $END
```

Figure 12

Restricted 3-Body Trajectory Printout

```
 V MASS EARTH-MOON WITH EPH2CO / R JESSUP

VIRTUAL MASS TRAJECTORY 15 SEP 70 10:17:34

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 10:17:35


$PRT001
INCNTR  =              +0
TTREHR  =         .000000000000000000D+000
ENERGY  =         .703398973878472816D+007
RSCXIN  =        -.112608800000000000D+004,        -.543309509999999998D+004,
                  .195972700000000000D+003
VSCXIN  =         .183648789999999999D+005,         .315253209999999999D+004,
                  .106248890000000000D+005

$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 10:17:45


$PRT001
INCNTR  =            +482
TTREHR  =         .700000000000000001D+002
ENERGY  =         .703398968736980931D+007
RSCXIN  =        -.778666752334666350D+003,         .206033724300051233D+006,
                  .156425026016198595D+003
VSCXIN  =         .174608203928771911D+004,         .162831802859940714D+004,
                 -.401880706991815647D+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 10:17:45


$PRT001
INCNTR  =            +482
TTREHR  =         .700000000000000001D+002
ENERGY  =         .703398968736980931D+007
RSCXIN  =        -.778666752334666350D+003,         .206033724300051233D+006,
                  .156425026016198595D+003
VSCXIN  =         .174608203928771911D+004,         .162831802859940714D+004,
                 -.401880706991815647D+003

$PRT001
INCNTR  =            +533
TTREHR  =         .703387527405850402D+002
ENERGY  =         .703398968893849843D+007
RSCXIN  =         .556258210891710836D-001,         .206373036471956116D+006,
                  .154723814284322860D-001
VSCXIN  =         .269323809276366216D+004,         .884395020550645450D-001,
                 -.504434140338037887D+003

$END
```

Figure 13

Restricted 3-Body Trajectory with Graphics

```
VMASS*RFJK.RUN310
@ HDG    V MASS EARTH-MOON WITH EPHOCO AND OVERLAYS / R JESSUP
@ PDP,FLI @EVENT/RUN310
@EVENT PROC
C      REFERENCE:  D H NOVAK, ''VIRTUAL MASS TECHNIQUE FOR COMPUTING
C      SPACE TRAJECTORIES,'' ER 14045, CONTRACT NAS 9-4370,
C      MARTIN COMPANY, JANUARY 1966.
       CALL EVNACC ('START', F0, 'START', TTRE, F0, $310)
C      EVENT START ONE TIME CALCULATIONS.  CONVERT INPUT TO DIMENSIONLESS
C      UNITS FOR COMPATABILITY WITH 2-BODY EPHEMERIS
          G02 = .01214328900
          G01 = F1 - G02
          RUNIT = 207747.2D0
          TUNIT = RTOD / .5490149300
          VUNIT = RUNIT / TUNIT
          CALL SCALE (F1 / RUNIT, RSCDDK, RSCEIN)
          CALL SCALE (F1 / VUNIT, VSCDDK, VSCEIN)
C      INITIALIZE SIMULATOR
          TEE = - 93.5911770D0 / TUNIT
          TUTODA = TUNIT / F24
          CALL EPHEM
          CALL VMXACT
          IEVREP = IEVCUR
          IENVA = 'ENVAR' + IRUN
          ISTEP = 'STEPS' + IRUN
          ITTRE = 'TTREHR' + IRUN
          IRUN = IRUN + 1
          DO 112 I = 1, 3
             CALL CULOUT ('RVMEUW' + I)
             CALL CULOUT ('R02EUW' + I)
  112        CALL CULOUT ('RSCEUW' + I)
          LPRNOW = .TRUE.
          GO TO 9915
  310  CALL EVNACC ('MDCORS', 'START', 'MDCORS', 70.D0, TTREHR, $320)
          CALL EOCEQ
          GO TO 9915
  320  CALL EVNACC ('CUTOFF', 'MDCORS', 'CUTOFF', A2, F0, $340)
          CALL EOCEQ
          IEVSTP = IEVCUR
          LPRNOW = .FALSE.
C      PLOT RESULTS
          CALL PTX ('RSCEUX', 'RSCEUY')
          REAL EYE(3) / .1E6, .15E8, .1E7 / ORIGIN(3) / 3*0. /
          CALL Q3DORG (ORIGIN, 1)
          CALL Q3DEYE (EYE, 1)
          CALL Q3DSHA (.0, 3)
          CALL PLTSIM (1)
          CALL PLTICS ('RSCEUX', -2000., .15E5)
          CALL PLTICS ('RSCEUY', -.1E5, .21E6)
          CALL PLTICS ('RSCEUZ', -100., 9100.)
          CALL Q3DPER ('RSCEUX', 'RSCEUY', 'RSCEUZ')
          CALL Q3DSHA (0., 0)
          CALL Q3DPER ('RVMEUX', 'RVMEUY', 'RVMEUZ')
          CALL Q3DPER ('R02EUX', 'R02EUY', 'R02EUZ')
          IF (IRUN .NE. 1) GO TO 314
             X0 = RSCEIN(1)
             Y0 = RSCEIN(2)
```

```
                    Z0 = RSCEIN(3)
                    GO TO 316
      314       T1 = (X0-RSCEIN(1))**2 + (Y0-RSCEIN(2))**2 + (Z0-RSCEIN(3))**2
                CALL COLECT ('DEV', SNGL (SQRT (T1) * RUNIT) * 1.E+04)
                CALL COLECT ('P  ', SNGL (RTOPRF) * 1.E+15)
                CALL PLTOUT
                    CALL PLTDMP
      316       IF (.NOT. LSTDDK) GO TO 9915
                    CALL PLTSIM (1)
                    CALL PLTLOG (IENVA + 1 - IRUN, 1)
                    CALL PLTICS (IENVA + 1 - IRUN, 1.E+00, 1.E+10)
                    DO 324 I = 1, IRUN
                        ENCODE (K, 322) I
      322               FORMAT (I3)
                        CALL PLTAGS (K)
                        CALL PXY (ITTRE+I-IRUN, IENVA+I-IRUN)
      324               CALL QXY (ITTRE+I-IRUN, IENVA+I-IRUN)
                    CALL AWRITE (12, 500, 1024, 180, 15, 'SCALED BY 1.E12')
                    CALL PLTSIM (1)
                    CALL PLTLOG (ISTEP + 1 - IRUN, 1)
                    CALL PLTICS (ISTEP + 1 - IRUN, 1.E+00, 1.E+04)
                    DO 326 I = 1, IRUN
                        ENCODE (K, 322) I
                        CALL PLTAGS (K)
                        CALL PXY (ITTRE+I-IRUN, ISTEP+I-IRUN)
      326               CALL QXY (ITTRE+I-IRUN, ISTEP+I-IRUN)
                CALL PLTAGS (0)
                CALL PLTSIM (0)
                CALL PLTICS ('DEV', 1.E0, 1.E8)
                CALL PLTICS ('P  ', 1.E0, 1.E10)
                CALL PLTLOG ('DEV', 1)
                CALL PLTLOG ('P  ', 1)
                CALL QXY ('P', 'DEV')
                    CALL AWRITE (12, 500, 1024, 180, 15, 'SCALED BY 1.E04')
                    CALL AWRITE (550, 12, 1024, 090, 15, 'SCALED BY 1.E15')
                    CALL PLTEND
                    CALL EXIT
      340   CONTINUE
      END
     @ PDP,FLI QAUXEQ/RUN310
     QAUXEQ PROC
            TTREHR = TTRE * TUNIT
            ENERGY = (F2 * (G01 / RSCD01 + G02 / RSCD02)
          1          - F2 * (RSCEIN(2) * VSCEIN(1)  -  RSCEIN(1) * VSCEIN(2))
          2           - DOT (VSCEIN, VSCEIN)) * VUNIT * VUNIT
            IF (ENO .EQ. F0) ENO = ENERGY
     C      AUXILLIARY EQUATIONS FOR CALCULATIONS CONTROLLING EVENTS MDCORS
     C      AND CUTOFF
            IF (IEVCUR .EQ. 'START') RETURN
            IF (IEVCUR .EQ. 'MDCORS') A2 = DOT (RSCE02, VSCE02)
      END
     @ PDP,FLI QEOCEQ/RUN310
     QEOCEQ PROC
            DO 410 I = 1, 3
                CALL COLECT ('RSCEUW' + I, SNGL (RSCEIN(I) * RUNIT))
                CALL COLECT ('RVMEUW' + I, SNGL (RVMEIN(I) * RUNIT))
      410       CALL COLECT ('R02EUW' + I, SNGL (R02EIN(I) * RUNIT))
```

```
            CALL COLECT (ISTEP, FLOAT (INCNTR))
            CALL COLECT (ITTRE, SNGL (TTREHR))
            CALL COLECT (IENVA, ABS (SNGL ((ENERGY - ENO) / ENO)) * 1.E12)
      END
     @ PDP,FLI QPRINT/RUN310
     QPRINT PROC
            NAME LIST / PRT001 / INCNTR, TTREHR, ENERGY, RSCXIN, VSCXIN
            CALL SCALE (RUNIT, RSCEIN, RSCXIN)
            CALL SCALE (VUNIT, VSCEIN, VSCXIN)
            WRITE (IOPRT, PRT001)
      END
     @ PDP,FLI QSTORE/RUN310
     QSTORE PROC
            EQUIVALENCE                  (VAC(1), TTREHR),    (VAC(2), A1     ),
          1    (VAC(3), A2     ),        (VAC(4), RUNIT ),    (VAC(5), TUNIT ),
          2    (VAC(6), VUNIT ),         (VAC(7), TEE    ),   (VAC(8), ENERGY),
          3    (VAC(9), RSCXIN),         (VAC(10), VSCXIN),
          4    (G00    , G01    ),       (G00(22) , G02),     (R00EIN(22), R02EIN),
          5    (RSCE00, RSCE01),         (RSCE00(22), RSCE02),
          6    (VSCE00, VSCE01),         (VSCE00(22), VSCE02),
          7    (RSCD00, RSCD01),         (RSCD00(22), RSCD02)
            DIMENSION RSCXIN(3), VSCXIN(3), R02EIN(3)
      END
     @ COPY,S LVMASS.COMMON,COMMON
     @ COPY,S LVMASS.PRTXXX,PRTXXX
     @ COPY,S LVMASS.EPH2CO,EPH2CO
     @ COPY,S LVMASS.SIMXXX,SIMXXX
     @ FOR,W  SIMXXX,SIMXXX
     @ FOR,W  PRTXXX,PRTXXX
     @ FOR,W  EPH2CO,EPH2CO
     @ MAP    LVMASS.MAP,MAP310
     @ ASG,TH 31,T,TAPE
     @ XQT    MAP310
      $DATADK
       RSCDDK = -1126.088D0, -5433.0951D0,   195.9727D0,
       VSCDDK = 18364.879D0,  3152.5321D0, 10624.889D0,
       RTOPRF = 1.D-19,
      $END
      $DATADK
       RTOPRF = 1.D-15,
      $END
      $DATADK
       RTOPRF = 1.D-10,
      $END
      $DATADK
       LSTDDK = .TRUE.,
       RTOPRF = 1.D-05,
      $END
     @ ASG,TM PLOTFILE,T,PLOT
     @ 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.; COMPUTED 15 SEP 70 15:06:48

$PRT001
INCNTR  =            +0
TTREHR  =        .00000000000000000D+000
ENERGY  =        .70339897387847281 6D+007
RSCXIN  =       -.11260880000000000D+004,        -.54330950999999998D+004,
                 .19597270000000000D+003
VSCXIN  =        .18364878999999999D+005,         .31525320999999999D+004,
                 .10624889000000000D+005

$END
$PRT001
INCNTR  =            +500
TTREHR  =        .99781548271831308 1D+001
ENERGY  =        .70339897387493772 8D+007
RSCXIN  =        .12354547278314908 5D+005,       .50168130609317233 9D+005,
                 .90298347239500831 0D+004
VSCXIN  =       -.46636341875116502 3D+002,       .44095713844145966 1D+004,
                 .40459543436376376 3D+002

$END
$PRT001
INCNTR  =            +1000
TTREHR  =        .47922695235131735 6D+002
ENERGY  =        .70339897387161972 5D+007
RSCXIN  =        .24270149516747446 1D+004,       .16670508519592921 8D+006,
                 .44601795883254813 2D+004
VSCXIN  =       -.29800907873409878 8D+003,       .19681854041752019 0D+004,
                -.17335502539903207 0D+003

$END
$PRT001
INCNTR  =            +1500
TTREHR  =        .65361934711301909 6D+002
ENERGY  =        .70339897386648043 5D+007
RSCXIN  =       -.18708602790894960 2D+004,       .19789622645042710 1D+006,
                 .12614997976522215 4D+004
VSCXIN  =       -.10914659461991967 7D+003,       .16856992714691217 0D+004,
                -.20078768043929128 1D+003

$END
$PRT001
INCNTR  =            +2000
TTREHR  =        .59658981970577043 3D+002
ENERGY  =        .70339897386675101 4D+007
RSCXIN  =       -.12292919452755297 6D+004,       .20542337448882161 6D+006,
                 .27779500231604524 1D+003
VSCXIN  =        .99188284941023896 8D+003,       .18640681744537357 9D+004,
                -.32022950181327903 1D+003

$END
```

Figure 14 - Page 2

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

EVNDT  *** DIVERGENCE
IEV(  1)   MDCORS   MDCORS     1301      601      14      0                 .50846147-004             -.11102230-015

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
TTREHR  =         .70000000000000001D+002
ENERGY  =         .703398973866385495D+007
RSCXIN  =        -.778667189427662595D+003,      .206033724040885004D+006,
                  .156425015699904788D+003
VSCXIN  =         .174608169390297516D+004,      .162831826259571446D+004,
                 -.401880626635059031D+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
TTREHR  =         .70000000000000001D+002
ENERGY  =         .703398973866385495D+007
RSCXIN  =        -.778667189427662595D+003,      .206033724040885004D+006,
                  .156425015699904788D+003
VSCXIN  =         .174608169390297516D+004,      .162831826259571446D+004,
                 -.401880626635059031D+003

$END

EVNDT  *** 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
TTREHR  =         .703387528575119709D+002
ENERGY  =         .703398973866680190D+007
RSCXIN  =         .554193164678232946D-001,      .205373036399900176D+006,
                  .154372501815774756D-001
VSCXIN  =         .269323821566097462D+004,      .883268189010980038D-001,
                 -.504434041947894176D+003

$END
```

Figure 14 - Page 3

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

VIRTUAL MASS TRAJECTORY 15 SEP 70 15:07:33

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 15:07:33

$PRT001
INCNTR  =          +0
TTREHR  =     .00000000000000000D+000
ENERGY  =     .70339897387847281 6D+007
RSCKIN  =    -.11260880000000000D+004,      -.54330950999999998D+004,
              .19597270000000000D+003
VSCKIN  =     .18364878999999999D+005,       .31525320999999999D+004,
              .10524889000000000D+005

$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 15:07:46

$PRT001
INCNTR  =          +482
TTREHR  =     .70000000000000000D+002
ENERGY  =     .70339896873698226 0D+007
RSCKIN  =    -.77866675233461244 6D+003,     .20603372430005123 8D+006,
              .15642502601619242 2D+003
VSCKIN  =     .17460820392877674 2D+004,     .16283180285993911 0D+004,
             -.40188070699182028 1D+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 15:07:46

$PRT001
INCNTR  =          +482
TTREHR  =     .70000000000000000D+002
ENERGY  =     .70339896873698226 0D+007
RSCKIN  =    -.77866675233461244 6D+003,     .20603372430005123 8D+006,
              .15642502601619242 2D+003
VSCKIN  =     .17460820392877674 2D+004,     .16283180285993911 0D+004,
             -.40188070699182028 1D+003

$END
$PRT001
INCNTR  =          +500
TTREHR  =     .70143113754651986 9D+002
ENERGY  =     .70339896879150242 0D+007
RSCKIN  =    -.49422455487408344 2D+003,     .20624174400543313 6D+006,
              .95169428845074464 1D+002
VSCKIN  =     .22406836117839377 2D+004,     .12216564666655182 3D+004,
             -.45542877532069056 4D+003

$END

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 15:07:48
```

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

$PRT001
INCNTR  =          +533
TTREHR  =         .70338752740585025 7D+002
ENERGY  =         .70339896889385207 8D+007
RSCXIN  =         .55625821124009822 6D-001,          .20637303647195609 7D+006,
                  .15472381427619759 3D-001
VSCXIN  =         .26932380927636901 2D+004,          .88439502028753831 1D-001,
                 -.50443414033803999 5D+003

$END
```

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

VIRTUAL MASS TRAJECTORY 15 SEP 70 15:07:49

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 15:07:49

$PRT001
INCNTR  =              +0
TTREHR  =         .000000000000000000D+000
ENERGY  =         .703398973878472816D+007
RSCXIN  =        -.112608800000000000D+004,      -.543309509999999998D+004,
                  .195972700000000000D+003
VSCXIN  =         .183648789999999999D+005,       .315253209999999999D+004,
                  .106248890000000000D+005

$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 15:07:51

$PRT001
INCNTR  =             +87
TTREHR  =         .700000000000013320D+002
ENERGY  =         .703397555192838889D+007
RSCXIN  =        -.778516093995517420D+003,       .206033985036964683D+006,
                  .155468272053672665D+003
VSCXIN  =         .174614704832420726D+004,       .162815708826244468D+004,
                 -.401932189618918599D+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 15:07:51

$PRT001
INCNTR  =             +87
TTREHR  =         .700000000000013320D+002
ENERGY  =         .703397555192838889D+007
RSCXIN  =        -.778516093995517420D+003,       .206033985036964683D+006,
                  .155468272053672665D+003
VSCXIN  =         .174614704832420726D+004,       .162815708826244468D+004,
                 -.401932189618918599D+003

$END

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 15:07:52

$PRT001
INCNTR  =             +94
TTREHR  =         .703387138527520084D+002
ENERGY  =         .703398297735892491D+007
RSCXIN  =         .674712142667303863D-001,       .206373234359742416D+006,
                  .561439035440255210D-001
VSCXIN  =         .269290729442860722D+004,       .369141237017654740D+000,
                 -.504513753301538197D+003

$END
```

```
  V MASS EARTH-MOON WITH EPH2CO AND OVERLAYS / 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
 TTREHR  =          .0000000000000000000D+000
 ENERGY  =          .7033989738784728160D+007
 RSCXIN  =         -.1126088000000000000D+004,        -.5433095099999999980D+004,
                    .1959727000000000000D+003
 VSCXIN  =          .1835487899999999990D+005,         .3152532099999999990D+004,
                    .1062488900000000000D+005

 $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 15:07:55


 $PRT001
 INCNTR  =              +23
 TTREHR  =          .7000000092500606340D+002
 ENERGY  =          .7014419347735413750D+007
 RSCXIN  =         -.6362136900555752040D+003,         .2062051407861079530D+006,
                    .1738319962149533210D+003
 VSCXIN  =          .1822000202697742860D+004,         .1496194691678577850D+004,
                   -.4368215910185485760D+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 15:07:55


 $PRT001
 INCNTR  =              +23
 TTREHR  =          .7000000092500606340D+002
 ENERGY  =          .7014419347735413750D+007
 RSCXIN  =         -.6362136900555752040D+003,         .2062051407861079530D+006,
                    .1738319962149533210D+003
 VSCXIN  =          .1822000202697742860D+004,         .1496194691678577850D+004,
                   -.4368215910185485760D+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
 TTREHR  =          .7030114754545397890D+002
 ENERGY  =          .7015592991234619980D+007
 RSCXIN  =          .3761318634581082570D+002,         .2064856345040345470D+006,
                    .2439649905916025170D+002
 VSCXIN  =          .2517047207889472840D+004,         .1406548210692773670D+003,
                   -.5426687928920409240D+003

 $END
```

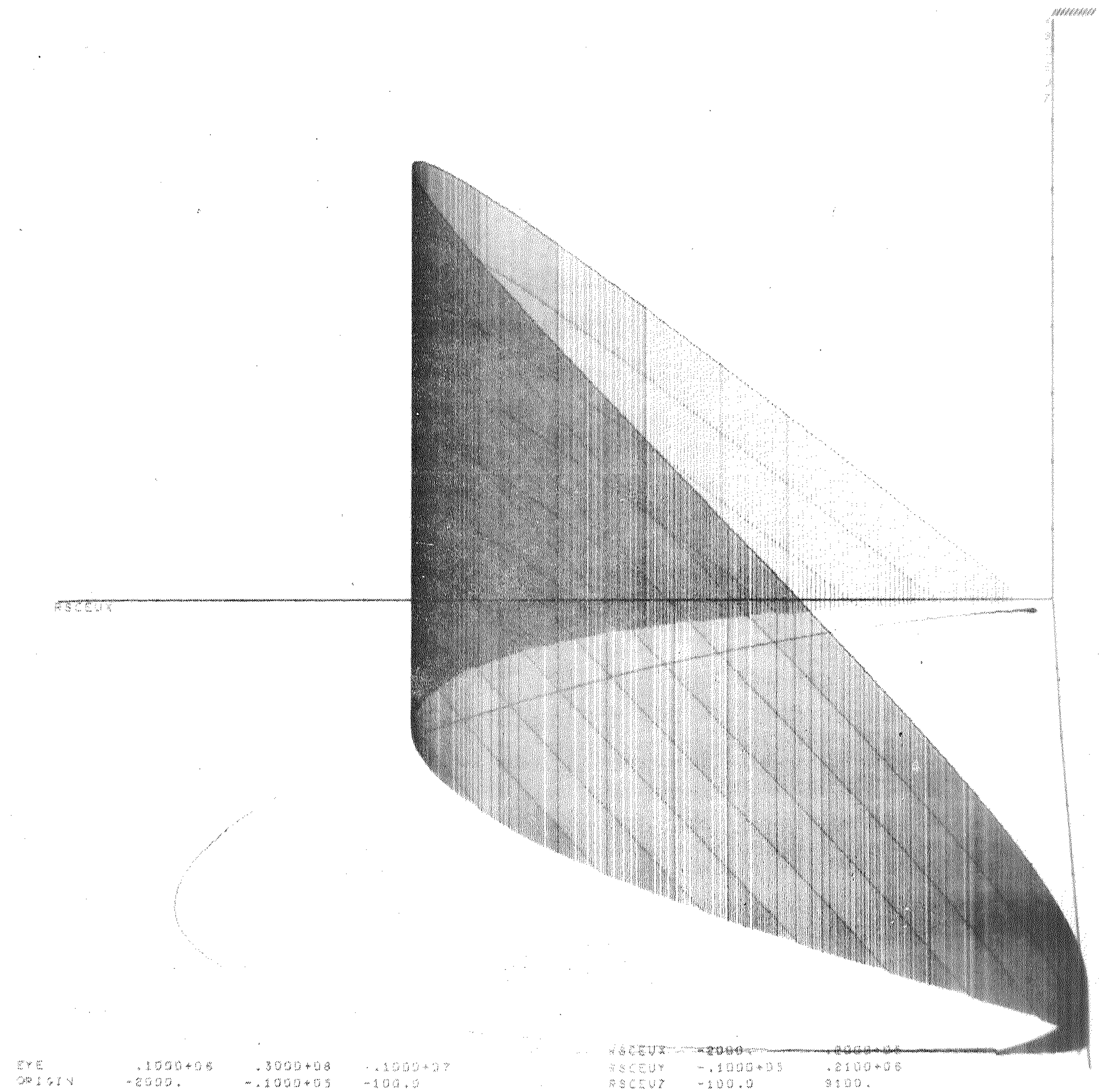


FIGURE 15 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF = 10^{-19})

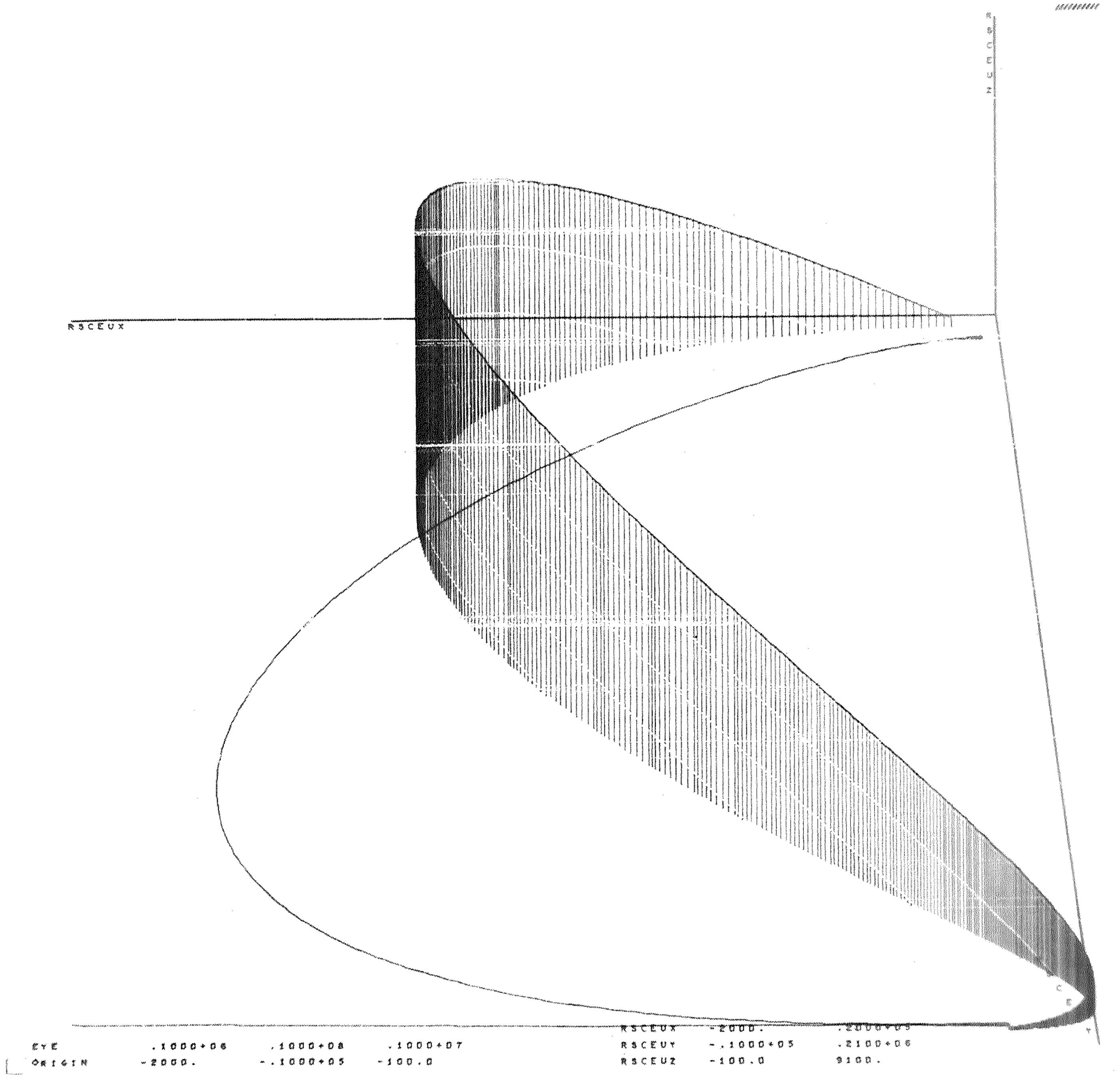


FIGURE 16 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF = 10^{-15})

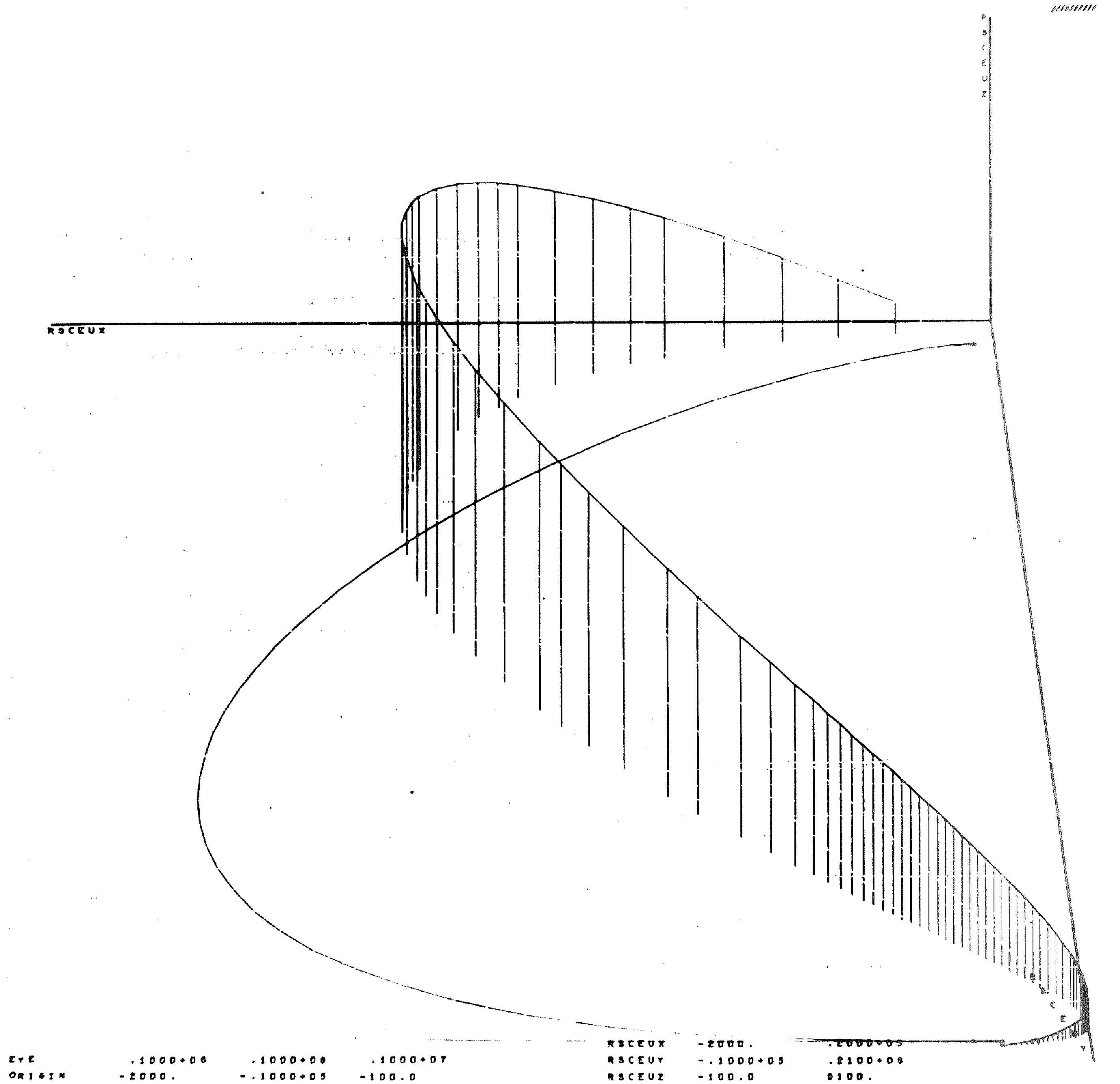


FIGURE 17 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF = 10^{-10})

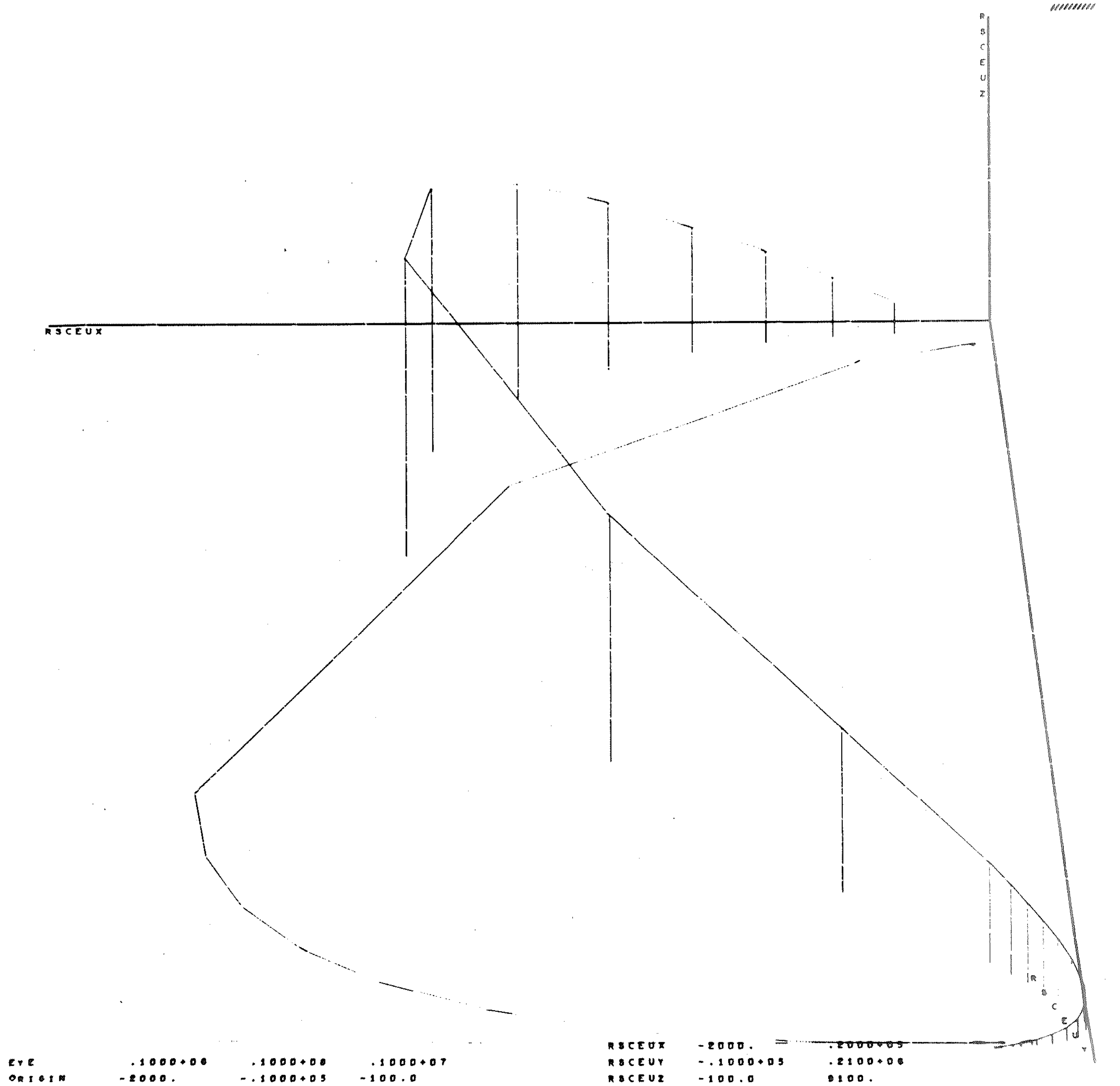


FIGURE 18 - RESTRICTED 3-BODY TRAJECTORY (RTOPRF = 10^{-5})

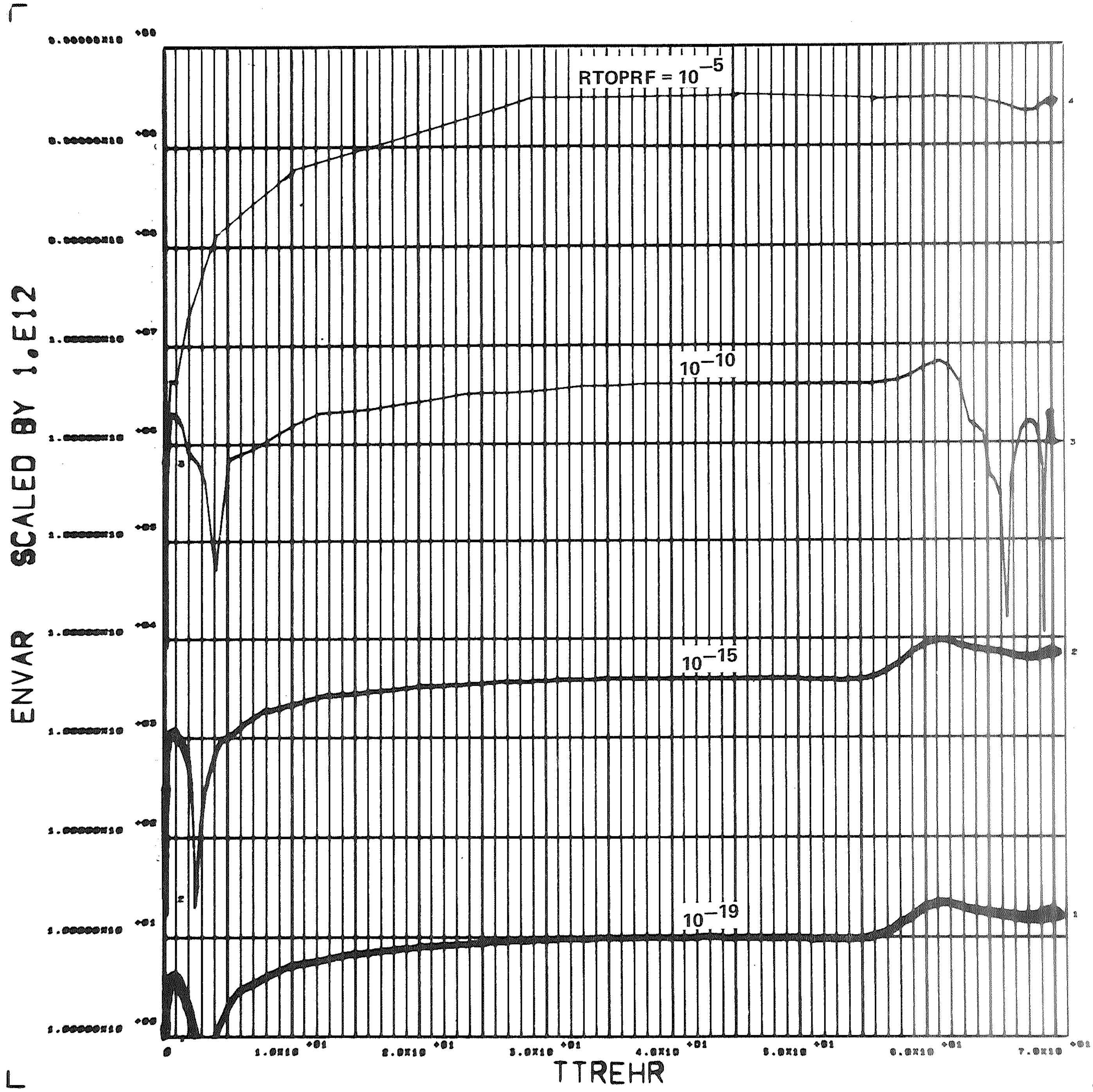


FIGURE 19 - JACOBI ENERGY VARIATIONS VS TRAJECTORY TIME

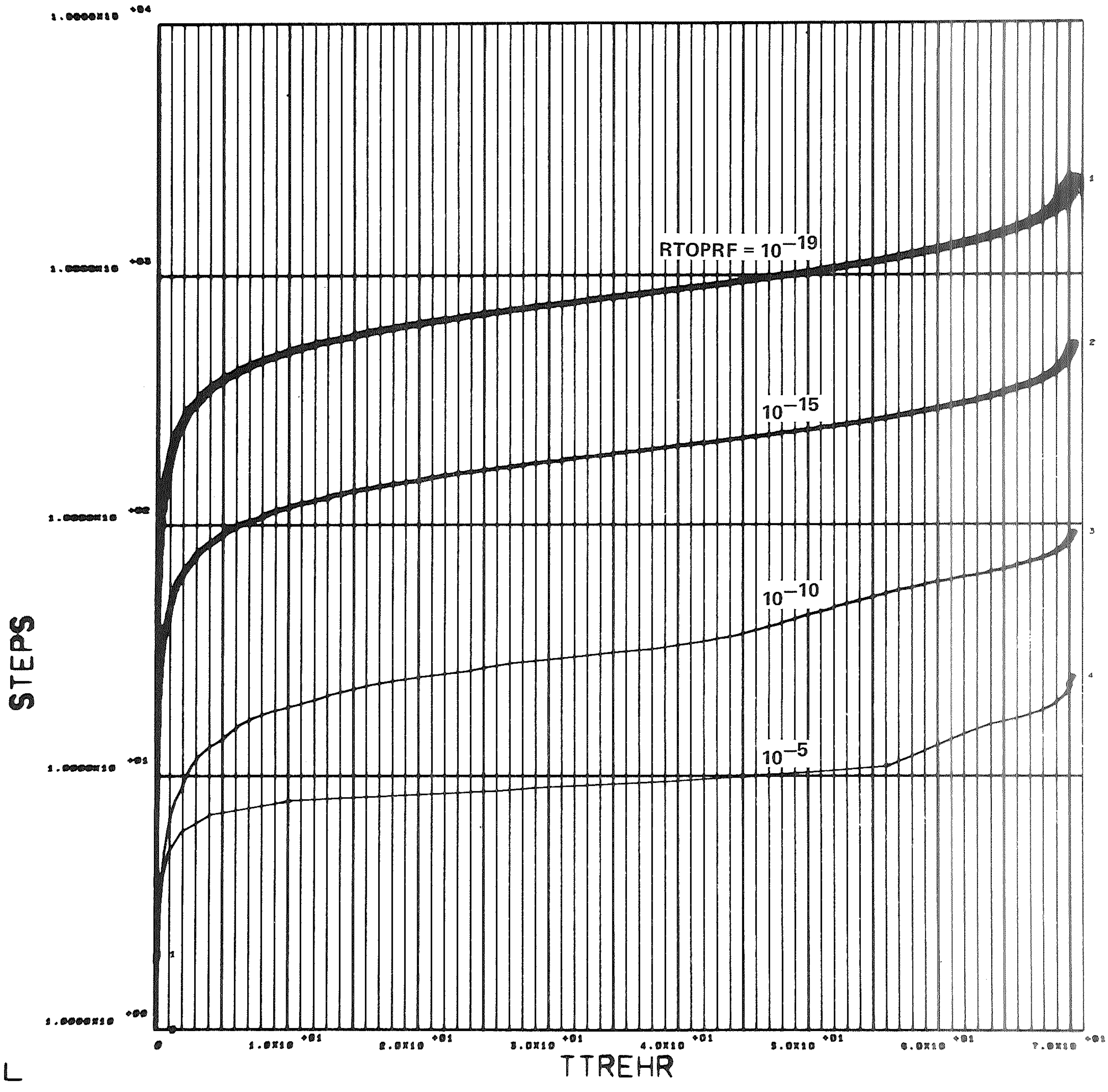


FIGURE 20 - COMPUTING STEPS VS TRAJECTORY TIME

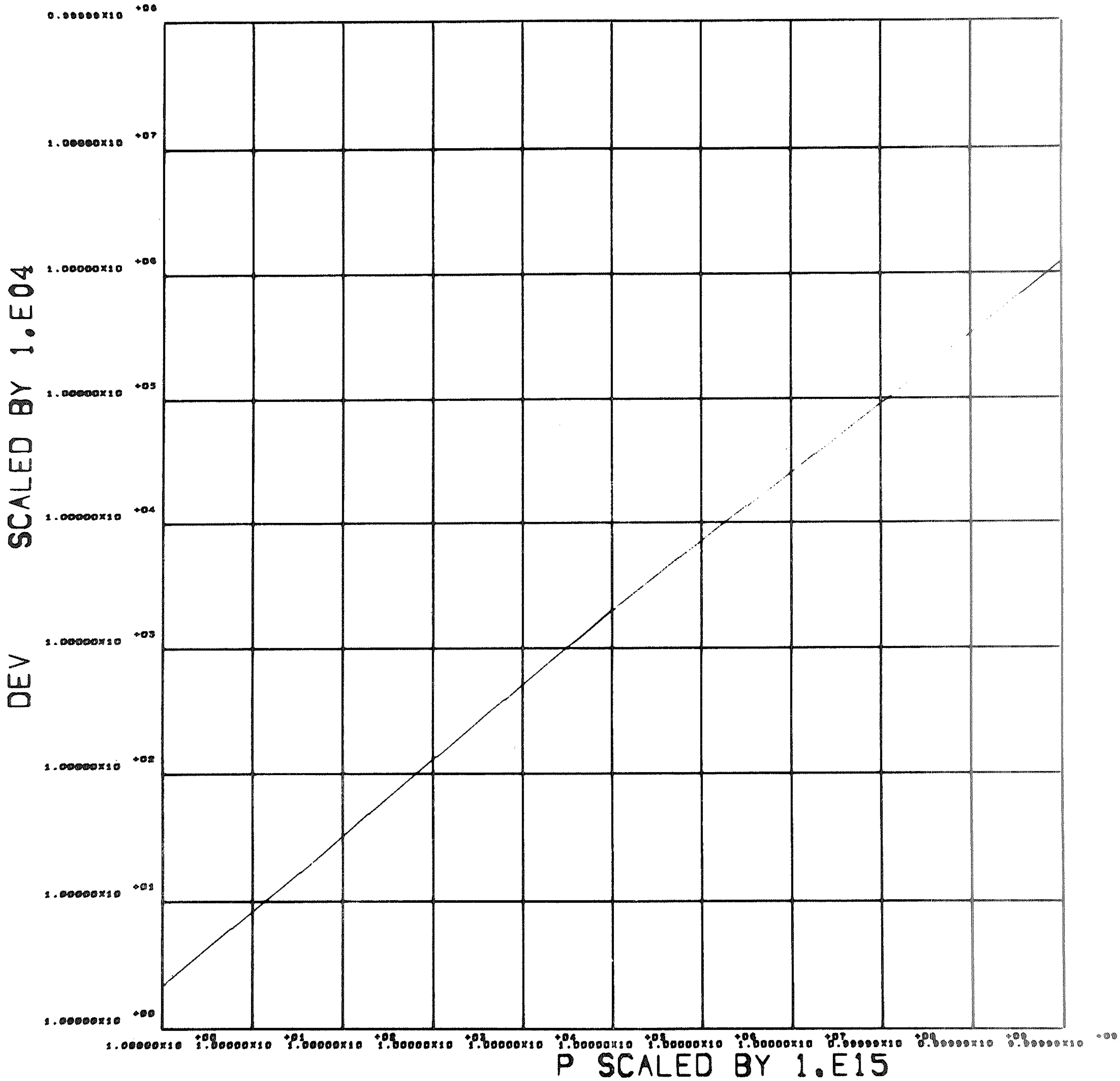


FIGURE 21 - FINAL POSITION VARIATION VS TRAJECTORY REFERENCE PRECISION

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.

Position	Character	Physical Quantity
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

Position	Character	System Corresponding to Physical Quantity
2,3	AT	Atomic time
	SC	Spacecraft
	VM	Virtual Mass
	EP	Ephemeris

Table 1 (Contd.)

Position	Character	System Corresponding to Physical Quantity
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
	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.)

Position	Character	Point at Which Quantity Evaluated
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.)

Position	Character	Reference Coordinate System	
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	(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	

Table 2

Definitions of VMASS COMMON Symbols

Symbol	Dimension	Equivalence	Definition	Appearances*
AU	-	$IETBL1_1$	Astronomical unit expressed in desired READE output units	DDREAD, LODCOM, READE
AUTOKM	-	FAC_{33}	Kilometers per AU	RUN100, RUN110,
CT2	-	VAR_{12}	Taylor's series term	KEPLER
DANEOO	1	$EPH_{17,1}$	Right ascension	PLANET
DLNEOO	1	$EPH_{19,1}$	Longitude of node at reference time	PLANET
DOBEOO	1	$EPH_{18,1}$	Obliquity	PLANET
DTAI	-	VAR_{15}	True anomaly increment	DDREAD, LODCOM, STEPIN
DTAIDN	-	VAR_{16}	Decrement factor for DTAI changes	DDREAD, LODCOM, STEPIN
DTAIMN	-	VAR_{17}	Minimum value for DTAI	DDREAD, LODCOM, STEPIN
DTAIUP	-	VAR_{18}	Increment factor for DTAI changes	DDREAD, LODCOM, STEPIN
DTOR	-	FAC_{31}	Radians per degree	LODCOM

*See also elements COMMON and PRTVAR

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
EMRAT	-	$IETBL1_7$	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 beginning of interval	
EVN	4,21	IEV_1	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_i	-	FAC_j	Floating point constant i, i = 0, 1, 2, ..., appearing throughout the VMASS system	
GOO	1	$EPH_{15,1}$	Array of gravitational constant times masses	EPH2CO, LODCOM, RUN300, RUN310, VMXACT

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
GOOR	1	$EPH_{16,1}$	Array of GOO data ratio to total mass	EPHEM, LODCOM
GVMB	-	VAR_{24}	Back value of GVME	CONIC, CONICT, INDEX, KEPLER, PREPAR
GVME			V.M. magnitude times Univ. Grav. Const. at beginning of interval	
HRCAVM	-	VAR_{27}	|HRCBVM|	CONIC, CONICT, KEPLER
HRCBVM	3	VAR_{95}	Back value of HRCEVM	CONIC, CONICT, KEPLER
HRCEVM			Reference conic angular momentum relative to V.M. at beginning of interval	
ICW	-	$IETBL2_{1}$	Indicator for initial call to ephemeris files	EPHEM, GETTA, LODCOM
ICENT	-	$IETBL2_{2}$	Indicator for central body in ephemeris data	EPHEM, LODCOM, READE
IDTRAJ	3	ISW_{42}	Input trajectory identification	DDREAD, PRTRAJ, SETUP
IEV	8,21	EVN_{1}	Events table, integer form	EVNACC, EVNDT, INDEX

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
IEVCUR	-	ISW_3	Current event	EVNACC, PRTIME, PRTRAJ, ROLLIO, RUN100, RUN110, RUN300, RUN310, SETUP
IEVFLG	-	ISW_4	Events table update flag	EVNACC, EVNDT, PRTRAJ, SIMXXX,
IEVM	-	ISW_1	Events table maximum size	EVNACC, LODCOM,
IEVN	-	ISW_2	Events table current size	EVNACC, EVNDT, INDEX
IEVREP	2	ISW_{45}	Events interval to be repeated	DDREAD, RUN100, RUN310, SETUP
IEVRUN	2	ISW_{47}	Events interval to be simulated	DDREAD
IEVSTP	-	ISW_6	Stopping event	DDREAD, PRTRAJ, RUN100, RUN110, RUN300, RUN310, SIMXXX
IEVSTR	-	ISW_7	Starting event	DDREAD, PRTRAJ, SIMXXX
INCNTR	-	ISW_9	Computing increment counter	INDEX, RECORD, RUN100, RUN110, RUN300, RUN310
IOCDR	-	IOU_1	Input card reader	DDREAD, LODCOM, SETUP
IOEPH	-	IOU_4	Ephemeris file	DDREAD, GETTA, LODCOM
IOMES	-	IOU_5	Diagnostic message printer	CONICT, DDREAD, EPHEM, EVNACC, EVNDT, KEPLER, LODCOM, PRTRAJ, SETUP, SIMXXX

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
IOPCH	-	IOU_3	Output card punch	DDREAD, LODCOM
IOPRL	-	IOU_7	Print later file	DDREAD, LODCOM, PRTRAJ, RECORD, ROLLIO
IOPRT	-	IOU_2	Output printer	EPHEM, LODCOM, PRTIME, RECORD, ROLLIO, RUN100, RUN110, RUN300, RUN310, SETUP, STEPIN
IOROL	-	IOU_8	Roll back file	DDREAD, LODCOM ROLLIO, SETUP
IOSCR	-	IOU_6	Scratch file	DDREAD, LODCOM
IOU	25	$KVAR_{951}$	Input-output file name array	DDREAD
IPLATM	-	ISW_{10}	Computing increments between writes to IOPRL	DDREAD, LODCOM, RECORD
IPLOTM	-	ISW_{12}	Computing increments between calls for plots	DDREAD, LODCOM, RUN110
IPNOWM	-	ISW_{13}	Computing increments between calls to PRTOUT	DDREAD, LODCOM, RECORD
IREQ	13	$IETBL2_3$	READE table of output requests	EPHEM, LODCOM, READE

Table 2 (Contd.)

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

Table 2 (Contd.)

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

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
LIMPEV	-	LSW_2	Flag showing iteration to an event	EVNDT, KEPLER, SIMXXX, STEPIN
LINTDN	-	LSW_{18}	Flag showing reduction of computing increment	EVNDT, STEPIN
LLDCOM	-	LSW_{14}	Flag showing COMMON is to be loaded	PRTRAJ, SETUP SIMXXX
LOGOUT	-	LSW_1	Flag showing subroutine log out on error condition	CONICT, DDREAD, EVNACC, KEPLER, ROLLIO, SETUP, SIMXXX, STEPIN
LOOP	-	LIMPEV	See LIMPEV	KEPLER
LRESKP	-	LSW_{16}	Flag showing by-pass of trajectory recording	EVNACC, RECORD
LSTDDK	-	LSW_4	Flag showing last data deck of a series has been read	DDREAD, RUN100, RUN110, RUN300, RUN310, SETUP
LSW	50	$KVAR_{901}$	Array of logical flags	DDREAD
LXECEV	-	LSW_3	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.)

Symbol	Dimension	Equivalence	Definition	Appearances
MVAR	-	ISW_{10}	Maximum size of array KVAR	LODCOM, SETUP
NMAX	-	ISW_{22}	Maximum number of terms in CONICT	CONICT, LODCOM
PHC	50	COMMON	COMMON array for physical constants	DDREAD
QVME	-	VAR_{36}	Current value virtual mass rate	INDEX, PREPAR, STEPIN, VMXACT
RE	-	$IETBL1_{3}$	Earth radius	DDREAD, LODCOM, READE
RETOKM	-	FAC_{34}	Kilometers per RE	LODCOM
RMNEOO	1	$EPH_{17,1}$	Planetographic mass point position	
ROOD	1	$EPH_{21,1}$	Gravitating body radius	
ROOEIN	1	$EPH_{1,1}$	Array of current gravitating bodies inertial positions	EPHEM, EPH2CO, PLANET, RUN100, RUN110, RUN310, VMXACT
RRCAVM	-	VAR_{38}	$\|RRCBVM\|$	CONIC, INDEX, PREPAR, STEPIN
RRCBVM	3	VAR_{110}	Reference conic position vector relative to the V.M. at beginning of computation interval	CONIC, CONICT, INDEX, KEPLER, PREPAR

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
RRCDVM	-	VAR_{39}	|RRCEVM|	INDEX, KEPLER, PREPAR, STEPIN
RRCEVM	3	VAR_{116}	Reference conic position vector relative to the V.M. at end of computation interval	INDEX, KEPLER, PREPAR
RRCIVM	3	VAR_{101}	RRCEVM increment (RRCEVM - RRCBVM)	CONICT, KEPLER
RSCAOO	1	$EPH_{13,1}$	Back values of RSCDOO	INDEX
RSCAVM	-	VAR_{38}	|RSCBVM|	
RSCBIN	3	VAR_{122}	Back value of RSCEIN	INDEX
RSCBVM	3	VAR_{110}	Back value of RSCEVM	
RSCDDK	3	VAR_{137}	Spacecraft position from data deck	DDREAD, RUN100, RUN110, RUN300, RUN310
RSCDOO	1	$EPH_{14,1}$	|RSCEOO|	INDEX, RUN300, RUN310, VMXACT
RSCDVM	-	VAR_{41}	|RSCEVM|	INDEX, STEPIN, VMXACT
RSCEIN	3	VAR_{128}	Inertial spacecraft position at end of computing interval	DDREAD, INDEX, RUN100, RUN110, RUN300, RUN310, STEPIN, VMXACT

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
RSCEVM	3		Spacecraft position relative to the V.M. at end of computing interval	
RSCEOO	1	$EPH_{7,1}$	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_{30}	Degrees per radian	LODCOM, RUN300, RUN310
RTOPRF	-	VAR_{47}	Position tolerance precision reference	DDREAD, LODCOM, RUN100, RUN110, RUN310, STEPIN
RVMBIN	3	VAR_{158}	Back value of RVMEIN	INDEX, PREPAR
RVMEIN	3	VAR_{164}	Inertial Virtual Mass position at end of computing interval	INDEX, PREPAR, RUN100, RUN110, RUN310, VMXACT
SPLEOO	1	$EPH_{20,1}$	Sidereal rate	PLANET
SV_i	3	TEM_j	Temporary vector storage	
S_i	-	TEM_j	Temporary scalar storage	

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
TABOUT	6,12	$IETBL4_1$	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_1	Current Julian day	EPHEM, PLANET, PRTIME, RUN100, RUN110
TEPEJS	-	VAR_2	Current Julian day fraction in seconds	EPHEM, PLANET PRTIME, RUN100, RUN110
TEPI	-	TTRI	See TTRI	KEPLER, STEPIN
TEPOJD	-	VAR_3	Initial Julian day	DDREAD, EPHEM, PLANET, PRTIME, RUN100, RUN110
TEPOJS	-	VAR_4	Initial Julian day fraction in seconds	DDREAD, EPHEM, PLANET, PRTIME, RUN100, RUN110
TEVI	-	VAR_9	Time increment to nearest event	EVNACC, EVNDT, STEPIN
TPD	-	$IETBL1_5$	Simulation time units per ephemeris day	DDREAD, LODCOM, READE

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
TTO	-	VAR_{53}	Time tolerance precision reference	CONICT, EVNDT, KEPLER, STEPIN
TTRB	-	VAR_{6}	Back value of TTRE	STEPIN
TTRE	-	VAR_{7}	Elapsed trajectory time at end of computing interval	EPHEM, EPH2CO, PRTIME, RUN300, RUN310, STEPIN
TTRI	-	VAR_{11}	TIRE increment	KEPLER, STEPIN
TUTODA	-	VAR_{54}	Ephemeris days per simulation time units	EPHEM, LODCOM, PRTIME, RUN300, RUN310
TV_i	3	TEM_j	Temporary vector storage	-
T_i	-	TEM_j	Temporary scalar storage	-
VAC	50		COMMON storage for user-coded trajectory descriptions	DDREAD, RUN100, RUN110, RUN300, RUN310
VAR	300	$KVAR_1$	COMMON storage for program working variables	DDREAD, DEVAR, EVNACC
VOOEIN	1	$EPH_{4,1}$	Inertial velocities of gravitating points at end of computing interval	EPHEM, EPH2CO, PLANET, RUN100, RUN110, VMXACT

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
VRCBVM	3	VAR_{113}	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_{125}	Back value of VSCEIN	INDEX
VSCBVM	3	VAR_{149}	Back value of VSCEVM	
VSCDDK	3	VAR_{140}	Spacecraft velocity from data deck	DDREAD, RUN100, RUN110, RUN300, RUN310
VSCEIN	3	VAR_{131}	Inertial spacecraft velocity at end of computing interval	DDREAD, INDEX, RUN100, RUN110, RUN300, RUN310, STEPIN, VMXACT
VSCEOO	1	$EPH_{10,1}$	Spacecraft velocity relative to gravitating body at end of computing interval	RUN100, RUN110, RUN300, RUN310, VMXACT
VSCEVM	3	VAR_{155}	Spacecraft velocity relative to the V.M. at end of computing interval	INDEX, PREPAR, STEPIN, VMXACT
VVMBIN	3	VAR_{161}	Back value of VVMEIN	INDEX, PREPAR

Table 2 (Contd.)

Symbol	Dimension	Equivalence	Definition	Appearances
VVMEIN	3	VAR_{167}	Inertial virtual mass velocity at end of computing interval	INDEX, PREPAR, VMXACT
XV_i	3	TEM_j	Temporary vector storage	-
X_i	-	TEM_j	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.

Table 4

VMASS Program Storage Requirements

Element	Storage Requirements		
	Code	Data	COMMON
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.)

Element	Storage Requirements Code	Data	COMMON
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.)

Element	Storage Requirements		
	Code	Data	COMMON
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.)

Element	Storage Requirements Code	Data	COMMON
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

Identification	Definition	Program Appearances
ENTRY	Location of subroutine 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*RFJK.COMMON

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

```
       END
      EVARIO PROC
      C
            DIMENSION IOU(25)
            EQUIVALENCE (KVAR(951), IOU)
      C
      C     IOCDR     ON-LINE CARD READER
      C     IOPRT     ON-LINE PRINTER
      C     IOPCH     ON-LINE CARD PUNCH
      C     IOPRL     PRINT LATER
      C     IOROL     UNIT CONTAINING EVENTS STATES FOR ''ROLL-BACK'' PURPOSES
      C     IOEPH     EPHEMERIS DATA FILE
      C     IOMES     ERROR AND PROGRAM CONTROL FLOW MESSAGES
      C     IOSCR     SCRATCH FILE - AVAILABLE FOR ANY LOCAL USE
      C
            EQUIVALENCE             (IOU(1), IOCDR),    (IOU(2), IOPRT),
           1   (IOU(3), IOPCH),     (IOU(4), IOEPH),    (IOU(5), IOMES),
           2   (IOU(6), IOSCR),     (IOU(7), IOPRL),    (IOU(8), IOROL)
       END
      EVARS PROC
      C
            EQUIVALENCE             (VAR(1), TEPEJD),   (VAR(2), TEPEJS),
           1   (VAR(3), TEPOJD),    (VAR(4), TEPOJS),
           2   (VAR(6), TTRB  ),    (VAR(7), TTRE  ),
           3   (VAR(9), TEVI  ),    (VAR(10), TRCI  ),  (VAR(11), TTRI  ),
           4   (VAR(12), CT2   ),
           5   (VAR(15), DTAI  ),   (VAR(16), DTAIDN),  (VAR(17), DTAIMN),
           6   (VAR(18), DTAIUP),
           7   (VAR(21), ERCAVM),
           8   (VAR(24), GVMB  ),   (VAR(25), GVME  ),
           9   (VAR(27), HRCAVM)
            EQUIVALENCE
           1
           2   (VAR(35), QVMB  ),   (VAR(36), QVME  ),
           3   (VAR(38), RRCAVM),   (VAR(39), RRCDVM),  (VAR(40), RSCAVM),
           4   (VAR(41), RSCDVM),
           5   (VAR(44), RTOPCM),
           6   (VAR(47), RTOPRF),
           7   (VAR(50), RUTOAU),
           8   (VAR(53), TTO   ),   (VAR(54), TUTODA),
           9   (VAR(56), VUTOAD)
            EQUIVALENCE (TEPI, TTRI), (TEPE, TTRE), (TEPB, TTRB),
           1   (RTO, RTOPRF)
       END
      EVARV PROC
            EQUIVALENCE
           1
           2
           3
           4   (VAR(92), ERCBVM),   (VAR(95), HRCBVM),
           5   (VAR(101), RRCIVM),
           6   (VAR(110), RRCBVM), (VAR(113), VRCBVM), (VAR(116), RRCEVM),
           7   (VAR(119), VRCEVM), (VAR(122), RSCBIN), (VAR(125), VSCBIN),
           8   (VAR(128), RSCEIN), (VAR(131), VSCEIN),
           9   (VAR(137), RSCDDK), (VAR(140), VSCDDK)
            EQUIVALENCE             (VAR(146), RSCBVM), (VAR(149), VSCBVM),
           1   (VAR(152), RSCEVM), (VAR(155), VSCEVM), (VAR(158), RVMBIN),
```

```
          2   (VAR(161), VVMBIN), (VAR(164), RVMEIN), (VAR(167), VVMEIN)
     C
           DIMENSION RSCBIN(3), RSCEIN(3), RVMBIN(3), VSCBIN(3), VSCEIN(3),
          1   VVMBIN(3), VVMEIN(3), RVMEIN(3),
          2          RSCBVM(3), VSCBVM(3), RSCEVM(3), VSCEVM(3), RRCBVM(3),
          3          VRCBVM(3), RRCIVM(3), HRCBVM(3), ERCBVM(3), RRCEVM(3),
          4   VRCEVM(3), RSCDDK(3), VSCDDK(3)
      END
     QEPH  PROC
     C
     C     THESE LABELS GIVE THE INSTANTANEOUS EPHEMERIS OF THOSE
     C     BODIES HAVING SIGNIFICANT EFFECT ON THE SPACECRAFT
     C
           COMMON / CEPH / EPH(21,20)
     C
           EQUIVALENCE                (EPH(01,1), ROOEIN), (EPH(04,1), VOOEIN),
          1   (EPH(07,1), RSCE00), (EPH(10,1), VSCE00), (EPH(13,1), RSCA00),
          2   (EPH(14,1), RSCD00), (EPH(15,1), GOO    ), (EPH(16,1), GOOR   ),
          3   (EPH(17,1), DANE00), (EPH(18,1), DOBE00), (EPH(19,1), DLNE00),
          4   (EPH(20,1), SPLE00), (EPH(21,1), ROOD   ),
          5   (EPH(17,1), RMNE00)
           DIMENSION     ROOEIN(3), VOOEIN(3), RSCE00(3), VSCE00(3),
          1   RSCA00(1), RSCD00(1), GOO(1)    , ROOD(1)   , DANE00(1),
          2   DOBE00(1), DLNE00(1), SPLE00(1), GOOR(1),
          3   RMNE00(3)
     C
     C  EPHEMERIS LAYOUT
     C
     C   EACH GRAVITATING POINT I (SEE READE: I = 1, ..., 11 FOR SOLAR SYS)
     C
     C     EPH(01,I)  ROOEIN   INERTIAL POSITION
     C     EPH(04,I)  VOOEIN   INERTIAL VELOCITY
     C     EPH(07,I)  RSCE00   SPACE CRAFT POSITION RELATIVE TO I
     C     EPH(10,I)  VSCE00   SPACE CRAFT VELOCITY RELATIVE TO I
     C     EPH(13,I)  RSCA00   PREVIOUS DISTANCE BETWEEN SPACE CRAFT AND I
     C     EPH(14,I)  RSCD00   CURRENT  DISTANCE BETWEEN SPACE CRAFT AND I
     C     EPH(15,I)  GOO      GRAVITATIONAL CONSTANT TIMES MASS OF I
     C     EPH(16,I)  GOOR     MASS RATIO OF I TO ALL THE PLANETS
     C
     C   EACH PLANET
     C
     C     EPH(17,I)  DANE00   RIGHT ASCENSION
     C     EPH(18,I)  DOBE00   OBLIQUITY
     C     EPH(19,I)  DLNE00   LONGITUDE OF NODE AT REFERENCE TIME
     C     EPH(20,I)  SPLE00   ANGULAR ROTATION RATE (SIDEREAL)
     C     EPH(21,I)  ROOD     RADIUS OF (ASSUMED SPHERICAL) I
     C
     C  EACH MASS POINT FROM PLANET MASS POINT REPRESENTATION
     C
     C     EPH(17,I)  RMNE00   PLANETOGRAPHIC POSITION
     C
           JKLOC(I) = 1 + JKSPEC * (I - 1)
     C
     C  NOTE:  LIST JK OF JKN ENTRIES GIVEN SUBSCRIPTS I OF GRAVITATING
     C         POINTS CONTRIBUTING TO CURRENT COMPUTATIONS
     C
      END
```

```
      EEPH  PROC
       END
      QMON  PROC
            COMMON / CMON / MONTH(12)
            DATA MONTH / 'JAN@01FEB@02MAR@03APR@04MAY@05JUN@06',
           1             'JUL@07AUG@08SEP@09OCT@10NOV@11DEC@12'
       END
      QFAC  PROC
      C
      C     THESE NUMERICAL FACTORS AND CONSTANTS ARE FOR SPACE SAVINGS AND
      C     FOR ASSURING MAXIMUM PRECISION IN DOUBLE PRECISION REPRESENTATIONS
      C
            COMMON / CFAC / FAC(58)
      C
            EQUIVALENCE                (FAC(1), F0),        (FAC(2), F9TNTH),
           1    (FAC(3), F1),          (FAC(4), F2),        (FAC(5), F3),
           2    (FAC(6), F4),          (FAC(7), F5),        (FAC(8), F6),
           3    (FAC(9), F7),          (FAC(10), F8),       (FAC(11), F9),
           4    (FAC(12), F10),        (FAC(13), F11),      (FAC(14), F12),
           5    (FAC(15), F13),        (FAC(16), F14),      (FAC(17), F15),
           6   (FAC(18), F24    ),  (FAC(19), F60    ),  (FAC(20), F1440 ),
           7   (FAC(21), F3600 ),   (FAC(22), F86400),   (FAC(23), FINITY),
           8   (FAC(24), FPT15 ),   (FAC(25), FPT4   ),  (FAC(26), F1PT5 ),
           9   (FAC(27), F2PT5 ),   (FAC(28), F20    ),  (FAC(29), F30    )
            EQUIVALENCE             (FAC(30), RTOD   ),  (FAC(31), DTOR   ),
           1   (FAC(32), F43200),   (FAC(33), AUTOKM),   (FAC(34), RETOKM),
           2   (FAC(35), FPT5  ),   (FAC(36), F001   ),  (FAC(37), F00001),
           3   (FAC(38), F1PT1 ),   (FAC(39), FTTO   ),  (FAC(40), FRTO   ),
           4   (FAC(41), FINTES),   (FAC(42), FTTOKM)
       END
      EFAC  PROC
       END
      QIEV  PROC
      C
      C     TABLE OF ''ACTIVE'' EVENTS --- SEE EVNACC FOR LAYOUT
      C
            DIMENSION IEV(8,20), EVN(4,20)
            EQUIVALENCE (KVAR(1001), IEV, EVN)
       END
      QPHC  PROC
      C
      C     THESE PHYSICAL CONSTANTS DO NOT CHANGE DURING THE COMPUTER RUN
      C
            COMMON / CPHC / PHC(50)
      C
            EQUIVALENCE              (PHC(01), GS     ),  (PHC(02), GMSUN ),
           1   (PHC(03), PIMASS)
            DIMENSION PIMASS(12)
       END
      EPHC  PROC
       END
      QREADE PROC
      C
      C     READ JPL EPHEMERIS AT THE JULIAN EPHEMERIS DATE                  READ0060
      C     GIVEN BY (JED+TSEC/86400.D0)                                     READ0070
      C                                                                      READ0080
      C**   ITEMS COMMUNICATED THROUGH THE CALLING SEQUENCE **               READ0090
```

```
C                                                                 READ0100
C     JED       REFERENCE JULIAN EPHEMERIS DATE.                  READ0110
C     TSEC      SECONDS OF EPHEMERIS TIME PAST JED.               READ0120
C               ANY COMBINATION OF VALUES OF JED AND TSEC         READ0130
C               IS ACCEPTABLE AS LONG AS (JED+TSEC/86400.D0)      READ0140
C               IS WITHIN THE RANGE OF THE EPHEMERIS TAPE         READ0150
C               BEING USED. HOWEVER TO OBTAIN THE                 READ0160
C               FINEST POSSIBLE RESOLUTION IN INTERPOLATION       READ0170
C               THE NUMBER JED MUST BE AN EXACT MACHINE           READ0180
C               NUMBER. FOR EXAMPLE JED COULD BE A DATE ENDING    READ0190
C               WITH .0 OR .5 .                                   READ0200
C     IERR      ERROR FLAG                                        READ0210
C               0=NO ERROR                                        READ0220
C               1=(JED+TSEC/86400.D0) LESS THAN FIRST DATE        READ0230
C                  ON TAPE                                        READ0240
C               2=(JED+TSEC/86400.D0) GREATER THAN LAST DATE      READ0250
C                  ON TAPE                                        READ0260
C               3=SOME IREQ(I) IS NOT 0,1, OR 2                   READ0270
C               4=ICENT   IS NOT IN THE RANGE 1 THRU 11           READ0280
C               5=ICW IS NOT 1,2, OR 3                            READ0290
C                                                                 READ0300
C** THE FOLLOWING ITEMS ARE INPUT THROUGH COMMON **                READ0310
C *  COMMON BLOCK CETBL1 *                                         READ0320
C     AU              A.U. EXPRESSED IN DESIRED OUTPUT UNITS      READ0330
C     RE              EQUATORIAL RADIUS OF EARTH IN DESIRED OUTPUT UNITS   READ0340
C                     RE IS USED TO SCALE THE LUNAR EPHEMERIS     READ0350
C     TPD                  DESIRED NUMBER OF TIME UNITS PER DAY   READ0360
C     EMRAT     EARTH MOON MASS RATIO. SUGGESTED VALUE=81.3D0     READ0370
C                                                                 READ0380
C     SUGGESTED VALUES FOR AU AND RE  DEPEND UPON                 READ0390
C     DESIRED OUTPUT UNITS AS FOLLOWS..                           READ0400
C     FOR OUTPUT IN EARTH RADII AU=23454.794001225117D0, RE =1.D0  READ0410
C     FOR OUTPUT IN KILOMETERS  AU=149598640.D0,        RE =6378.169D0 READ0420
C     FOR OUTPUT IN A.U.  AU=1.D0,  RE =4.2635207111508500D-5     READ0430
C                                                                 READ0440
C     SET TPD=86400.D0 FOR VELOCITY IN LINEAR UNITS PER SECOND.   READ0450
C     SET TPD=    1.D0 FOR VELOCITY IN LINEAR UNITS PER DAY.      READ0460
C                                                                 READ0470
C *  COMMON BLOCK CETBL2 *                                         READ0480
C     ICW       FLAG INDICATING STATUS OF COMMON BLOCKS REC2 AND CETBL3  READ0490
C               1 MEANS NEITHER BLOCKS CONTAIN VALID DATA         READ0500
C               2 MEANS BOTH BLOCKS CONTAIN VALID DATA            READ0510
C               3 MEANS REC2 IS VALID, CETBL3 IS NOT              READ0520
C               USER MUST SET ICW=1 BEFORE INITIAL CALL           READ0530
C     ICENTR    SPECIFIES CENTRAL BODY FOR COORDINATE             READ0540
C               TRANSLATION AS FOLLOWS..                          READ0550
C                1 MERC      5 JUP      9 PLUTO                   READ0560
C                2 VENUS     6 SAT     10 SUN                     READ0570
C                3 EARTH     7 URANUS  11 MOON                    READ0580
C                4 MARS      8 NEP                                READ0590
C     IREQ()    IREQ(J) SPECIFIES OUTPUT DESIRED FOR              READ0600
C               BODY NO. J.                                       READ0610
C               IREQ(J)=0 NO OUTPUT                               READ0620
C                       1 POSITION                                READ0630
C                       2 POSITION AND VELOCITY                   READ0640
C               J RUNS FROM 1 TO 11 AS FOLLOWS..                  READ0650
C                1 MERC      5 JUP      9 PLUTO                   READ0660
```

```
      C                 2 VENUS    6 SAT      10 SUN                            READ0670
      C                 3 EARTH    7 URANUS   11 MOON                           READ0680
      C                 4 MARS     8 NEP      12 ERTH-MN-BARYCENTER             READ0690
      C                                       13 NUTATION                       READ0700
      C *  COMMON BLOCK CETBL3 *                                                 READ0710
      C   TAB3 829 DOUBLE PREC. WORD BUFFER TO ACCOMODATE  J.D. AND EPHEMERIS.READ0720
      C   NUTAT 204 SINGLE PREC. WORD BUFFER TO ACCOMODATE NUTATION DATA.       READ0730
      C   CKSUM 1 S.P. WORD FOR CHECKSUM.                                       READ0740
      C** THE FOLLOWING ITEMS ARE OUTPUT THROUGH COMMON **                      READ0750
      C *  COMMON BLOCK CETBL4 *                                                 READ0760
      C    TABOUT( , )  PLANETARY AND LUNAR OUTPUT, SCALED AND                  READ0770
      C      TRANSLATED WITH RESPECT TO CENTER.                                 READ0780
      C                 TABOUT (I,J) CONTAINS OUTPUT FOR                        READ0790
      C                 BODY NO. J. (1 .LE. J .LE. 12)                          READ0800
      C                 THE INDEX I IDENTIFIES COMPONENTS AS FOLLOWS..          READ0810
      C                 1=X        2=Y      3=Z                                 READ0820
      C                 4=XDOT     5=YDOT   6=ZDOT                              READ0830
      C    NUT( )       NUTATION OUTPUT                                         READ0840
      C                 NUT(1)=DELTA LONGITUDE                                  READ0850
      C                 NUT(2)=DELTA OBLIQUITY                                  READ0860
      C                 NUT(3)=TIME DERIVATIVE OF NUT(1)                        READ0870
      C                 NUT(4)=TIME DERIVATIVE OF NUT(2)                        READ0880
      C *  COMMON BLOCK CETBL5 *                                                 READ0890
      C    BIVECT( , )  WORKING ARRAY. CONTENTS ARE INTERPOLATED                READ0900
      C                 AND SCALED BUT NOT TRANSLATED. 1ST INDEX RUNS           READ0910
      C                 OVER X,Y,Z,XDOT,YDOT,ZDOT AS IN TABOUT                  READ0920
      C                 BUT 2ND INDEX IS DIFFERENT AS FOLLOWS..                 READ0930
      C             BODIES 1 THRU 9 ARE HELIOCENTRIC.                           READ0940
      C                 1 MERC      5 JUP       9 PLUTO                         READ0950
      C                 2 VENUS     6 SAT      10 MOON REL TO EARTH             READ0960
      C                 3 ERTHMN    7 URANUS   11 ERTHMN REL TO EARTH           READ0970
      C                 4 MARS      8 NEP      12 ERTHMN REL TO MOON            READ0980
      C                                        13 SEE 4092+                     READ0990
      C                                                                         READ1000
      C     THE COMMON BLOCK 'CETBL9' IS FOR COMMUNICATION                      READ1010
      C     BETWEEN RDEP2 AND GETR2.                                            READ1020
            COMMON / CETBL1 / IETBL1(8)
           1       / CETBL2 / IETBL2(15)
           2       / CETBL3 / IETBL3(1836)
           3       / CETBL4 / IETBL4(148)
           4       / CETBL5 / IETBL5(156)
           5       / CETBL9 / IETBL9(7)
            INTEGER IREQ(13)
            REAL NUTAT(204), NUT(4), CKSUM
            DOUBLE PRECISION AU, JD1, JDIF, EMRAT, RE, TPD, TDAY,
           1   TAB3(829), TABOUT(6,12), BIVECT(6,13)
            EQUIVALENCE              (IETBL1(1), AU),     (IETBL1(3), RE),
           1   (IETBL1(5), TPD),     (IETBL1(7), EMRAT),
           2   (IETBL2(1), ICW),     (IETBL2(2), ICENT), (IETBL2(3), IREQ),
           3   (IETBL3(1), TAB3),    (IETBL3(1659),NUTAT), (IETBL3(1863),CKSUM),
           4   (IETBL4(1),TABOUT),   (IETBL4(145), NUT), (IETBL5(1), BIVECT),
           5   (IETBL9(1), JD1),     (IETBL9(3), TDAY),   (IETBL9(5), JDIF),
           6   (IETBL9(7), IERR1)
      C
       END
      EREADE PROC
       END
```

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

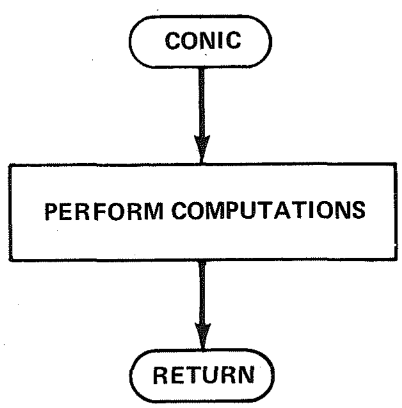
Computations: (See Table 2 for definition of symbols):

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

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

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

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



VMASS*RFJK.CONIC

```
      C
      C  PURPOSE
      C
      C     COMPUTES THE VECTOR ORBITAL ELEMENTS OF THE REFERENCE CONIC
      C     SECTION, GIVEN THE POSITION AND VELOCITY VECTORS RELATIVE TO THE
      C     VIRTUAL MASS.  OUTPUTS THESE VECTOR ELEMENTS IN THE VIRTUAL MASS
      C     CENTERED INERTIALLY ORIENTED COORDINATE FRAME.
      C
      C  INPUT
      C
      C     GVMB      PRODUCT OF UNIVERSAL GRAVITATION CONSTANT AND THE VIRTUAL
      C               MASS MAGNITUDE
      C     RRCBVM    THREE COMPONENTS OF THE REFERENCE CONIC
      C               POSTION RELATIVE TO THE VIRTUAL MASS
      C     VRCBVM    REF CONIC  VELOCITY RELATIVE TO THE VIRTUAL MASS
      C     RRCAVM    REF CONIC  DISTANCE FROM THE VIRTUAL MASS (MAGNITUDE OF
      C               RRCBVM)
      C
      C  OUTPUT
      C
      C     HRCBVM    ANGULAR MOMENTUM (AREAL RATE) VECTOR OF REFERENCE CONIC
      C               ORBIT RELATIVE TO THE VIRTUAL MASS
      C     HRCAVM    MAGNITUDE OF ANGULAR MOMENTUM VECTOR (HRCBVM)
      C     ERCBVM    ECCENTRICITY VECTOR OF CONIC ORBIT RELATIVE TO THE V MASS
      C     ERCAVM    MAGNITUDE OF ERCBVM
      C
            SUBROUTINE CONIC
      C
            INCLUDE QTYPE
            INCLUDE QVAR
            INCLUDE QTEM
            INCLUDE EVARS
            INCLUDE EVARV
            INCLUDE ETEM
      C
            CALL CROSS (RRCBVM, VRCBVM, HRCBVM)
            HRCAVM = VALUE (HRCBVM)
            CALL CROSS (HRCBVM, VRCBVM, TV1)
            DO 10 I = 1, 3
        10     ERCBVM(I) = -RRCBVM(I) / RRCAVM - TV1(I) / GVMB
            ERCAVM = VALUE (ERCBVM)
      C
            RETURN
            END
```

<u>Program</u>: CONICT

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

<u>Called by</u>: KEPLER

<u>Calls</u>: CROSS, DOT

<u>Errors and Diagnostics</u>: 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.

<u>Language</u>: Fortran subroutine

<u>Computations</u>:

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

$$T_2 = 1 - ERCAVM^2$$

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

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

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

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

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

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

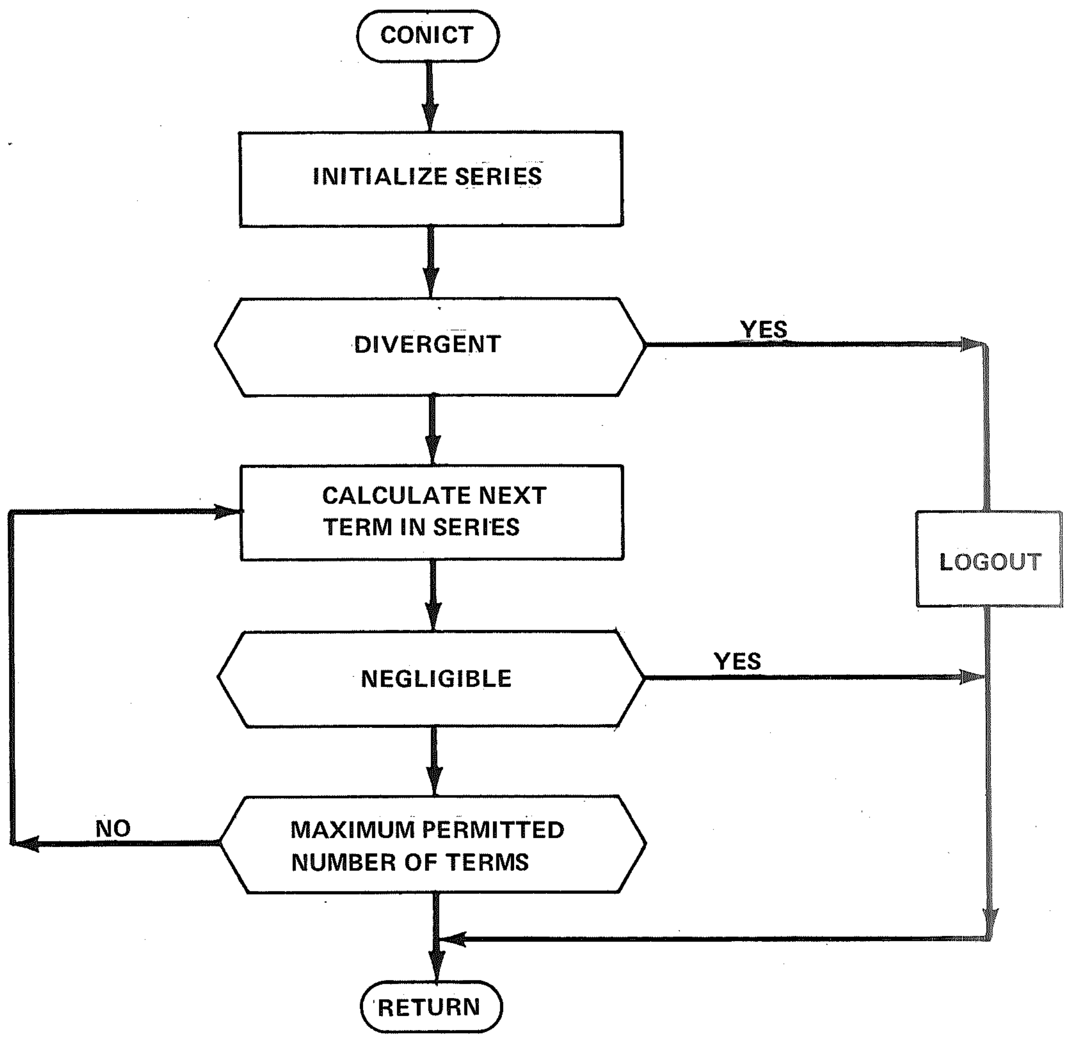
$$TRCI_0 = V_3 \circ RRCBVM$$

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

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

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

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



VMASS*RFJK.CONICT

```
C
C  PURPOSE
C
C      THIS PROGRAM COMPUTES THE REFERENCE CONIC TIME OF FLIGHT FROM
C      INITIAL POSITION VECTOR AND INCREMENTAL VECTOR TO DESIRED FINAL
C      POSITION ON ORBIT.  THE REFERENCE CONIC ORBIT IS SPECIFIED BY THE
C      VECTOR ORBITAL ELEMENTS.
C
C  INPUTS
C
C      HRCBVM   REFERENCE CONIC ANGULAR MOMENTUM (AREAL RATE)
C      HRCAVM   MAGNITUDE OF THE ANGULAR MOMENTUM
C      ERCBVM   REFERENCE CONIC ECCENTRICITY VECTOR
C      ERCAVM   MAGNITUDE OF ECCENTRICITY
C      RRCBVM   REFERENCE CONIC INITIAL POSITION VECTOR
C      RRCIVM   REFERENCE CONIC POSITION VECTOR INCREMENT
C      NMAX     MAXIMUM NUMBER OF TERMS IN SERIES EXPANSION FOR EVAL TIME
C      GVMB     PRODUCT OF UNIVERSAL GRAVITATION CONST AND VIRT MASS MAG
C      RTO      TOLERANCE DEFINING ACCURACY OF COMPUTED POS INCREMENT
C
C  OUTPUT
C
C      TRCI     REFERENCE CONIC INCREMENTAL TIME
C
      SUBROUTINE CONICT
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QTEM
      INCLUDE QFAC
      INCLUDE EVARS
      INCLUDE EVARV
      INCLUDE EVARIO
      INCLUDE EVARI
      INCLUDE EVARL
C
      DO 1 I = 1, 3
         TV1(I) = RRCIVM(I) / HRCAVM
 1       TV2(I) = HRCBVM(I) / HRCAVM
      CALL CROSS (TV1, TV2, TV3)
      TRCI = DOT (TV3, RRCBVM)
C
      T2  = F1 - ERCAVM ** 2
      T3  = GVMB * T2 / HRCAVM ** 2
      DO 29 I = 1, 3
 29      TV5(I) = ERCBVM(I) + RRCBVM(I) * T3
      T4  = DOT (TV5, TV3)
      T5 = ABS (T2) * (GVMB * T4 / HRCAVM) ** 2
C
C     CHECK FOR DIVERGENT TIME SERIES
C
      IF (T5 .LE. F1) GO TO 5
               WRITE (IOMES, 2)
 2             FORMAT (' CONICT ** TIME SERIES DIVERGENT ** ARG .GT. 1')
               LOGOUT = .TRUE.
               RETURN
```

```
    5      T6  = GVMB * T4 ** 3
           T7  = F1
           T9 = T2
           DO 20 N = 1, NMAX
                   T7 = T7 - T7 / (N + N)
                   T8 = T6 * T7 / (N + N + 1)
                   IF (ABS (T8) .LE. TTO) RETURN
                   IF (ABS (T8) .LE. FINTES * TRCI) RETURN
                   T9 = T9 * T2
                   TRCI = TRCI + SIGN (T8, T9)
   20              T6 = T6 * T5
    C
    C      TIME SERIES REQUIRES MORE THAN NMAX TERMS FOR CONVERGENCE
    C
           WRITE (IOMES, 30) NMAX
    30     FORMAT (' CONICT *** TIME SERIES TERMS EXCEEDING', I6)
           RETURN
           END
```

Program: CROSS (a, b, c)

Purpose: Calculate vector cross product.

Called by: CONIC, CONICT, KEPLER

Calls: None

Errors and Diagnostics: None

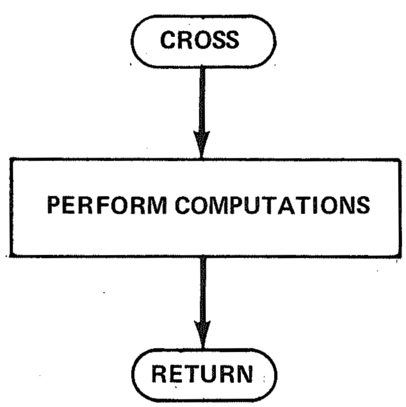
Language: Fortran subroutine

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

```
C
C  PURPOSE
C
C     CALCULATE VECTOR C THE CROSS PRODUCT OF VECTORS A AND B (3 COMP)
C
      SUBROUTINE CROSS (A, B, C)
C
      INCLUDE QTYPE
      DIMENSION A(3), B(3), C(3)
C
      DO 10 I = 1, 3
         J = 1 + MOD (I    , 3)
         K = 1 + MOD (I + 1, 3)
 10      C(I) = A(J) * B(K)  -  A(K) * B(J)
C
      RETURN
      END
```

Program: DDREAD

Purpose: Read input data deck via Fortran Name List

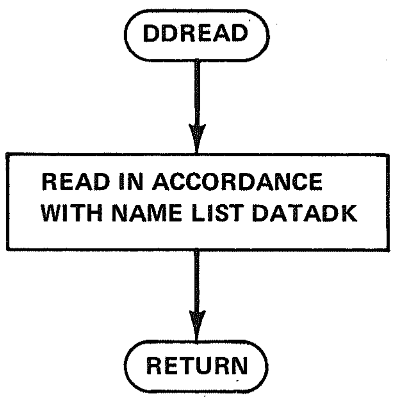
Called by: SETUP

Calls: None

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

Language: Fortran subroutine

Computations: None



VMASS*RFJK.DDREAD

```
C
C  PURPOSE
C
C     READ DATA DECK - SOME LOCATIONS BY NAME - OTHERS BY COMMON BLOCK
C
      SUBROUTINE DDREAD
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QEPH
      INCLUDE QPHC
      INCLUDE QFAC
      INCLUDE QREADE
      INCLUDE EVARI
      INCLUDE EVARIO
      INCLUDE EVARL
      INCLUDE EVARS
      INCLUDE EVARV
C
C
      NAME LIST / DATADK / EPH, FAC, IOU, ISW, LSW, PHC, VAC, VAR,
     1   DTAI,   DTAIDN, DTAIMN, DTAIUP, IDTRAJ, IEVREP, IEVRUN, IEVSTP,
     2   IEVSTR, IPLATM, IPLOTM, IPNOWM, LPRLAT, LPRNOW, LSTDDK,
     3   RTOPRF, TEPOJD, TEPOJS, ITEPOG, AU, RE, TPD, EMRAT, ICENTR,
     4   IOEPH , IOMES , IOPRL , IOROL , LOGOUT, LINIT ,
     5   RSCDDK, VSCDDK, LPRNOW, JK    , JKN   , RSCEIN, VSCEIN,
     6   IOPCH , IOPRT , IOSCR ,
C
      READ  (IOCDR, DATADK)
      IF (LOGOUT) WRITE (IOMES, 10)
   10 FORMAT (' DDREAD *** DATA DECK LOG OUT')
C
      RETURN
      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

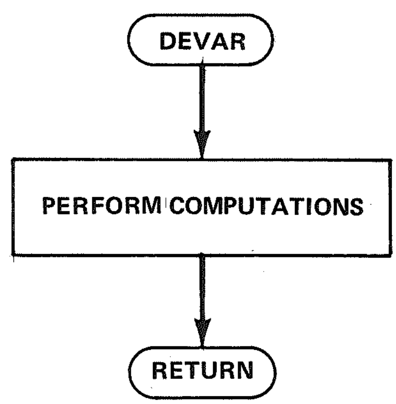
Calls: None

Errors and Diagnostics: None

Language: Fortran function

Computations:

$$x = V_m - V_n$$



VMASS SUBROUTINE DEVAR

VMASS*RFJK.DEVAR

```
C
C PURPOSE
C
C     OBTAIN DIFFERENCE BETWEEN TWO DOUBLE PRECISION VAR LOCATIONS
C     FOR WHICH SINGLE PRECISION INDICES HAVE BEEN COMPUTED
C
      FUNCTION DEVAR (M, N)
C
C     WRITE
C
C        X = DEVAR (M, N)
C
C     RATHER THAN
C
C        X = VAR(M) - VAR(N)
C
C     WHICH AT OBJECT CODE LEVEL BECOMES
C
C        X = VAR(2*M) - VAR(2*N)
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QTEM
      EQUIVALENCE (T1, IT1), (T3, IT2)
      DIMENSION IT1(2), IT2(2)
      IT1(1) = KVAR(M)
      IT1(2) = KVAR(M+1)
      IT2(1) = KVAR(N)
      IT2(2) = KVAR(N+1)
      DEVAR  = T1 - T3
      RETURN
      END
```

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

Purpose: Calculate vector dot product

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

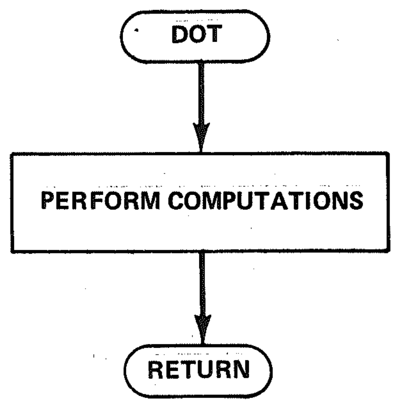
Calls: None

Errors and Diagnostics: None

Language: Fortran function

Computations:

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



VMASS SUBROUTINE DOT

```
VMASS*RFJK.DOT
C
C  PURPOSE
C
C       CALCULATE DOT PRODUCT OF TWO 3 COMPONENT VECTORS
C
        FUNCTION DOT (A, B)
C
        INCLUDE QTYPE
        DIMENSION A(3), B(3)
C
        DOT  =  A(1) * B(1)  +  A(2) * B(2)  +  A(3) * B(3)
C
        RETURN
        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

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

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

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

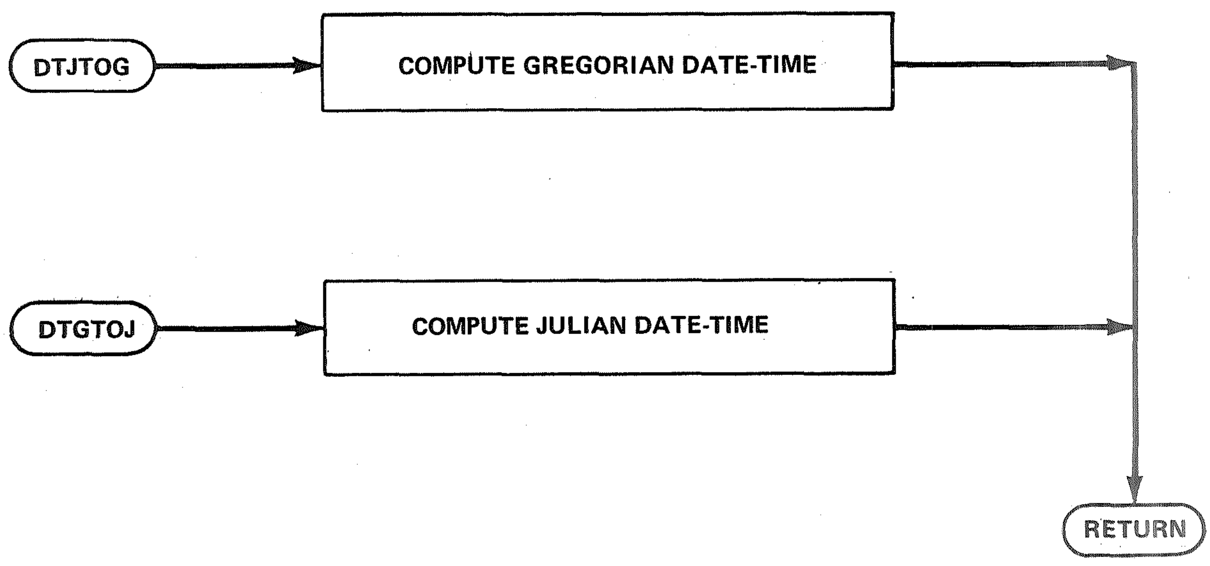
$$G_6 = 43200 + t - G_5 \cdot 60$$

On the reverse calculation for Gregorian to Julian

$$J_d = .5 + k - 32075$$
$$+ 1561 (i + 4800 + (j - 14) / 12) / 4$$
$$+ 367 (j - 2 - (j - 14) / 12) / 12$$
$$- 4 ((i + 4900 + (j - 14) / 12) / 100) / 4$$
$$J_t = 3600 G_4 + 60 G_5 + G_6$$

where

$$i = G_1$$
$$j = G_2$$
$$k = G_3$$



VMASS*RFJK.DTJTOG

```
      C     DTJTOG   R JESSUP - BELLCOMM - 21 JAN 69
      C
      C  PURPOSE
      C
      C     CONVERT DATE - TIME FROM JULIAN TO GREGORY (DTJTOG) OR
      C     VICE VERSA (DTGTOJ)
      C
      C  REFERENCE
      C
      C     FLIEGEL AND VAN FLANDERS COMMUNICATIONS ACM OCTOBER 68
      C
      C  CALLING SEQUENCE
      C
            SUBROUTINE DTJTOG (DJ, TJ, IG)
      C
      C  INPUTS
      C
      C     DJ         JULIAN DATE
      C     TJ         TIME IN SECONDS PAST JULIAN DATE
      C
      C  OUTPUT
      C
      C     IG         GREGORIAN DAY, MONTH, YEAR, HOUR, MINUTE, SECOND
      C                (24 HOUR CLOCK)
      C
            INCLUDE QTYPE
            INCLUDE QFAC
            INCLUDE QTEM
      C
            INTEGER IG(6)
      C
      C     NORMALIZE DJ + TJ TO READ D1.5 + T1 (T1 POSITIVE FRACTION OF DAY)
      C
            D1 = IFIX (FPT5 + DJ + TJ / F86400)
       10   T1 = TJ + F86400 * (DJ - D1)
      C
      C     COMPUTE GREGORY DATE
      C
       70   M = D1 + 68569
            N = 4 * M / 146097
            M = M - (146097 * N + 3) / 4
            I = 4000 * (M + 1) / 1461001
            M = M - 1461 * I / 4 + 31
            J = 80 * M / 2447
            K = M - 2447 * J / 80
            M = J / 11
            J = J + 2 - 12 * M
            I = 100 * (N - 49) + I + M
            IG(1) = I
            IG(2) = J
            IG(3) = K
            IG(4) = (F43200 + T1) / F3600
            T1 = T1 - IG(4) * F3600
            IG(5) = (F43200 + T1) / F60
            T1 = T1 - IG(5) * F60
            IG(6) = (F43200 + T1)
```

```
            RETURN
      C
      C
            ENTRY DTGTOJ (IG1, DJ1, TJ1)
      C
      C
      C     COMPUTED JULIAN DATE GIVEN AS DJ.5 + TJ
      C
            I = IG1(1)
            J = IG1(2)
            K = IG1(3)
            DJ1 = 0.5 + FLOAT (K - 32075
           1   + 1461 * (I + 4800 + (J - 14) / 12) / 4
           2   + 367 * (J - 2 - (J - 14)/ 12 * 12) / 12
           3   - 3 *((I + 4900 + (J - 14) / 12) / 100) / 4)
            TJ1 = IG1(4) * F3600 + IG1(5) * F60 + IG1(6)
            RETURN
            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.

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

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

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

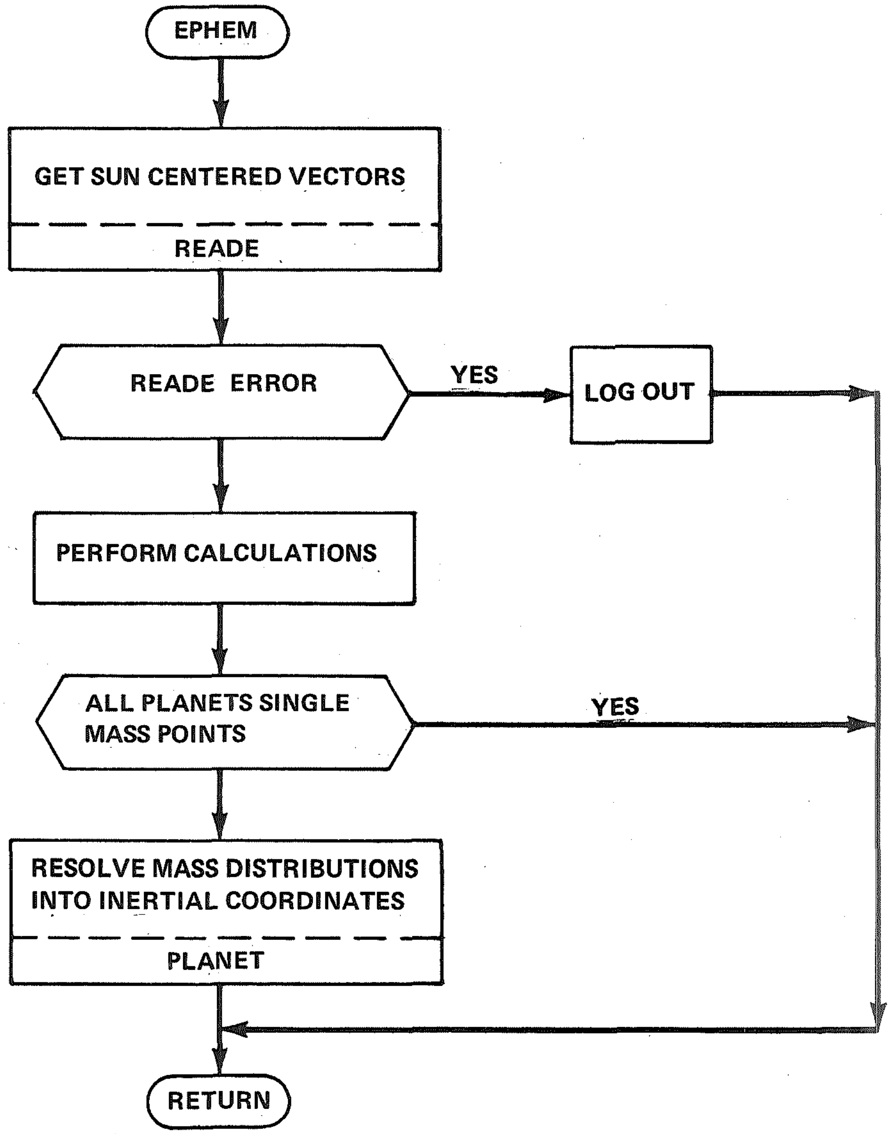
$$V_1 = \sum_j GOOR_k \; TABOUT_{i,j}$$

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

where

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

k = JKLOC (j)


EPHEM
GET SUN CENTERED VECTORS
READE
READE ERROR
YES
LOG OUT
PERFORM CALCULATIONS
ALL PLANETS SINGLE MASS POINTS
YES
RESOLVE MASS DISTRIBUTIONS INTO INERTIAL COORDINATES
PLANET
RETURN

VMASS*RFJK.EPHEM

```
      C       EPHEM     R JESSUP - BELLCOMM - 09 JAN 69
      C
      C  PURPOSE
      C
      C       DETERMINES FROM JPL EPHEMERIS FILES THE POSITIONS AND VELOCITIES
      C       OF THE GRAVITATING BODIES
      C
      C  INPUTS
      C
      C       TTRE      ELAPSED TRAJECTORY TIME FROM FIRST EVENT
      C       TEP0JD    JULIAN EPH DATE AT FIRST EVENT
      C       TEP0JS    JULIAN EPH TIME AT FIRST EVENT
      C       TEPEJD    CURRENT JULIAN EPHEMERIS DATE (DAYS)
      C       TEPEJS    CURRENT JULIAN EPHEMERIS TIME (SECONDS)
      C       TUTODA    MULTIPLIER FOR CONVERTING TRAJECTORY TIME TO DAYS
      C
      C  OUTPUTS
      C
      C       JK        LIST OF CURRENT GRAVITATING MASS POINTS
      C       JKN       NUMBER OF ENTRIES IN JK
      C       RJKEIN    CURRENT INERTIAL POS VECTOR OF MASS PT JK
      C       VJKEIN    CURRENT INERTIAL VEL VECTOR OF MASS PT JK
      C       GJKR      RATIO OF MASS OF GRAV PT JK TO MASS OF ALL THE GRAV PTS
      C
      C
              SUBROUTINE EPHEM
      C
              INCLUDE QTYPE
              INCLUDE QVAR
              INCLUDE QPHC
              INCLUDE QTEM
              INCLUDE QEPH
              INCLUDE QFAC
              INCLUDE QREADE
              INCLUDE EVARIO
              INCLUDE EVARS
              INCLUDE EVARI
      C
              DIMENSION              IEPER1(2), IEPER2(4,5)
              DATA IEPER1 / ' READE *** ' /
             1   IEPER2 / 'DATE TOO EARLY FOR EPH ' ,
             2            'DATE TOO LATE  FOR EPH ' ,
             3            'IMPROPER OUTPUT REQUEST' ,
             4            'INVALID CENTRAL BODY   ' ,
             5            'ICW OUT OF RANGE       ' /
      C
      C
      C          INITIALIZE
      C
      C
      C       READ SUN-CENTERED EPHEMERITIES VIA JPL READE
      C
         80   T1     = TTRE * TUTODA
              I      = T1
              T2     = I
              TEPEJD = TEP0JD + T2
```

```
           TEPEJS = TEPOJS + (T1 - T2) * F86400
           I      = TEPEJS / F86400
           T3     = I
           TEPEJS = TEPEJS - F86400 * T3
           TEPEJD = TEPEJD + T3
           CALL READE (TEPEJD, TEPEJS, I)
           IF (I .EQ. 0) GO TO 104
                   WRITE (IOPRT, 82) IEPER1, (IEPER2(J,I), J = 1, 4)
                   WRITE (IOMES, 82) IEPER1, (IEPER2(J,I), J = 1, 4)
    82             FORMAT (12A6)
               WRITE (IOMES, 102)
    102        FORMAT (' READE  CALLED BY EPHEM')
               LOGOUT = .TRUE.
               RETURN
C
C          CONVERT TO INERTIAL COORDINATES (CENTER AT SOLAR BARY-CENTER)
C
    104    DO 110 I = 1, 3
               TV1(I) = F0
    110        TV2(I) = F0
           DO 130 J = 1, 12
               IF (IREQ(J) .NE. 2) GO TO 130
                   K = JKLOC (J)
                   DO 120 I = 1, 3
                       TV1(I) = TV1(I) + GOOR(K) * TABOUT(I,J)
    120                TV2(I) = TV2(I) + GOOR(K) * TABOUT(I+3,J)
    130                CONTINUE
           DO 150 J = 1, 12
               IF (IREQ(J) .NE. 2) GO TO 150
                   K = JKLOC (J)
                   DO 140 I = 1, 3
                       ROOEIN(I+K-1) = TABOUT(I,J)   - TV1(I)
    140                VOOEIN(I+K-1) = TABOUT(I+3,J) - TV2(I)
    150        CONTINUE
C
C
C          RESOLVE PLANETS INTO INERTIALLY BASED COMPONENTS PER LIST JKPTS
C
           IF (JKPTSN .NE. 0) CALL PLANET
C
           RETURN
           END
```

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:

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

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

$$VO1EIN_x = -\ RO1EIN_y$$

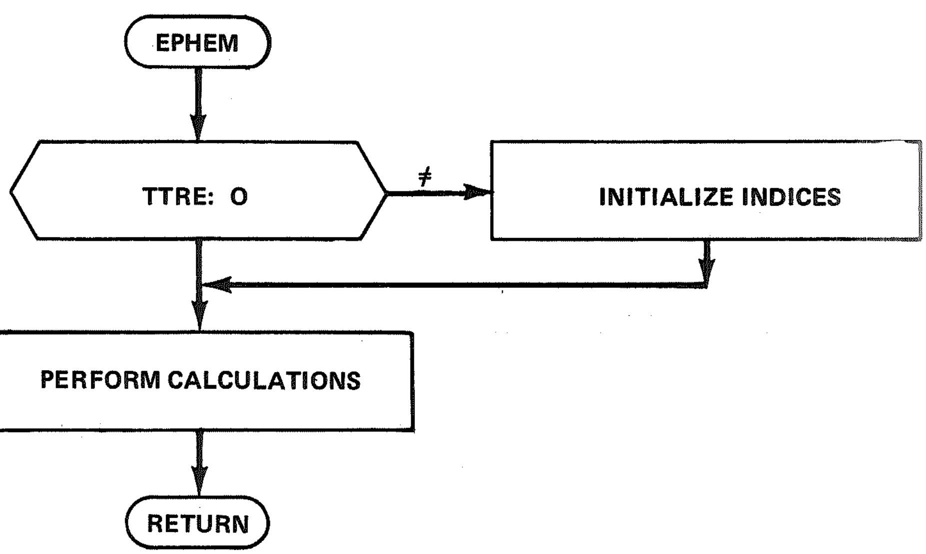
$$VO1EIN_y = RO1EIN_x$$

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

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

$$VO2EIN_x = -\ RO2EIN_y$$

$$VO2EIN_y = RO2EIN_x$$



VMASS*RFJK.EPH2CO

```
C      EPHEM    D NOVAK AND R JESSUP    BELLCOMM      01 APR 69
C
C   PURPOSE
C
C      DETERMINE POSITIONS AND VELOCITIES OF TWO GRAVITATING POINTS IN
C      CIRCULAR ORBIT ABOUT EACH OTHER.  NORMALIZED SYSTEM WITH TOTAL
C      MASS 1 AND ANGULAR ROTATION 1 IN INERTIAL COORDINATES.
C
C   INPUTS
C
C      TTRE     CURRENT VALUE TRAJECTORY TIME (NORMALIZED ELAPSED EPH T)
C      TEE      INITIAL VALUE NORMALIZED EPHEMERIS TIME
C      GOO      TABLE OF GRAVITATIONAL CONSTANTS
C
C   OUTPUTS
C
C      ROIEIN   POSITION ITH POINT INERTIAL
C      VOIEIN   VELOCITY ITH POINT INERTIAL
C
       SUBROUTINE EPHEM
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE QTEM
       INCLUDE QEPH
       INCLUDE QSTORE
       INCLUDE EVARI
       INCLUDE EVARS
       INCLUDE QFAC
C
       IF (TTRE .NE. F0) GO TO 10
          JK(1) = 1
          JK(2) = 2
          JKN   = 2
          K01   = JKLOC (JK(1))
          K02   = JKLOC (JK(2))
  10   T1 = TTRE - TEE
       T2 = COS (T1)
       T3 = SIN (T1)
       T4 =   T2 * GOO(K01)
       T5 =   T3 * GOO(K01)
       T6 = - T2 * GOO(K02)
       T7 = - T3 * GOO(K02)
C
       ROOEIN(K02  ) = T4
       ROOEIN(K02+1) = T5
       ROOEIN(K01  ) = T6
       ROOEIN(K01+1) = T7
       VOOEIN(K02  ) = - T5
       VOOEIN(K02+1) =   T4
       VOOEIN(K01  ) = - T7
       VOOEIN(K01+1) =   T6
       RETURN
       END
```

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

Purpose: Build and maintain table of trajectory events.

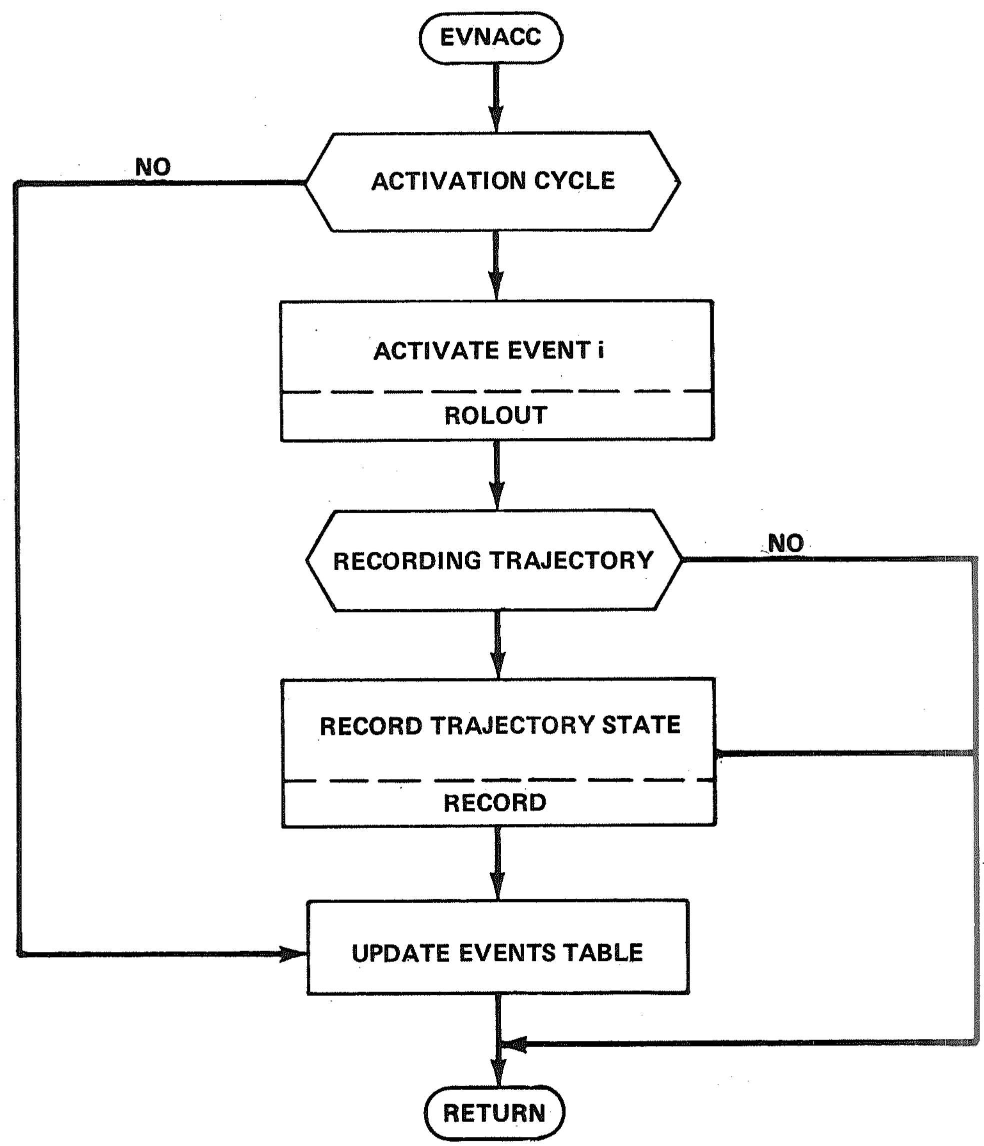
Called by: SIMXXX

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
C      EVNACC   R JESSUP - BELLCOMM - 06 JAN 69
C
C   PURPOSE
C
C      VMASS USER CODING IN THE SIMULATOR (SIMXXX) CONSISTS OF SUCCESSIVE
C      CALLS TO EVNACC, ONE CALL FOR EACH TRAJECTORY EVENT.  ON EVENT
C      ACTIVATION (IEVFLG I OR 0) EVNACC RETURNS CONTROL TO ONE TIME
C      EVENT CALCULATIONS FOLLOWING THE CALL.  ON TABLE UPDATES (IEVFLG 1
C      AS SET BY THE SIMULATOR AFTER EVENT ACTIVATION AND BEFORE MOVING
C      ON TOWARD NEXT EVENT) EVNACC RETURNS CONTROL TO NEXT CALL OR TO
C      THE SIMULATOR AFTER ALL CALLS HAVE BEEN MADE.
C
C
C   CALLING SEQUENCE
C
C      CALL EVNACC (I, IACT, IDACT, CRIT, AIMPT, $MISS)
C
C   INPUTS
C
C      I         ANY EVENT DESCRIBED IN THE SIMULATOR (EVENTS LIST)
C      IACT      ''ACTIVATING,' EVENT FOR I (IACT MUST ''OCCUR'' BEFORE I)
C      IDACT     ''DE-ACTIVATING'' EV FOR I (THE ''OCCURRENCE'' OF IDACT
C                PRE-EMPTS FURTHER CONSIDERATION OF I)
C      CRIT      EVENT CRITERION
C      AIMPT     AIM-POINT OF THE CRITERION
C      MISS      CALLING PRG STATEMENT NUM TO RECEIVE CONTROL ON EV MISS
C
C      TRCI      REFERENCE CONIC TIME INCREMENT FOR NORMAL COMPUTING STEP
C      IEVFLG    NAME OF EVENT EXPECTED TO BE DECLARED (AS SET BY 'EVNDT')
C                OR FLAG SET TO 0 BY THE SIMULATOR FOR TABLE UPDATES
C
C   OUTPUTS
C
C      TEVI      TIME REMAING FOR OCCURRENCE OF APPARENT NEAREST EVENT
C      IEVCUR    LATEST OR CURRENT DECLARED EVENT
C
C   EVENTS TABLE LAYOUT
C
C      EVN(1,I) CRITERION DEVIATION FROM AIMPT IN LAST COMPUTING INTERVAL
C      IEV(3,I) EVENT I
C      IEV(4,I) DE-ACTIVATING EVENT FOR I
C      IEV(5,I) VAR INDEX POINTER TO CRITERION FOR I
C      IEV(6,I) VAR INDEX POINTER TO AIM-POINT FOR I
C      IEV(7,I) STEPS TAKEN IN ATTEMPT TO CONVERGE TO THIS EVENT
C      IEV(8,I) FLAG FOR EVENT INITATION THROUGH PROGRAM CONTROL
C
       SUBROUTINE EVNACC (I, IACT, IDACT, CRIT, AIMPT, MISS)
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE QTEM
       INCLUDE QFAC
       INCLUDE QIEV
       INCLUDE EVARIO
       INCLUDE EVARS
       INCLUDE EVARI
```

```
            INCLUDE EVARL
      C
            IF (IEVFLG .EQ. 1) GO TO 30
      C
      C        EVENT ACTIVATION (IEVFLG I OR 0).  NOTE EVENT AND MAKE RECORD
      C        OF PRESENT PROGRAM STATE.
      C
               IF (IEVFLG .NE. I .AND. IEVFLG .NE. 0) RETURN 6
                LXECEV = .TRUE.
                IEVCUR = I
                CALL ROLOUT
                IF (.NOT. LRESKP) CALL RECORD
                RETURN
      C
      C     TABLE UPDATING (IEVFLG 1).  REMOVE INHIBITED (DEACTIVATED) ENTRIES
      C
       30   IF (IEVCUR .NE. IDACT) GO TO 50
                DO 40 K = 1, IEVN
                     IF (IEVCUR .NE. IEV(4,K)) GO TO 40
                          DO 32 M = 1, 8
       32                    IEV(M,K) = 0
                          DO 34 M = IEVM, 0, -1
                             IEVN = M
       34                    IF (IEV(3,M) .NE. 0) RETURN 6
       40            CONTINUE
                RETURN 6
       50   IF (I .NE. IACT .AND. IEVCUR .NE. IACT) RETURN 6
      C
      C     INSERT ENABLED EVENTS (WHICH CAN LOGICALLY FOLLOW I IN TIME FLOW)
      C
            DO 60 K = 1, IEVN
       60      IF (I .EQ. IEV(3,K)) RETURN 6
            DO 70 K = 1, IEVM
               IF (IEV(3,K) .NE. 0) GO TO 70
                  IEVN = MAX (IEVN, K)
                  IEV(8,K) = 0
                  IEV(7,K) = 0
                  IEV(6,K) = INDRS (AIMPT)
                  IEV(5,K) = INDRS (CRIT )
                  IEV(4,K) = IDACT
                  IEV(3,K) = I
                  EVN(1,K) = DEVAR (IEV(5,K), IEV(6,K))
                  RETURN 6
       70      CONTINUE
      C
            LOGOUT = .TRUE.
            WRITE (IOMES, 90)
       90   FORMAT (' EVNACC *** EVENTS TABLE OVERFLOW')
            RETURN 6
      C
            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

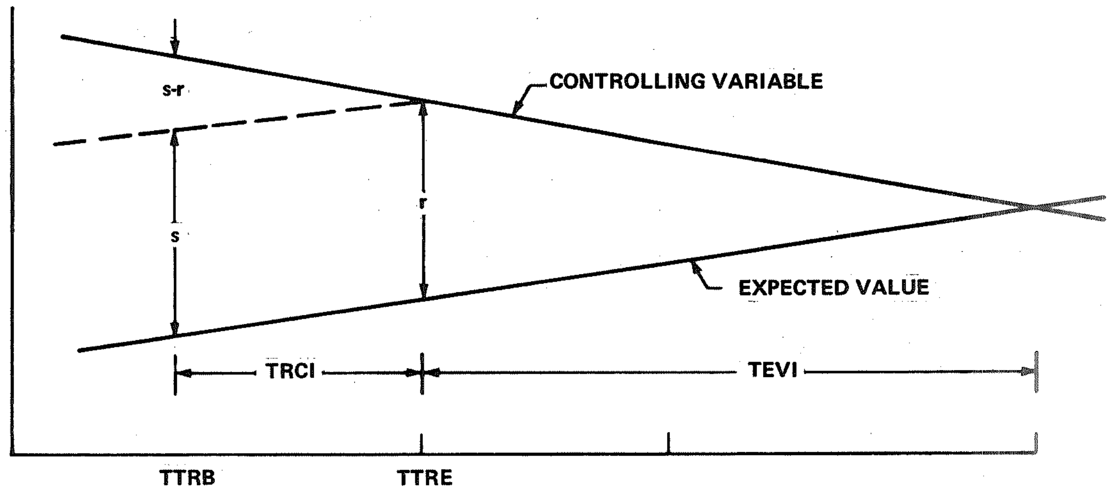
Computation: The predicted time, TEVI, to an event is given by the linear extrapolation

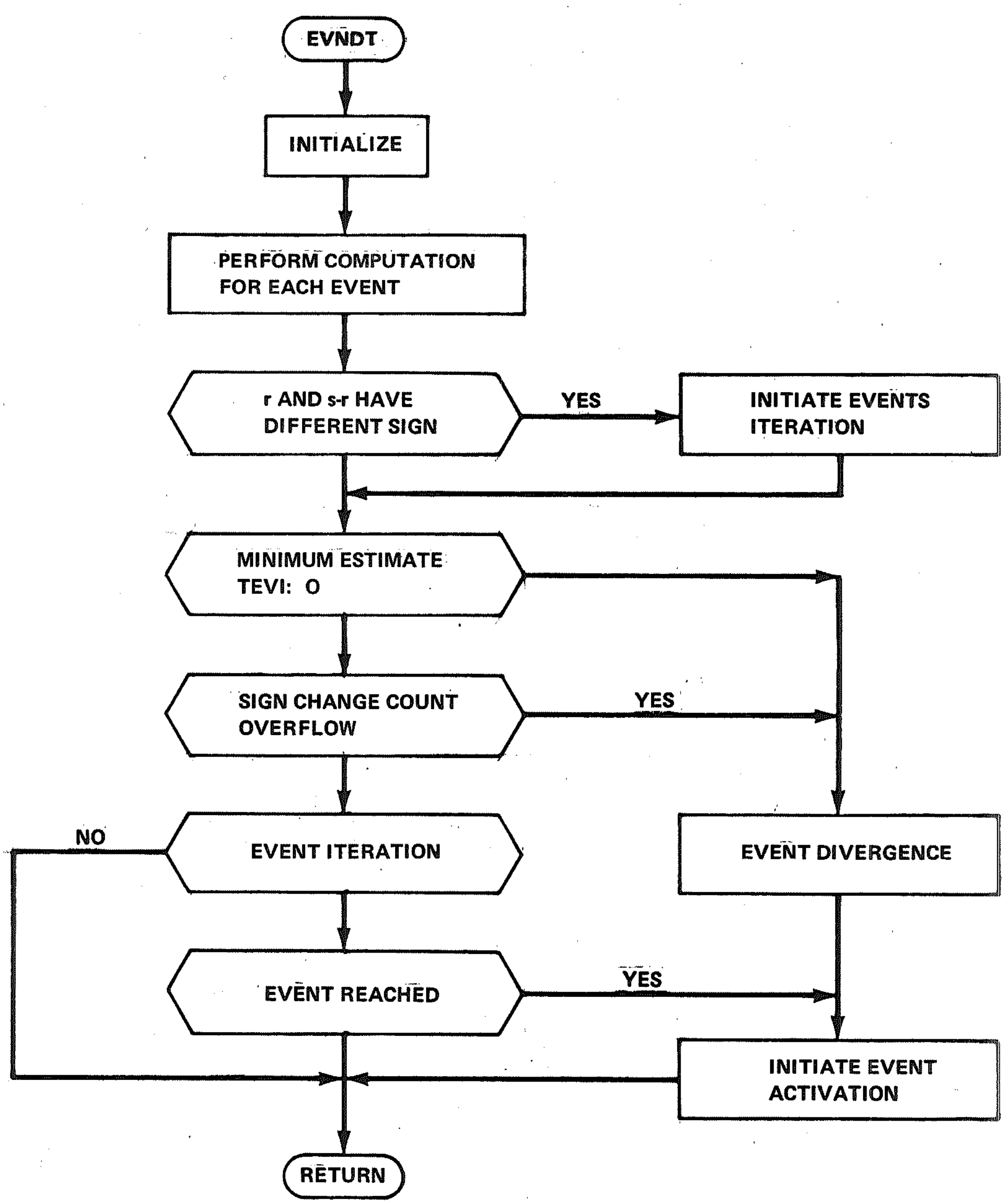
$$\frac{TEVI}{TRCI} = \frac{r}{s-r}$$

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.


s-r
CONTROLLING VARIABLE
r
s
EXPECTED VALUE
TRCI
TEVI
TTRB
TTRE


EVNDT
INITIALIZE
PERFORM COMPUTATION FOR EACH EVENT
r AND s-r HAVE DIFFERENT SIGN
YES
INITIATE EVENTS ITERATION
MINIMUM ESTIMATE TEVI: O
SIGN CHANGE COUNT OVERFLOW
YES
NO
EVENT ITERATION
EVENT DIVERGENCE
EVENT REACHED
YES
INITIATE EVENT ACTIVATION
RETURN

VMASS*RFJK.EVNDT

```
C
C  INPUTS
C
C      TRCI     REFERENCE CONIC TIME INCREMENT OVER COMPUTING INTERVAL
C      LINTDN   LENGTH OF COMP INTERVAL IS DOWN FROM EARLIER VALUE
C
C  OUTPUTS
C
C      TEVI     TIME TO GO TO NEAREST EVENT
C      IEVFLG   IF SOME EVENT WILL OCCUR IN CUR COMP INT THIS IS ITS NAME
C      LIMPEV   FLAG FOR INDICATING IMPENDING EVENT OCCURRENCE
C      LXECEV   LOGICAL EVENT EXECUTION FLAG - WE ARE THERE
C
C  PURPOSE
C
C      FIND SMALLEST TIME TO GO FOR ''NEAREST'' EVENT AS BASED ON EARLIER
C      AND PRESENT CRITERION - AIMPOINT DEVIATIONS
C
       SUBROUTINE EVNDT
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE QTEM
       INCLUDE QFAC
       INCLUDE QIEV
       INCLUDE EVARI
       INCLUDE EVARL
       INCLUDE EVARS
       INCLUDE EVARIO
C
       TEVI = FINITY
       IF (IEVN .EQ. 0) RETURN
       IF (.NOT. LINTDN) GO TO 205
          LIMPEV = .FALSE.
          LINTDN = .FALSE.
  205  DO 280 K = 1, IEVN
          IF (IEV(3,K) .EQ. 0) GO TO 280
          IF (IEV(8,K) .EQ. 0) GO TO 210
             TEVI = TRCI
             GO TO 230
  210     T1 = DEVAR (IEV(5,K), IEV(6,K))
          IF (SIGN (F1, EVN(1,K)) .NE. SIGN (F1, T1)) IEV(7,K)=IEV(7,K)+1
          T2 = EVN(1,K) - T1
          IF (T2 .NE. F0) TEVI = MIN (TEVI, TRCI * EVN(1,K) / T2)
          IF (TEVI .GT. F0) GO TO 240
  220        WRITE (IOMES, 222)
  222        FORMAT (// ' EVNDT  *** DIVERGENCE')
             DO 224 I = 1, IEVN
                T1 = DEVAR (IEV(5,K), IEV(6,I))
  224           WRITE (IOMES, 226) I,(IEV(J,I),J=3,8), EVN(1,I), T1
  226           FORMAT (' IEV(',I3,')',2A9,4I9,2G30.8)
  230        IEVFLG = IEV(3,K)
             LXECEV = .TRUE.
             LIMPEV = .FALSE.
             RETURN
  240     IF (IEV(7,K) .EQ. 0) GO TO 250
```

```
                 LIMPEV = .TRUE.
                 IEV(7,K) = IEV(7,K) + 1
                 IF (IEV(7,K) .GT. 12) GO TO 220
      250     IF (.NOT. LIMPEV) GO TO 280
                 T3 = ABS (TEVI - TRCI)
                 IF (T3 .LE. TTO) GO TO 230
                 IF (T3 .LE. FINTES * (TEVI + TRCI)) GO TO 230
      280     CONTINUE
            RETURN
            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.

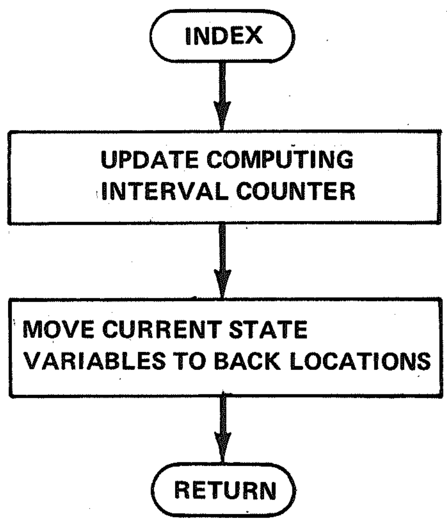
Called by: STEPIN

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS*RFJK.INDEX

```
C
C  PURPOSE
C
C     THIS PROGRAM INDEXES THE APPROPRIATE VALUES AT THE END OF A
C     COMPLETED COMPUTING INTERVAL TO COMPRISE THE BEGINNING VALUES FOR
C     THE NEXT INTERVAL.
C
C  INPUTS
C
C     RSCEIN
C     VSCEIN
C     RRCEVM    REFERENCE CONIC FINAL POSITION REL TO THE VIRT MASS
C     VRCEVM                 ''          VELOCITY        ''
C     RVMEIN    VIRTUAL MASS FINAL POSITION RELATIVE TO INERTIAL FRAME
C     VVMEIN                 ''        VELOCITY          ''
C     GVME      G TIMES FINAL VIRTUAL MASS MAGNITUDE
C     QVME                   ''          MAGNITUDE RATE
C     TEPE      FINAL EPHEMERIS TIME (AT END OF OLD INTERVAL)
C     RRCDVM    REFERENCE CONIC FINAL DISTANCE FROM VIRTUAL MASS
C     RSCDJK    DIST FROM SPACECRAFT TO MASS PT JK AT END OF COMP INT
C
C  OUTPUTS
C
C     RSCBIN
C     VSCBIN
C     RRCBVM    REFERENCE CONIC INITIAL POSITION REL TO THE VIRT MASS
C     VRCBVM                 ''            VELOCITY                ''
C     RVMBIN    VIRTUAL MASS INITIAL POSTION REL TO THE INERTIAL FRAME
C     VVMBIN                 ''            VELOCITY                ''
C     GVMB      G TIMES INITIAL VIRTUAL MASS MAGNITUDE
C     QVMB                   ''            MAGNITUDE RATE
C     TEPB      INITIAL EPHEMERIS TIME (AT START OF NEW INTERVAL)
C     RRCAVM    REFERENCE CONIC INITIAL DISTANCE FROM VIRTUAL MASS
C     RSCAJK    DIST FROM SPACECRAFT TO MASS PT JK AT BEG OF COMP INT
C
      SUBROUTINE INDEX
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QEPH
      INCLUDE QIEV
      INCLUDE EVARI
      INCLUDE EVARS
      INCLUDE EVARV
C
      INCNTR = INCNTR + 1
      DO 10 I = 1, JKN
         J = JKLOC (JK(I))
10       RSCAOO(J) = RSCDOO(J)
      DO 20 I = 1, 3
         RSCBIN(I) = RSCEIN(I)
         VSCBIN(I) = VSCEIN(I)
         RRCBVM(I) = RSCEVM(I)
         VRCBVM(I) = VSCEVM(I)
         RVMBIN(I) = RVMEIN(I)
20       VVMBIN(I) = VVMEIN(I)
```

```
      DO 30 I = 1, IEVN
   30    IF (IEV(3,I) .NE. 0) EVN(1,I) = DEVAR (IEV(5,I), IEV(6,I))
      RRCAVM = RSCDVM
      GVMB   = GVME
      QVMB   = QVME
      TEPB   = TEPE
      RETURN
      END
```

Program: INITAL

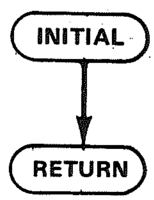
Purpose: Perform user coded calculations just after the data deck is read into the computer.

Called by: SETUP

Calls: None

Language: Dummy Fortran subroutine

Computations: None



VMASS SUBROUTINE INITAL

```
VMASS*RFJK.INITAL
      C
      C  PURPOSE
      C
      C     PROVIDE FOR USER CALCULATIONS ON DATA DECK JUST AFTER READ I
      C
            SUBROUTINE INITAL
      C
      C
      C
            RETURN
            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:

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

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

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

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

$$S_{5,k} = S_{4,k} \div (ERCBVM \circ V_{2,k} + |V_2|)$$

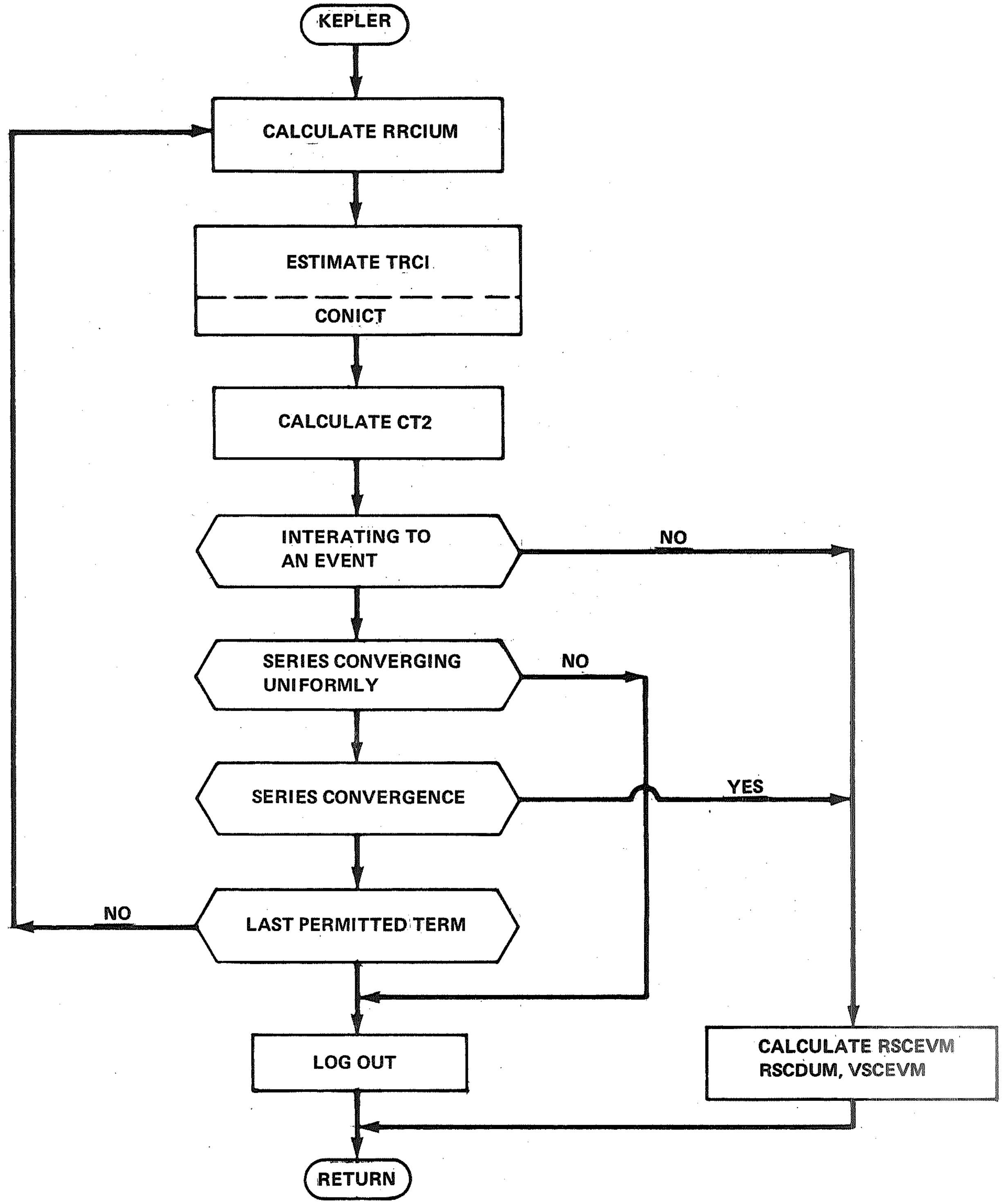
$$RRCIVM_k = V_1 + V_2 \cdot (S_{5,k} - 1)$$

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

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

$$RRCDVM = |RRCEVM|$$

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


KEPLER
CALCULATE RRCIUM
ESTIMATE TRCI
CONICT
CALCULATE CT2
INTERATING TO AN EVENT
NO
SERIES CONVERGING UNIFORMLY
NO
SERIES CONVERGENCE
YES
LAST PERMITTED TERM
NO
LOG OUT
CALCULATE RSCEVM RSCDUM, VSCEVM
RETURN

VMASS*RFJK.KEPLER

```
C
C  PURPOSE
C
C     GIVEN THE INITIAL POSITION AND VELOCITY VECTORS AND THE VECTOR
C     ORBITAL ELEMENTS RELATIVE TO THE VIRTUAL MASS, THIS PROGRAM COM-
C     PUTES THE FINAL RELATIVE POSITION AND VELOCITY VECTORS FOR AN ARC
C     OF THE REFERENCE CONIC SECTION.  DEPENDING UPON A FLAG SETTING THE
C     PROGRAM OPTIONALLY WILL ITERATE ACCURATELY TO A DESIRED FINAL TIME
C     OR, IN A SINGLE PASS, WILL OBTAIN FINAL CONDITIONS WHICH ONLY
C     APPROXIMATE THE SPECIFIED FINAL TIME.
C
C  INPUTS
C
C     LOOP      FLAG SET .TRUE. IF FINAL TIME TO BE ITERATED ACCURATELY
C               TO THE PRESCRIBED VALUE
C     RRCBVM    INITIAL POSITION ON REF CONIC RELATIVE TO VIRT MASS
C     VRCBVM    INITIAL VELOCITY                  ''
C     HRCBVM    ANGULAR MOMENTUM VECTOR OF REF CONIC REL TO THE VIRT MASS
C     ERCBVM    ECCENTRICITY                      ''
C     HRCAVM    MAGNITUDE OF THE VECTOR ANGULAR MOMENTUM
C     ERCAVM                     ''               ECCENTRICITY
C     GVMB      PRODUCT OF UNIV GRAVIT CONST AND VIRT MASS MAG
C     TEPI      DESIRED (EPHEMERIS) TIME INCREMENT
C     TTO       TOLERANCE ON ACCURACY OF COMPUTED TIME INCREMENT
C     CT2       COEFICIENT OF SQUARED TERM IN BIASED TIME EXPANSION.
C               THIS IS COMPUTED INTERNALLY IN THIS SUBROUTINE, BUT AN
C               INITIAL VALUE (OF 0) MUST BE PROVIDED FOR THE STARTING
C               COMPUTATION INTERVAL.  AFTER THIS THE LAST VALUE OF THE
C               PRECEEDING STEP IS USED AS A CURRENT FIRST GUESS.
C     KEPMAX    MAX ALLOWABLE NUM OF ITER IN THIS PROG FOR FINAL COND
C
C  OUTPUT
C
C     RRCEVM    REF CONIC FINAL POSITION REL TO VIRT MASS
C     VRCEVM    REF CONIC FINAL VELOCITY REL TO VIRT MASS
C     RRCDVM    REF CONIC FINAL DISTANCE FROM VIRT MASS
C     RRCIVM    REFERENCE CONIC POSITION INCREMENT RELATIVE TO V MASS
C     CT2
C
      SUBROUTINE KEPLER
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QFAC
      INCLUDE QTEM
      INCLUDE EVARI
      INCLUDE EVARIO
      INCLUDE EVARL
      INCLUDE EVARS
      INCLUDE EVARV
C
      S1 = FINITY
      DO 50 K = 1, KEPMAX
         S2 = (CT2 * TEPI + F1) * TEPI
         DO 10 I = 1, 3
            SV1(I) = VRCBVM(I) * S2
```

```
      10          SV2(I) = RRCBVM(I) + SV1(I)
               S3 = VALUE (SV2)
               S4 = (HRCAVM  / GVMB) * HRCAVM
               S5 = S4 / (DOT (ERCBVM, SV2) + S3)
               S6 = S5 - F1
     C
               DO 20 I = 1, 3
      20          RRCIVM(I) = SV1(I) + SV2(I) * S6
     C
               CALL CONICT
               IF (.NOT. LOGOUT) GO TO 30
                  WRITE (IOMES, 25)
      25          FORMAT (' CONICT CALLED BY KEPLER')
                  RETURN
      30       CT2 = (S2 / TRCI  -  F1) / TRCI
               IF (.NOT. LIMPEV) GO TO 70
                     S7 = ABS (TTRI - TRCI)
                  IF (S7 .LE. S1) GO TO 40
                     WRITE (IOMES, 110)
                  LOGOUT = .TRUE.
                  RETURN
      40          IF (S7 .LE. TTO) GO TO 70
                  IF (S7 .LE. FINTES * (TTRI + TRCI)) GO TO 70
      50             S1 = S7
            WRITE (IOMES, 120), KEPMAX
            RETURN
     C
      70    DO 75 I =  1, 3
      75       RRCEVM(I) = SV2(I) * S5
            RRCDVM = VALUE (RRCEVM)
            CALL UNITV (RRCEVM, SV4)
            DO 80 I = 1, 3
      80       SV6(I) = (ERCBVM(I) + SV4(I))/ S4
            CALL CROSS (HRCBVM, SV6, VRCEVM)
     C
            RETURN
     C
     110    FORMAT (' KEPLER *** CONVERGENCE NOT UNIFORM')
     120    FORMAT (' KEPLER *** TERMS IN SERIES EXCEEDING', I4)
     C
             END
```

Program: LODCOM

UPDCOM

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

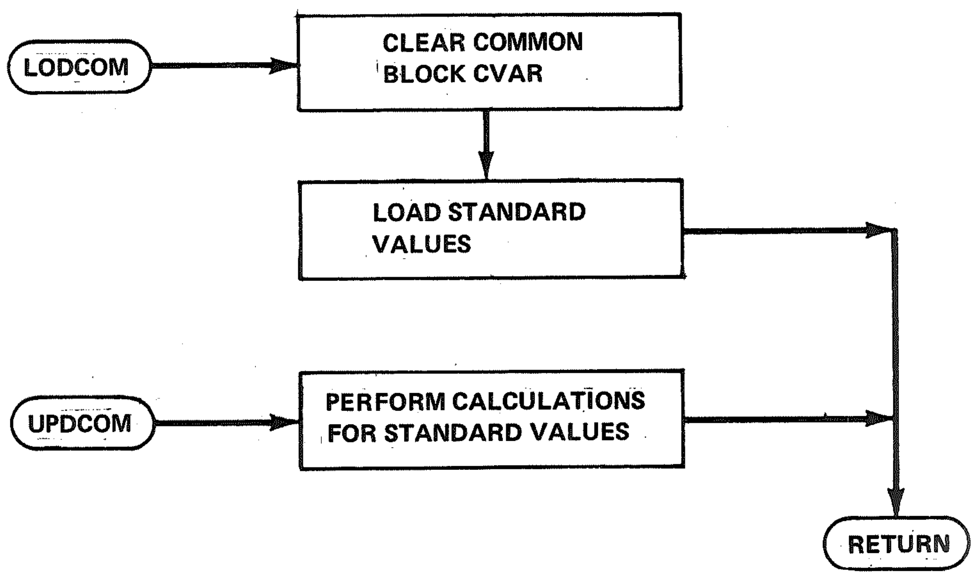
Called by: SIMXXX, SETUP

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: As shown in program



VMASS*RFJK.LODCOM

```
C
C  PURPOSE
C
C      IN THIS PROGRAM 'LODCOM' CLEARS ALL VARIABLE LOCATIONS TO ZERO AND
C      LOADS VALUES SHOWN FOR LOCATIONS NORMALLY REGARDED AS CONSTANT
C
       SUBROUTINE LODCOM
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE QEPH
       INCLUDE QFAC
       INCLUDE QPHC
       INCLUDE QTEM
       INCLUDE QREADE
C
       INCLUDE EVARI
       INCLUDE EVARS
       INCLUDE EVARIO
C
       DO 10 I = 1, 1160
 10       KVAR(I) = 0
       IEVM   = 20
       MVAR   = 1160
       KEPMAX = 6
       NMAX   = 30
C
       IOCDR = 5
       IOEPH = 12
       IOMES = 6
       IOPCH = 7
       IOPRL = 9
       IOPRT = 6
       IOROL = 8
       IOSCR = 11
       IPLOTM = 1
       IPLATM = 10000
       JKSPEC = 21
       IPNOWM = 500
C
C
       F0     = .0D0
       F001   = .001D0
       F00001 = .00001D0
       FPT5   = .5D0
       F1PT1  = 1.1D0
       FPT15  = .15D0
       FPT4   = .4D0
       F1     = 1.0
       F1PT5  = 1.5
       F2     = 2.0
       F2PT5  = 2.5
       F3     = 3.0
       F4     = 4.0
       F5     = 5.0
       F6     = 6.0
```

```
            F7     = 7.0
            F8     = 8.0
            F9     = 9.0
            F10    = 10.0
            F11    = 11.0
            F12    = 12.0
            F13    = 13.0
            F14    = 14.0
            F15    = 15.0
            F20    = 20.D0
            F24    = 24.0D0
            F30    = 30.D0
            F60    = 60.0D0
            F1440  = 1440.0D0
            F3600  = 3600.0D0
            F43200 = 43200.D0
            F86400 = 86400.0D0
            FINITY = 1.0E37
            FINTES = 1.D-18
            FTTO   = FPT5
            FRTO   = F5
      C
            RTOD = 57.29577951308232D0
            FTTOKM = .3048D-3
            RUTOAU = F1
            TUTODA = F1
      C
            RTOPRF = 1.D-15
            DTAIDN = FPT5
            DTAIUP = F1PT1
            DTAIMN = F1
      C
      C     FOR JPL READE AND FOR GRAVITATING BODIES REFERENCING
      C
            ICENT  = 10
            ICW    = 1
            TPD    = F1
            AU     = F1
            DO 210 I = 1, 11
               IREQ(I) = 2
        210    JK(I) = I
            JKN = 11
            KMERC  = JKLOC (JK(1))
            KVENUS = JKLOC (JK(2))
            KEARTH = JKLOC (JK(3))
            KMARS  = JKLOC (JK(4))
            KJUP   = JKLOC (JK(5))
            KSAT   = JKLOC (JK(6))
            KURAN  = JKLOC (JK(7))
            KNEP   = JKLOC (JK(8))
            KPLUTO = JKLOC (JK(9))
            KSUN   = JKLOC (JK(10))
            KMOON  = JKLOC (JK(11))
            KEMB   = JKLOC (JK(12))
      C
      C     FOLLOWING DATA FROM JPL COMPUTER PROGRAM DPTRAJ
            AUTOKM = 149597893.D0
```

```
          RETOKM = .63781492D4
          RE     = RETOKM / AUTOKM
          EMRAT  = 81.301D0
          T1     = F86400 ** 2 / AUTOKM ** 3
          GOO(KMERC ) = T1 * .221815976D5
          GOO(KVENUS) = T1 * .324860103D6
          GOO(KEARTH) = T1 * .398601200D6
          GOO(KMARS ) = T1 * .428284438D5
          GOO(KJUP  ) = T1 * .126707718D9
          GOO(KSAT  ) = T1 * .379265257D8
          GOO(KURAN ) = T1 * .578772346D7
          GOO(KNEP  ) = T1 * .689057627D7
          GOO(KPLUTO) = T1 * .732408934D5
          GOO(KSUN  ) = T1 * .132712499D12
          GOO(KMOON ) = T1 * .490278348D4
          GOO(KEMB  ) = GOO(KEARTH) + GOO(KMOON )
          RETURN
C
C
          ENTRY UPDCOM
C
C         THIS SECTION FOR COMPUTED VALUES OF COMMON.  THE ABOVE CONSTANTS CAN
C         BE REPLACED VIA DDREAD IF DESIRED.
C
          DTAI   = F1 / RTOPRF
          DTOR   = F1 / RTOD
          T1 = F0
          DO 402 I = 1, JKN
             IF (I .GT. 11) GO TO 404
                K = JKLOC (JK(I))
  402           T1 = T1 + GOO(K)
  404     DO 412 I = 1, JKN
             IF (I .GT. 11) GO TO 414
                K = JKLOC (JK(I))
  412           GOOR(K) = GOO(K) / T1
  414     CONTINUE
          RETURN
          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

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

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

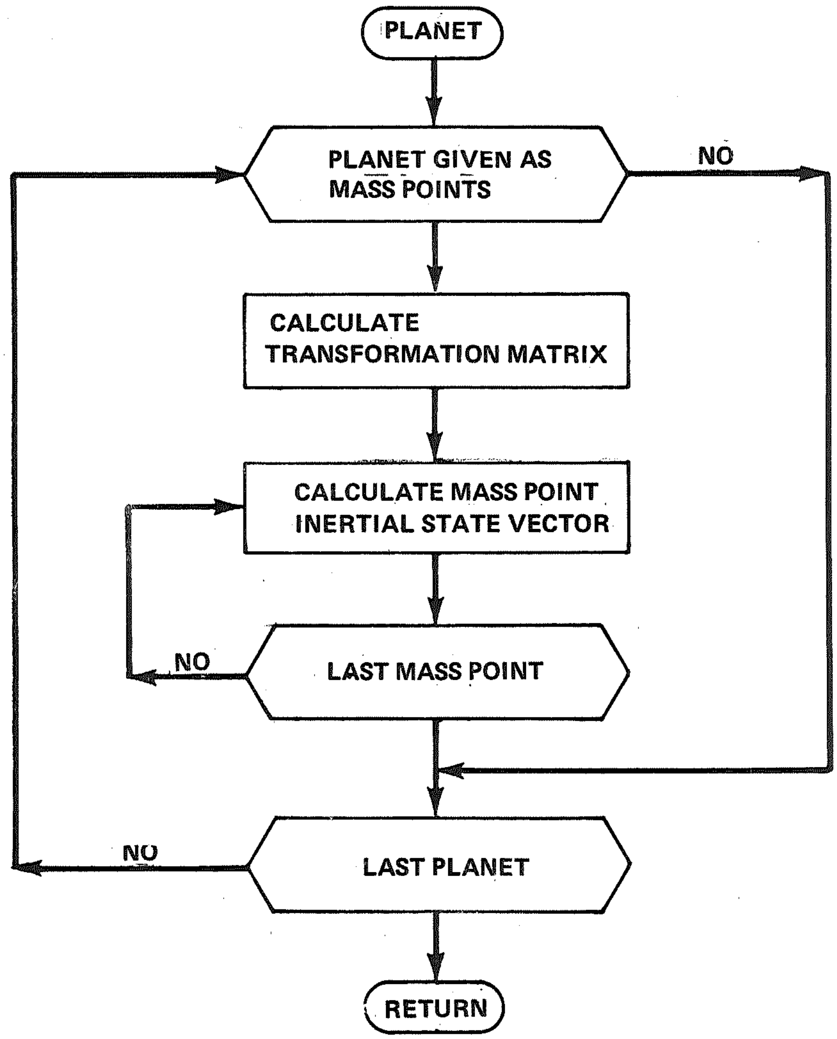
where $ROOEIN_q$ and $VOOEIN_q$ give the inertial state vector of the the planet itself.

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

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

$$i = 1, 2, 3$$

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


PLANET
PLANET GIVEN AS MASS POINTS
NO
CALCULATE TRANSFORMATION MATRIX
CALCULATE MASS POINT INERTIAL STATE VECTOR
NO
LAST MASS POINT
NO
LAST PLANET
RETURN

VMASS*RFJK.PLANET

```
C     PLANET   D NOVAK AND R JESSUP     BELLCOMM        17 MAR 69
C
C  PURPOSE
C
C     TO TRANSFORM TO THE REFERENCE INERTIAL COORDINATE SYSTEM THE
C     PLANETOGRAPHIC STATES (POS AND VEL) OF COLLECTIONS OF MASS POINTS
C     REPRESENTING ASPHERICAL POTENTIALS OF DESIGNATED PLANETS
C
C  INPUTS
C
C     TEP0JD   REFERENCE EPOCH OF EPHEMERIS TIME (JULIAN DAY COMPONENT)
C     TEP0JS   REFERENCE EPOCH OF EPHEMERIS TIME (JULIAN SEC COMPONENT)
C     TEPEJD   CURRENT DAY
C     TEPEJS   CURRENT DAY FRACTION IN SEC
C
C     JKPTSN   COUNT OF ALL POINT MASSES IN USE
C     JKPTS    PLANETARY MASS POINT LOCATORS
C
C     JKPTS(1,I) FOR PLANET I SUBSCRIPT IN RMN0JK OF FIRST POINT MASS
C     JKPTS(2,I) FOR PLANET I COUNT OF POINT MASSES
C
C    FOR EA PLANET JK (JK = 01,...,11) AT SETUP OR RECONFIGURATION TIME
C
C     RMN0JK   PLANETOGRAPHIC POS VECTOR OF MASS PT MN OF PLANET JK
C
C     DANEJK   RIGHT ASCENSION OF ASCENDING NODE OF EQUATORIAL PLANE
C              MEASURED FROM X REF DIRECTION IN BASE INERTIAL SYS XY PL
C     DOBEJK   OBLIQUITY OF EQ PL TO BASE INERTIAL XY PLANE
C     DLN0JK   LONGITUDE OF NODE AT REFERENCE TIME - (TEP0JD, TEP0JS)
C     SPLEJK   ANGULAR ROTATION RATE (SIDEREAL)
C
C    FOR EA PLANET AT CURRENT TIME
C
C     RJKEIN   INERTIAL POS VECTOR OF CENTER OF MASS (JK = 01, ..., 11)
C     VJKEIN   INERTIAL VEL VECTOR OF CENTER OF MASS (JK = 01, ..., 11)
C
C  OUTPUTS
C
C     RJKEIN   INERTIAL POS VECTOR OF EA MASS PT (JK = 11 + JKPTS + ..)
C     VJKEIN   INERTIAL VEL VECTOR OF EA MASS PT (JK = 11 + JKPTS + ..)
C
C
C  EPHEMERITIES STORAGE LAYOUT (SEE ALSO JPL READE)
C
C     R01EIN   CURRENT INERTIAL POS 01 MECURY    05 JUP       09 PLUTO
C     R02EIN                        02 VENUS     06 SAT       10 SUN
C      ...                          03 EARTH     07 URANUS    11 MOON
C     R12EIN                        04 MARS      08 NEP       12 EM B CEN
C
C     RJKEIN   RMN0JK AS LOCATED VIA LIST JKPTS (JK = 11 + ...)
C      ...
C
C     R..EIN   CURRENT INERTIAL POS PLANETARY MASS PTS
C
      SUBROUTINE PLANET
C
```

```
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QEPH
      INCLUDE QTEM
      INCLUDE QFAC
      INCLUDE QPHC
      INCLUDE QREADE
      INCLUDE EVARI
      INCLUDE EVARS
C
      DIMENSION X1(3,3)
C
      K1 = JKSPEC * (12 + JKPTSN)
      T1 = TEPEJD - TEPOJD + (TEPEJS - TEPOJS) * F86400
      DO 40 J = 1, 11
         IF (JKPTS(1,J) .EQ. 0) GO TO 40
            T2 = DLNEOO(J) + T1 * SPLEOO(J)
            T3 = COS (DANEOO(J))
            T4 = SIN (DANEOO(J))
            T5 = COS (DOBEOO(J))
            T6 = SIN (DOBEOO(J))
            T7 = COS (T2)
            T8 = SIN (T2)
            X1(1,1) =  T3 * T7  -  T4 * T5 * T8
            X1(1,2) = -T3 * T8  -  T4 * T5 * T7
            X1(1,3) =  T4 * T6
            X1(2,1) =  T4 * T7  +  T3 * T5 * T8
            X1(2,2) = -T4 * T8  +  T3 * T5 * T7
            X1(2,3) = -T3 * T6
            X1(3,1) =  T6 * T8
            X1(3,2) =  T6 * T7
            X1(3,3) =  T5
            N = JKPTS(2,J)
            K2 = JKSPEC * 12
            DO 30 M = 1, N
               DO 20 I1 = 1, 3
                  TV1(I1) = F0
                  DO 10 I2 = 1, 3
   10                TV1(I1) = TV1(I1) + X1(I1,I2) * ROOEIN(K2+I2-1)
                  TV2(I1) = SPLEOO(K2)
     1               * (X1(I1,2) * ROOEIN(K2-1) - X1(I1,1) * ROOEIN(K2))
                  ROOEIN(K1+I1-1) = ROOEIN(K2+I1-1) + TV1(I1)
   20             VOOEIN(K1+I1-1) = VOOEIN(K2+I1-1) + TV2(I1)
               K1 = K1 + JKSPEC
   30          K2 = K2 + JKSPEC
   40    CONTINUE
      RETURN
      END
```

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:

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

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

$$T_5 = QVMB \cdot RRCAVM^{-3}$$

$$T_6 = TRCI^2 \cdot T_5 \div 12$$

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

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

$$T_9 = TRCI^3 \cdot T_5 \div 20$$

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

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

$$SV_1 = RVMBIN + TRCI \cdot VVMBIN + T_9 \cdot RRCBVM + T_{10} \cdot RRCEVM - T_{11} \cdot VRCEVM$$

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

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

$$RSCEVM_0 = RRCEVM$$

$$VSCEVM_o = VRCEVM$$

$$GVME_o = GVMB + TRCI \cdot GVMB$$

$$QVME_o = QVMB$$

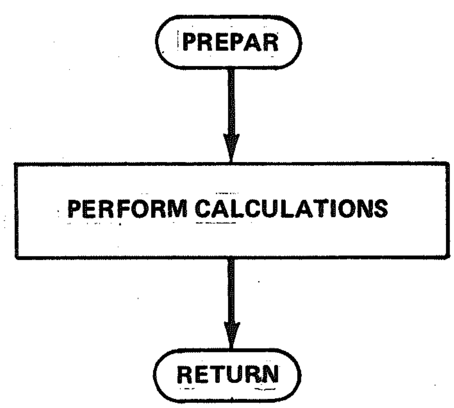
$$S_5 = TRCI^2 \div 20$$

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

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

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

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



VMASS*RFJK.PREPAR

```
C     PREPAR   D NOVAK AND R JESSUP     BELLCOMM  04 FEB 69
C
C  PURPOSE
C
C     TO MAKE FIRST ESTIMATE OF FINAL STATE OF VIRTUAL MASS AND TO
C     COMPUTE FIXED QUANTITIES FOR THE ITERATIVE SOLUTION OF THE
C     PERTURBATION EQUATION
C
C     PREPARE FOR ITERATION OF INTERVAL COMPUTATIONS. ESTIMATE END OF
C     INTERVAL VIRTUAL MASS STATE. SOLVE VIRTUAL MASS PERTUBATION
C     EQUATIONS.
C
C  INPUT
C
C     RRCBVM   INITIAL REFERENCE CONIC POSITION RELATIVE TO THE V MASS
C     VRCBVM   INITIAL REFERENCE CONIC VELOCITY RELATIVE TO THE V MASS
C     RRCEVM   FINAL   REFERENCE CONIC POSITION RELATIVE TO THE V MASS
C     VRCEVM   FINAL   REFERENCE CONIC VELOCITY RELATIVE TO THE V MASS
C     RVMBIN   INITIAL VIRTUAL MASS INERTIAL POSITION
C     VVMBIN   INITIAL VIRTUAL MASS INERTIAL VELOCITY
C     GVMB     INITIAL VIRTUAL MASS MAGNITUDE
C     QVMB     INITIAL VIRTUAL MASS MAGNITUDE RATE
C     RRCAVM   INITIAL DISTANCE BETWEEN REF CONIC AND V MASS
C     RRCDVM   FINAL   DISTANCE BETWEEN REF CONIC AND V MASS
C     TRCI     REF CONIC TIME INCREMENT FOR COMPUTING INTERVAL
C
C  OUTPUTS
C
C     ITRAT    ITERATION COUNTER (SET TO 0)
C     RVMEIN   FIRST ESTIMATE OF FINAL V MASS INERTIAL POSITION
C     VVMEIN   FIRST ESTIMATE OF FINAL V MASS INERTIAL VELOCITY
C     GVME     FINAL VIRTUAL MASS MAGNITUDE
C     QVME     FINAL VIRTUAL MASS MAGNITUDE RATE
C
      SUBROUTINE PREPAR
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QFAC
      INCLUDE QTEM
      INCLUDE EVARS
      INCLUDE EVARV
C
C     FIXED QUANTITIES FOR STEPIN HAVE S PREFIX
C
      T1 = TRCI * DOT (VRCEVM, RRCEVM) / RRCDVM ** 2
      T2 = TRCI ** 2
      T3 = TRCI * T2
      T4 = GVMB / RRCDVM ** 3
      T5 = QVMB / RRCAVM ** 3
      T6 = T2 * T5 / F12
      T7 = TRCI * (F2 + T1) * T4 / F4
      T8 = F1 - T2 * T4 / F12
      T9 = T3 * T5 / F20
      T10 = F1 + T2 * T4 * (FPT15 + T1 / F10)
      T11 = T3 * T4 / F30
```

```
          DO 20 I = 1, 3
             SV1(I) = RVMBIN(I) + TRCI * VVMBIN(I) - T9 * RRCBVM(I)
     1                + T10 * RRCEVM(I) - T11 * VRCEVM(I)
             SV2(I) = VVMBIN(I) - T6 * RRCBVM(I) + T7 * RRCEVM(I)
     1                + T8 * VRCEVM(I)
             RSCEVM(I) = RRCEVM(I)
             VSCEVM(I) = VRCEVM(I)
   20        SV3(I) = FINITY
          GVME = GVMB  +  TRCI * QVMB
          QVME = QVMB
          S5 = T2 / F20
          S4 = T2 / F4
          S3 = T2 / F12
          S2 = TRCI * FPT4
          S1 = TRCI / F2
C
          RETURN
          END
```

BELLCOMM, INC.

Program: PRTCOM

Purpose: Tabulate program COMMON by block name and subscript.

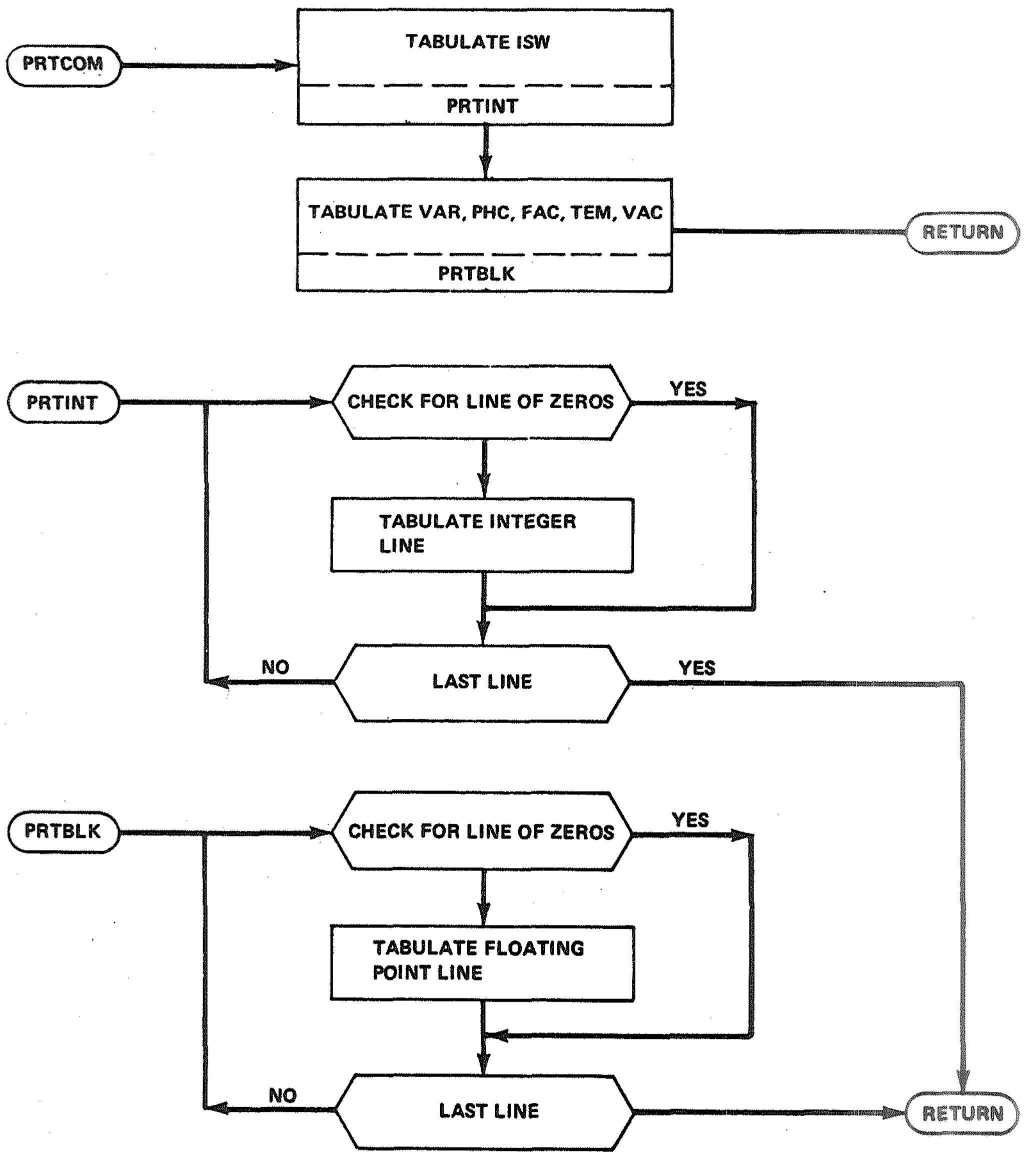
Called by: SETUP, SIMXXX

Calls: PRTBLK, PRTINT (internal subroutines)

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS*RFJK.PRTCOM

```
      C
      C  PURPOSE
      C
      C     PROVIDE COMPACT FEATURE FOR PRINTING COMMON BLOCKS
      C
            SUBROUTINE PRTCOM
      C
      C  INPUT
      C
      C     COMMON BLOCKS (WITHOUT EQUIVALENCES)
      C
      C  OUTPUT
      C
      C     PRINT OF EACH BLOCK AS SUBSCRIPTED ARRAY
      C
            INCLUDE QTYPE
            INCLUDE QVAR
            INCLUDE QTEM
            INCLUDE QPHC
            INCLUDE QFAC
            INCLUDE EVARI
            INCLUDE EVARIO
      C
            WRITE (IOPRT, 100)
            CALL PRTINT ('ISW', ISW, 300)
            CALL PRTBLK ('VAR', VAR, 300)
            CALL PRTBLK ('PHC', PHC, 50)
            CALL PRTBLK ('FAC', FAC, 58)
            CALL PRTBLK ('TEM', TEM, 100)
            CALL PRTBLK ('VAC', VAC, 50)
            RETURN
      C
       100  FORMAT (///' PRTCOM *** PRINT OF COMMON')
      C
            SUBROUTINE PRTBLK (NAME, BLOCK, NUMBER)
      C
            DOUBLE PRECISION BLOCK(NUMBER)
      C
            DO 30 I = 1, NUMBER, 4
                DO 10 K = 0, 3
       10         IF (BLOCK(I+K) .NE. F0) GO TO 20
                     GO TO 30
       20       WRITE (IOPRT, 110) NAME, I, (BLOCK(I+K), K = 0, 3)
       30      CONTINUE
            WRITE (IOPRT, 130)
            RETURN
      C
      C
            ENTRY PRTINT (NAM, IBLOC, NUM)
      C
            INTEGER IBLOC(NUM)
      C
            DO 80 I = 1, NUM, 8
               DO 60 K = 0, 7
       60         IF (IBLOC(I+K) .NE. 0) GO TO 70
                     GO TO 80
```

```
   70      WRITE (IOPRT, 120) NAM, I, (IBLOC(I+K), K = 0, 7),
     1        (IBLOC(I+K), K = 0, 7), (IBLOC(I+K), K = 0, 7)
   80      CONTINUE
        WRITE (IOPRT, 130)
        RETURN
C
C
  110   FORMAT (A7, '(', I3, ')', 4G30.18)
  120   FORMAT (A7, '(', I3, ')', 8O15/ 12X, 8A15/ 12X, 8I15)
  130   FORMAT (///)
        END
```

Program: PRTIME

Purpose: Print out various date-time groups

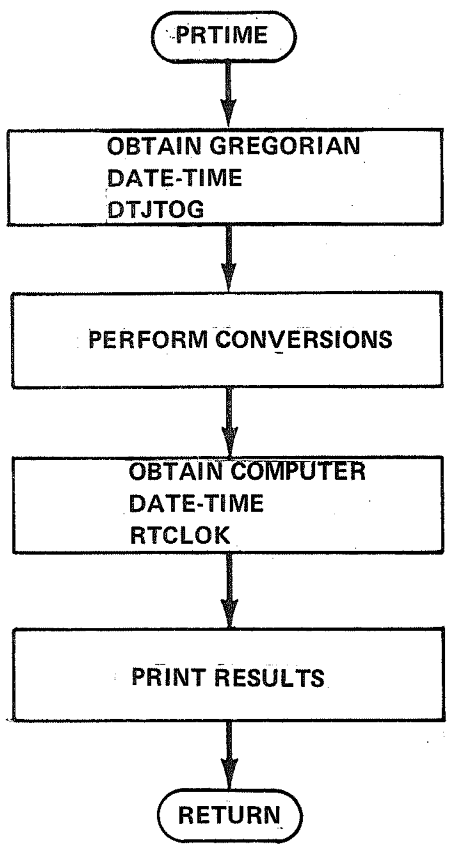
Called by: PRTXXX

Calls: DTJTOG, RTCLOK

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: Convert current trajectory time to calendar units.



VMASS*RFJK.PRTIME

```
      C     PRTIME    R JESSUP - BELLCOMM - 10 JAN 69
      C
      C  PURPOSE
      C
      C     PRINT OUT DATE TIME GROUP
      C
      C  INPUT
      C
      C     TTRE      CURRENT TRAJECTORY TIME
      C     TEPOJD    EPH DATE CURRENT AT FIRST EVENT (JULIAN DATE)
      C     TEPOJS    EPH TIME CURRENT AT FIRST EVENT
      C     TEPEJD    CURRENT EPH DATE
      C     TEPEJS    CURRENT EPH TIME
      C     IEVCUR    CURRENT EVENT
      C
      C  OUTPUT
      C
      C     IOPRT     DATE TIME STATEMENT (TRAJECTORY - EPHEMERIS - SIMULATOR)
      C
            SUBROUTINE PRTIME
      C
            INCLUDE QTYPE
            INCLUDE QVAR
            INCLUDE QFAC
            INCLUDE QTEM
            INCLUDE EVARI
            INCLUDE EVARS
            INCLUDE EVARIO
      C
            INCLUDE QMON
            INTEGER IGREG(6),  ITRAJ(4), ISTIME(3)
            EQUIVALENCE (     ITRAJ, ITRDA), (ITRAJ(2), ITRHR),
           1   (ITRAJ(3), ITRMN), (ITRAJ(4), ITRSC)
      C
            T1     = TTRE * TUTODA
            ITRDA = T1
            T2     = ITRDA
            ITRHR = (T1 - T2) * F24
            T3     = ITRHR
            ITRMN = (T1 - T2 - T3 / F24) * F1440
            ITRSC = (T1 - T2 - T3 / F24 - ITRMN / F1440) * F86400
            CALL DTJTOG (TEPEJD, TEPEJS, IGREG)
            CALL RTCLOK (ISTIME)
      C
            J = IGREG(2)
            IGREG(2) = MONTH(J)
            WRITE (IOPRT, 100) IEVCUR, ITRAJ, TEPEJD, TEPEJS, IGREG, ISTIME
        100 FORMAT (/ A7, ' AT', I5, ' DAYS', I3, ' HR', I3, ' MIN',
           1   I3, ' SEC; JULIAN', F10.1, ' DAYS', G12.6, ' SEC (',
           2   I4, 1X, A3, 2I3, 2(':',I2), '); COMPUTED ', 3A6//)
            RETURN
            END
```

Program: PRTRAJ

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

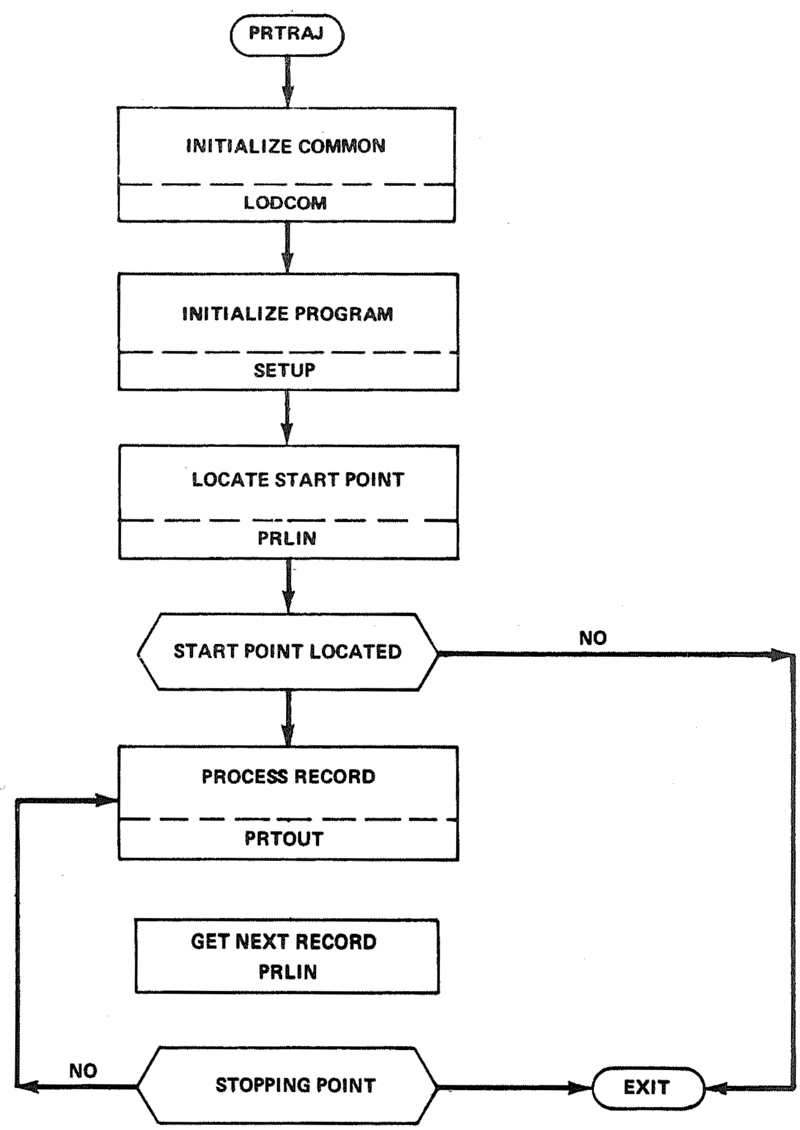
Called by: Separate computer execution

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

```
      C               PRTRAJ        R JESSUP       BELLCOMM      20 JUN 69
      C
      C  PURPOSE
      C
      C     PROCESS A TRAJECTORY STORED ON A SECONDARY FILE FOR EXTRA OUTPUT
      C     IN THE EVENT INTERVAL (IEVSTR, IEVSTP)
      C
      C  INPUT
      C
      C     IDTRAJ   IDENTIFICATION OF TRAJECTORY TO BE PROCESSED
      C     IEVSTR   EXTRA OUTPUT STARTING EVENT
      C     IOPRL    THE PROCESS LATER FILE
      C     IEVSTP   EXTRA OUTPUT STOPPING EVENT
      C
      C  OUTPUT
      C
      C     AS PROGRAMMED IN THE PRINT SUBROUTINE PRTXXX
      C
            INCLUDE QTYPE
            INCLUDE QVAR
            INCLUDE EVARL
            INCLUDE EVARIO
            INCLUDE EVARI
      C
        2   LLDCOM = .TRUE.
      C
            CALL LODCOM
            REWIND IOPRL
        5   CALL SETUP
      C
      C     SEARCH FOR TRAJECTORY AND STARTING EVENT (START AT BEGINNING IF
      C     NO STARTING EVENT IS GIVEN)
      C
            I1 = IEVSTR
            I2 = IEVSTP
       10   CALL PRLIN
            IF (IEVCUR .EQ. -1) GO TO 50
            IF (IDTRAJ(1) .EQ. 0) GO TO 30
               DO 12 I = 1, 3
       12         IF (IDTRAJ(I) .NE. IRTIME(I)) GO TO 14
                     GO TO 22
       14      WRITE (IOMES, 100)
               CALL PRTVAR
               GO TO 10
       22      IF (I1 .EQ. 0      ) GO TO 30
               IF (I1 .EQ. IEVCUR) GO TO 30
                  GO TO 5
      C
      C     PROCESS INFORMATION THROUGH SUBROUTINE PRTXXX
      C
       30   CALL PRTOUT
            CALL PRLIN
            IF (IEVCUR .EQ. -1) GO TO 50
      C
      C     TERMINATE AT STOPPING EVENT (OR AT END OF TRAJECTORY IF NO
      C     STOPPING EVENT IS GIVEN)
```

```
      C
            IF (IEVFLG .EQ. I2 .AND. I2 .NE. IEVCUR) GO TO 5
               GO TO 30
       50   PRINT 130
            GO TO 10
      100   FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED TRAJECTORY ')
      110   FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED START EVENT')
      120   FORMAT (' PRTRAJ *** CANNOT FIND SPECIFIED STOP EVENT ')
      130   FORMAT (' PRTRAJ *** END-OF-FILE')
            END
```

Program: PRTXXX (element name)

PRTOUT

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

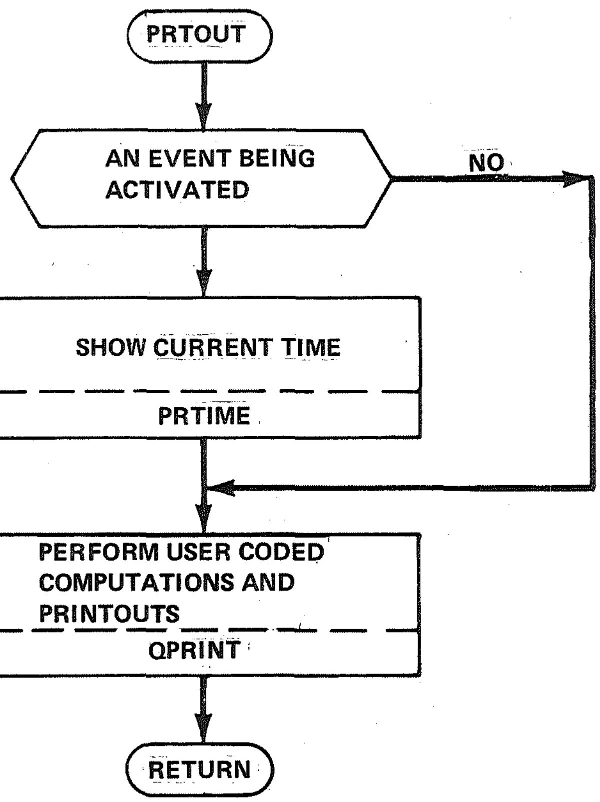
Called by: RECORD, PRTRAJ

Calls: PRTIME

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: As coded by user in element QPRINT.



VMASS*RFJK.PRTXXX

```
C      R JESSUP - BELLCOMM - 3 JAN 69
C
C  PURPOSE
C
C      PRESENT THE COMPUTED TRAJECTORY VIA THE SYSTEM PRINTER
C
C  INPUTS
C
C      ALL FORMALLY NAMED PROGRAM VARIABLES
C
C  OUTPUTS
C
C                 AS PROGRAMMED
C
       SUBROUTINE PRTOUT
C
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE EVARI
       INCLUDE EVARL
       INCLUDE EVARV
       INCLUDE EVARS
       INCLUDE EVARIO
       INCLUDE QSTORE
C
       IF (LXECEV) CALL PRTIME
C
       INCLUDE QPRINT, LIST
C
       RETURN
       END
```

Program: PRTVAR

PRTEPH

PRTFAC

PRTPHC

PRTTEM

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

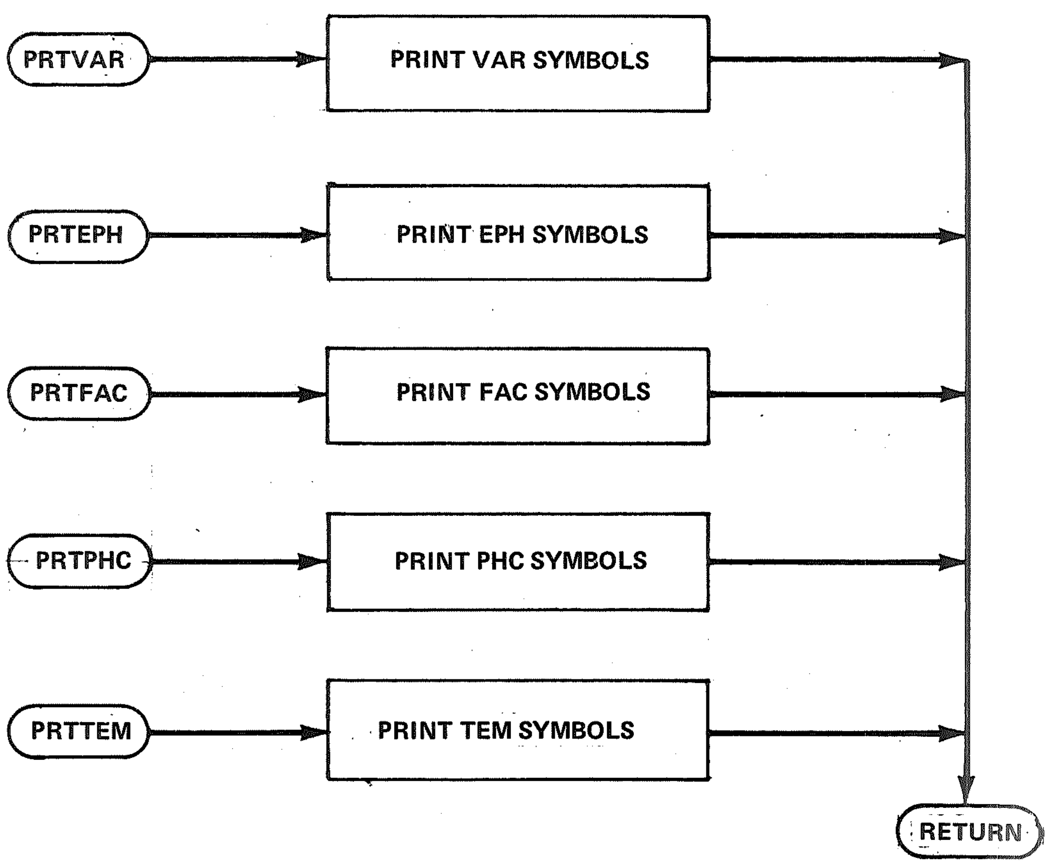
Called by: SIMXXX

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS*RFJK.PRTVAR

```
      C
      C  PURPOSE
      C
      C      TABULATE COMMON BLOCKS BY NAME
      C
             SUBROUTINE PRTVAR
      C
             INCLUDE QVAR, LIST
      C
             INCLUDE QTYPE
             INCLUDE QEPH
             INCLUDE QIEV
             INCLUDE QREADE
             INCLUDE EEPH
             INCLUDE EVARI
             INCLUDE EVARL
             INCLUDE EVARS
             INCLUDE EVARV
             INCLUDE EVARIO
      C
             NAME LIST / VARS /          TEPEJD, TEPEJS, TEPOJD, TEPOJS,
           1    TTRB,   TTRE,               TEVI,   TRCI,   TTRI,   CT2,
           2            DTAI,   DTAIDN, DTAIMN, DTAIUP,                 ERCAVM,
           3                    GVMB,   GVME,           HRCAVM,
           4    QVMB  , QVME  ,
           5    RRCAVM, RRCDVM, RSCAVM, RSCDVM,
           6    RTOPCM, RTOPRF,                 RUTOAU,                 TTO,
           7    TUTODA,         VUTOAD
      C
             NAME LIST / VARV /
           1    ERCBVM, HRCBVM,
           2    RRCIVM, RRCBVM, VRCBVM, RRCEVM, VRCEVM,
           3    RSCBIN, VSCBIN, RSCEIN, VSCEIN, RSCDDK, VSCDDK,
           4    RSCBVM, VSCBVM, RSCEVM, VSCEVM, RVMBIN, VVMBIN, RVMEIN, VVMEIN,
      C
             NAME LIST / VARI /          IEVCUR, IEVFLG, IEVM   ,
           1    IEVN  , IEVSTP, IEVSTR, INCNTR,
           2    IOCDR,  IOEPH,  IOMES,  IOPCH,  IOPRL,  IOPRT,  IOROL,
           3    IPLATM, IPLOTM, IPNOWM, ITRAT,  JKN,    KEPMAX,
           4    KEARTH, KEMB  , KJUP  , KMARS , KMERC , KMOON , KNEP  , KPLUTO,
           5    KSAT  , KSUN  , KURAN , KVAR  , MVAR  , NMAX  ,
             NAME LIST / VARL   /
           5    LIMPEV, LINTDN, LLDCOM, LOGOUT, LPRBKP, LPRCOM, LPREPH,
           6    LPRFAC, LPRLAT, LPRNOW, LPRPHC, LPRVAR, LRESKP, LSTDDK, LXECEV,
      C
             WRITE (IOPRT, VARI)
             WRITE (IOPRT, VARL)
             WRITE (IOPRT, VARS)
             WRITE (IOPRT, VARV)
                WRITE (IOPRT, 25) IDTRAJ, IRTIME, IEVREP, IEVRUN, IEVCUR,
           1          IEVSTR, IEVSTP
        25      FORMAT (' IDTRAJ ', 3A6, ', IRTIME ', 3A6, ', IEVREP', A7, '-',
           1          A6, ', IEVRUN', A7, '-', A6, ','/
           2          ' IEVCUR', A7, ', IEVSTR', A7, ', IEVSTP', A7)
                WRITE (IOPRT, 30) ((IEV(J,I), J = 3, 7), EVN(1,I), I = 1, IEVN)
        30      FORMAT (' EVENTS TABLE'/ (2A8, 3I8, G26.18))
```

```
          RETURN
C
C
C         PRINT BLOCK EPH BY NAME
C
          ENTRY PRTEPH
C
          NAME LIST / RDEPRT /  AU,  RE,  TPD, EMRAT, ICW, ICENT, IREQ, JK
C
          WRITE (IOPRT, RDEPRT)
          WRITE (IOPRT, 152)
          DO 150 I = 1, JKN
             K = JK(I)
  150        WRITE (IOPRT, 154) K, (EPH(J,K), J = 1, JKSPEC),
     1          (EPH(J,K), J = 17, 19)
  152     FORMAT ('0GRAVITATING POINT DESCRIPTIONS')
  154     FORMAT (I5, ' ROOEIN',G30.18,2G36.18/ 6X, 'VOOEIN',G30.18,2G36.18/
     1       6X, 'RSCEOO', G30.18, 2G36.18/ 6X, 'VSCEOO', G30.18, 2G36.18/
     2       6X, 'RSCAOO', G30.18, ' RSCDOO', G29.18, ' GOO   ', G29.18/
     3       6X, 'GOOR  ', G30.18, ' DANEOO', G29.18, ' DOBEOO', G29.18/
     4       6X, 'DLNEOO', G30.18, ' SPLEOO', G29.18, ' ROOD  ', G29.18/
     5       6X, 'RMNEOO', G30.18, 2G36.18//)
  160     CONTINUE
C
          LPREPH = .FALSE.
          RETURN
          END
```

VMASS*TPF$.PRTFAC

```
C
C  PURPOSE
C
C      TABULATE COMMON BLOCKS BY NAME
C
       SUBROUTINE PRTFAC
C
       INCLUDE QVAR, LIST
       INCLUDE QTYPE
       INCLUDE QIEV
       INCLUDE QPHC
       INCLUDE QTEM
       INCLUDE QFAC
       INCLUDE EVARI
       INCLUDE EVARL
       INCLUDE EVARIO
       INCLUDE ETEM
C
       NAME LIST / FACPRT /         FINITY, FINTES, FPT15  , FPT4   , FPT5   ,
     1   FRTO   , FTTO   , FTTOKM, F0     , F00001, F001   , F1     , F1PT1  ,
     2   F1PT5  , F10    , F11    , F12    , F13    , F14    , F1440  , F15    ,
     3   F2     , F2PT5  , F20    , F24    , F3     , F30    , F3600  , F4     ,
     4   F43200, F5     , F6     , F60    , F7     , F8     , F86400, F9     ,
     5   F9TNTH, DTOR   , RETOKM, RTOD   , AUTOKM
       WRITE (IOPRT, FACPRT)
       LPRFAC = .TRUE.
       RETURN
C
C
C      PRINT BLOCK PHC BY NAME
C
       ENTRY PRTPHC
C
C
       NAME LIST / PHCPRT / GS, PIMASS
       WRITE (IOPRT, PHCPRT)
       LPRPHC = .FALSE.
       RETURN
C
C
C      PRINT BLOCK TEM BY NAME
C
       ENTRY PRTTEM
C
C
       NAME LIST / TEMPRT /         TV1,     TV2,     TV3,     TV4,     TV5,
     1   TV6,     TV7,     TV8,     TV9,     TV10,    TV11,    SV1,     SV2,
     2   SV3,     SV4,     SV5,     SV6,     XV1,     XV2,     XV3,     XV4,
     3   XV5,     XV6,     T1,      T2,      T3,      T4,      T5,      T6,
     4   T7,      T8,      T9,      T10,     T11,     S1,      S2,      S3,
     5   S4,      S5,      S6,      S7,      S8,      S9,      X1,      X2,
     6   X3,      X4,      X5,      X6,      X7,      X8,      X9
       WRITE (IOPRT, TEMPRT)
       LPRTEM = .FALSE.
       RETURN
       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)

Program: RECORD

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

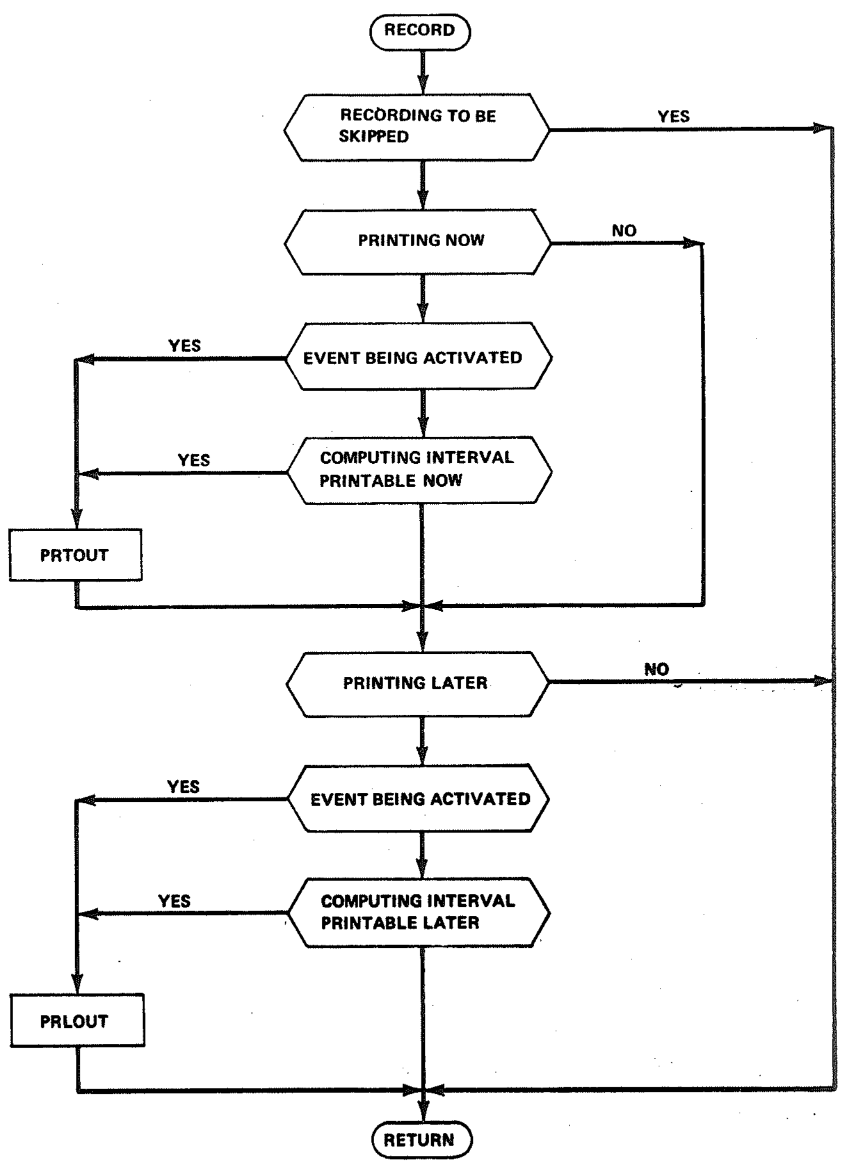
Called by: EVNACC, SIMXXX

Calls: PRLOUT, PRTOUT

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None



VMASS*RFJK.RECORD

```
C       RECORD   R JESSUP - BELLCOMM - 06 JAN 69
C
C  PURPOSE
C
C
C       MAKE RECORD OF THE RUN
C
C  INPUTS
C
C       LRESKP   RECORDING BEING SKIPPED (DURING TARGETING FOR EXAMPLE)
C       LPRNOW   PRINTING (VIA PRINT-LIST) OCCURRING DURING RUN
C       LPRLAT   PRINTING (VIA PRINT LIST) TO OCCUR LATER
C       IPLATM   INCREMENTAL COMPUTING STEPS PER PRINT LATER FILE WRITE
C       IPNOWM   INCREMENTAL COMPUTING STEPS PER CONCURRENT PRINTOUT CALL
C
C  OUTPUT
C
C       IOPRL    (PRLOUT) THE V MASS PROGRAM VARIABLES BLOCK - KVAR
C       IOPRT    (PRTOUT) EDITED PRINT OF THE PROGRAM RUN
C       INCNTR   INCREMENTAL COMPUTING STEPS COUNTER
C
        SUBROUTINE RECORD
C
        INCLUDE QVAR
        INCLUDE EVARI
        INCLUDE QTYPE
        INCLUDE EVARL
C
        IF (LRESKP) RETURN
        IF ((LXECEV .OR. MOD (INCNTR, IPNOWM) .EQ. 0) .AND. LPRNOW)
       1   CALL PRTOUT
        IF ((LXECEV .OR. MOD (INCNTR, IPLATM) .EQ. 0) .AND. LPRLAT)
       1   CALL PRLOUT
C
C
        RETURN
        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.

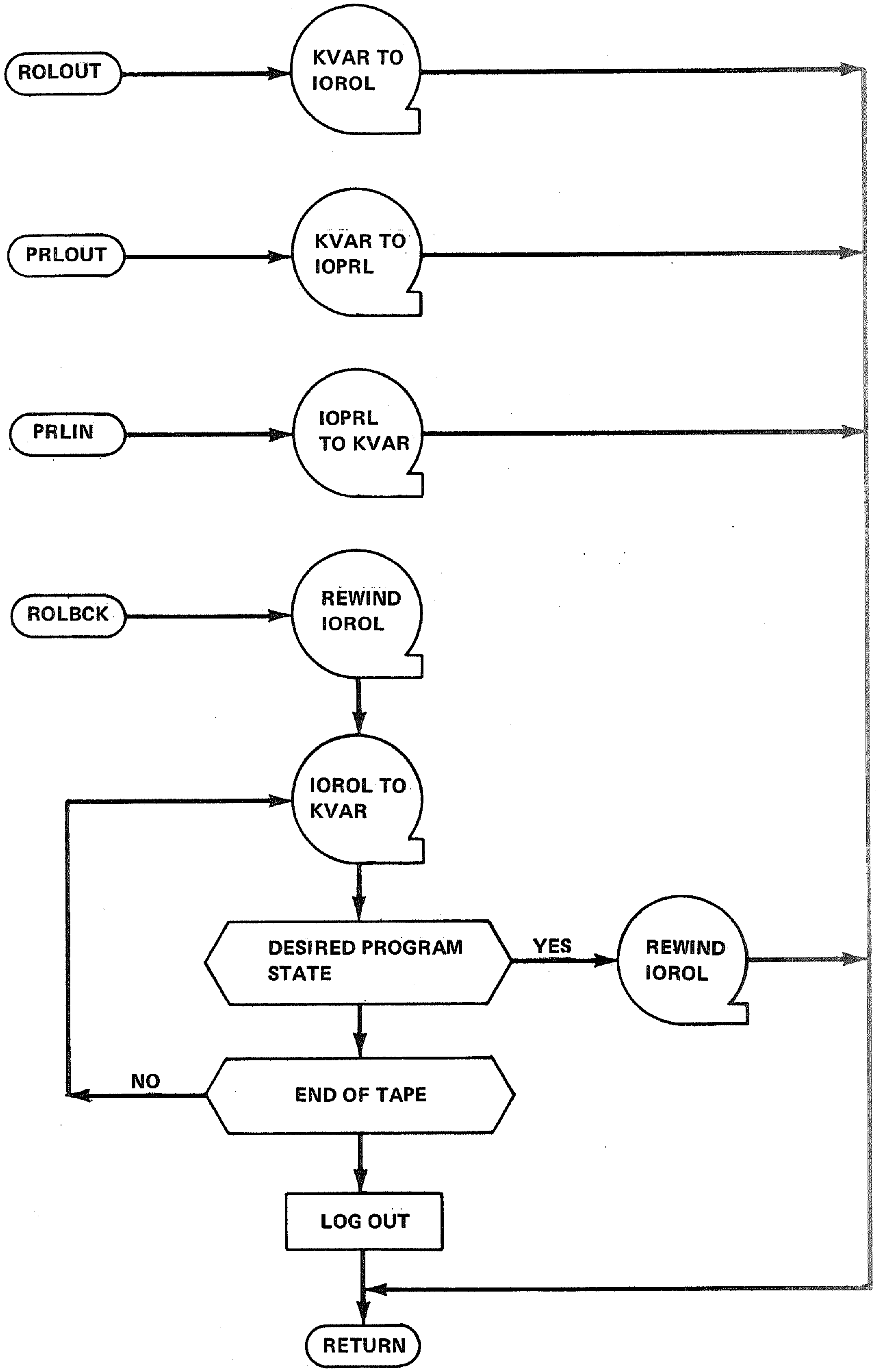
Called by: ROLOUT: EVNACC; ROLBCK; SETUP; PRLIN: PRTRAJ; PRLOUT: RECORD, SETUP

Calls: None

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

Language: Fortran subroutine

Computations: None


ROLOUT
KVAR TO IOROL
PRLOUT
KVAR TO IOPRL
PRLIN
IOPRL TO KVAR
ROLBCK
REWIND IOROL
IOROL TO KVAR
DESIRED PROGRAM STATE
YES
REWIND IOROL
NO
END OF TAPE
LOG OUT
RETURN

VMASS*RFJK.ROLLIO

```
C     ROLLIO   R JESSUP - BELLCOMM - 06 JAN 69
C
C  PURPOSE
C
C     IN THIS PROGRAM 'ROLOUT' AND 'PRLOUT' PLACE CURRENT V MASS SYSTE
C     STATE 'KVAR' ON SECONDARY FILES IOROL AND IOPRL, RESPECTIVELY.
C     'ROLBCK' AND 'PRLIN' READ SUCH A STATE BACK INTO THE COMPUTER.
C
C  INPUTS
C
C     IOPRL    (PRLIN) KVAR
C     IOROL    (ROLIN) KVAR
C
C  OUTPUT
C
C     IOPRL    (PRLOUT) KVAR AT EVENTS AND SELECTED COMPUTING INCREMEN
C                       BETWEEN EVENTS
C     IOROL    (ROLOUT) KVAR AT EVENTS
C
C
      SUBROUTINE ROLOUT
C
      INCLUDE QVAR
      INCLUDE QTYPE
      INCLUDE EVARI
      INCLUDE EVARL
      INCLUDE EVARIO
C
      WRITE (IOROL) KVAR
      RETURN
C
C
      ENTRY PRLOUT
      WRITE (IOPRL) KVAR
      RETURN
C
C
      ENTRY ROLBCK (I1)
      I2 = I1
      I3 = IEVCUR
      REWIND IOROL
   10 READ (IOROL) KVAR
      IF (IEVCUR .NE. I2) GO TO 20
         REWIND IOROL
         RETURN
   20 IF (IEVCUR .NE. I3) GO TO 10
         WRITE (IOPRT, 30)
         LOGOUT = .TRUE.
         RETURN
   30 FORMAT (' ROLBCK *** REQUESTED EVENT NOT ON ROLL BACK FILE')
C
C
      ENTRY PRLIN
      READ (IOPRL) KVAR
      RETURN
      END
```

Program: RTCLOK

RTCLOK (a)

Purpose: Obtain current value of computer real time clock

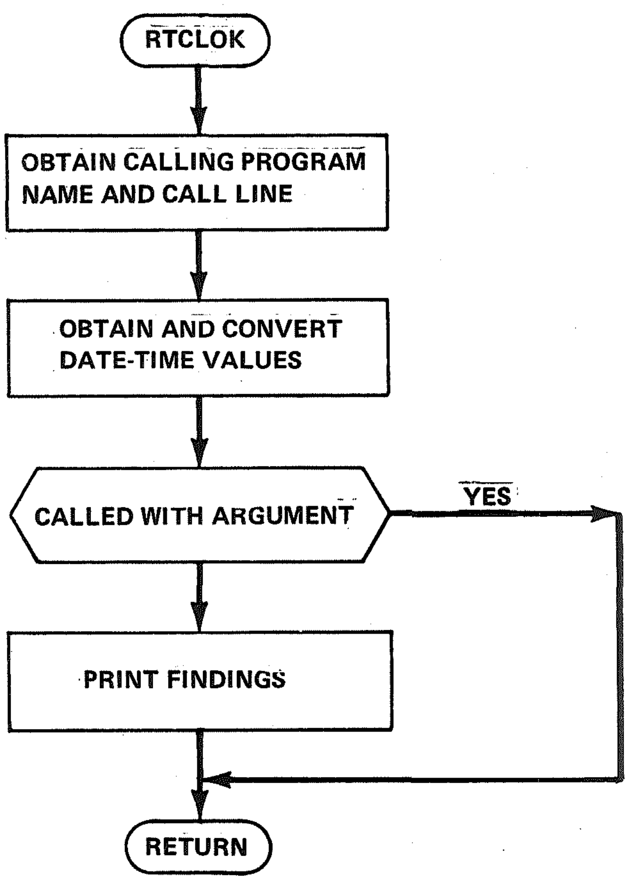
Called by: PRTIME, SETUP

Calls: None

Errors and Diagnostics: None

Language: 1108 Assembler subroutine

Computations: Convert clock readings to printable form.



VMASS*RFJK.RTCLOK

```
        .
        .  PURPOSE
        .
        .     OBTAIN CURRENT READING OF THE COMPUTER ''REAL TIME''CLOCK
        .
        .  CALLING SEQUENCE
        .
        .  FOR PRINT-OUT INCLUDING CALLING PROGRAM NAME AND CALL LOCATION
        .     CALL RTCLOK
        .
        .  FOR 18 CHARACTER RETURN VIA A IN THE FORM ''DA MON YR HR:MI:SE''
        .     CALL RTCLOK (A)
        .
        $(1)    LIT
        .         .    +0.    +1.    +2.    +3.    +4.    +5.    +6.    +7.    +8.    +9
        MSG     'RTCLOK ***  PRGNAM LINE NUMBER       DA MON YR HR:MI:SE.ONDS '
        .
        $(2)
        RTCLOK* EQU       $
                S         B11,('SAVB11')

                L         A1,0,B11         . INSERT LINE NUMBER CONVERTED TO FIELDATA
                L,XM      A2,5
                LSSL      A0,3
                LDSL      A0,3
                A,XM      A0,060
                JGD       A2,$-3
                S         A0,MSG+4
                SZ        A0               . INSERT  PROGRAM NAME (@@@@@@ FOR MAIN PR)
                SZ        MSG+2
                TNE,H2    A0,0,B11
                J         $+3
                L         A0,*0,B11
                S         A0,MSG+2
                ER        DATE$
                L         A2,(053000053000)         . :MI:SE
                DSL       A1,12
                SSC       A2,6
                DSL       A1,12
                SSC       A2,6
                S         A2,MSG+8
                L         A2,(005000005000)         .  YR HR
                DSL       A1,12
                SSC       A2,6
                DSL       A0,24
                SSL       A1,12
                DSL       A1,12
                SSC       A2,6
                S         A2,MSG+7
                L         B11,(1,0)
                TE,H2     A0,MON,*B11
                J         $-1
                L         A2,MON-1,B11
                L,XM      A3,5
                LDSC      A2,66
                DSL       A1,12
```

```
                S       A2,MSG+6
                L       B11,('SAVB11,)
                TNZ,H1  0,B11
                J       ONE
                P$RINT  MSG,8,1        . PRINT COMPOSED MESSAGE AND RETURN
                L       B11,('SAVB11,)
                J       1,B11
        .
        ONE     L       A0,0,B11
                L       A1,MSG+6
                S       A1,0,A0
                DL      A1,MSG+7
                DS      A1,1,A0
                J       2,B11
        $(3)
        CMON    INFO    2 3            . COMMON / CMON / MON(12)
        MON     RES     12
        .
        A0      EQU     014
        A1      EQU     A0+1
        A2      EQU     A0+2
        A3      EQU     A0+3
        B11     EQU     013
        C3      EQU     013
        H1      EQU     02
        H2      EQU     01
        R0      EQU     0100
        XM      EQU     017
        .
                END
```

Program: SCALE (s, u, v)

Purpose: Form product of a scalar and a vector

Called by: None

Calls: None

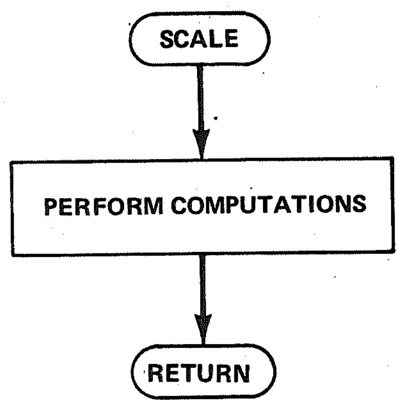
Errors and Diagnostics: None

Language: Fortran subroutine

Computations: For the vectors u and v compute

$$v_i = s\ u_i$$

for i = 1, 2, 3.



VMASS SUBROUTINE SCALE

```
VMASS*RFJK.SCALE
C
C  PURPOSE
C
C     FORM PRODUCT OF A SCALAR AND A VECTOR (3 - COMPONENTS)
C
C  INPUT
C
C     SCAL      SCALAR
C     VEC1      VECTOR INPUT
C
C  OUTPUT
C
C     VEC2      VECTOR RESULT
C
      SUBROUTINE SCALE (SCAL, VEC1, VEC2)
C
      INCLUDE QTYPE
      DIMENSION VEC1(3), VEC2(3)
C
      DO 10 I = 1, 3
10       VEC2(I) = SCAL * VEC1(I)
      RETURN
      END
```

Program: SETUP

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

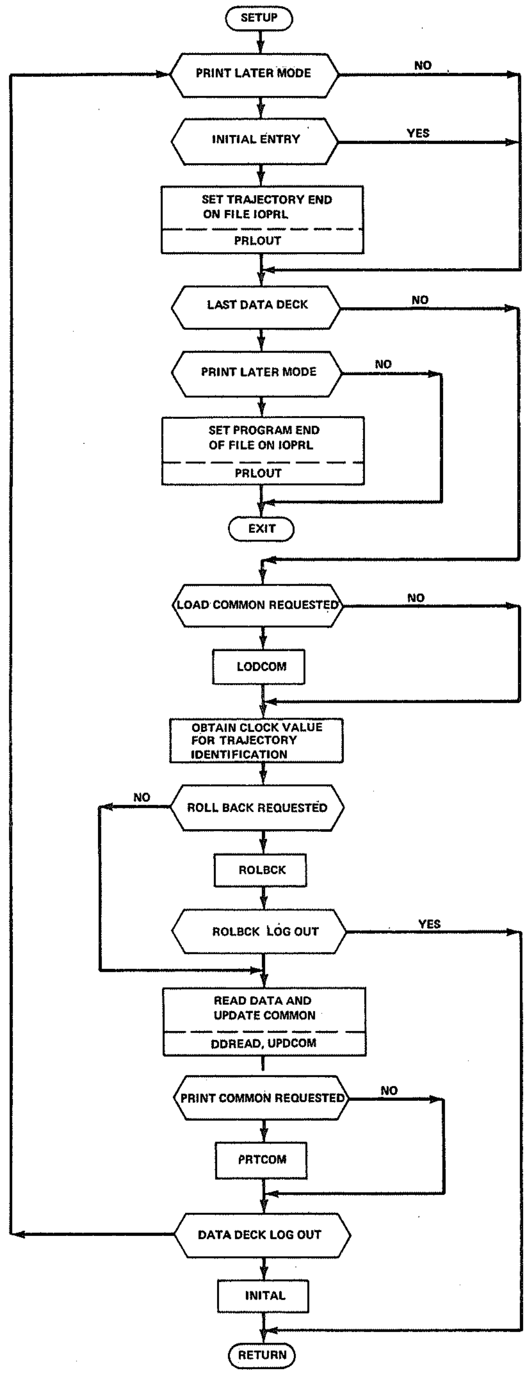
Called by: SIMXXX, PRTRAJ

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

Errors and Diagnostics: None

Language: Fortran subroutine

Computations: None


SETUP
PRINT LATER MODE
NO
INITIAL ENTRY
YES
SET TRAJECTORY END ON FILE IOPRL
PRLOUT
LAST DATA DECK
NO
PRINT LATER MODE
NO
SET PROGRAM END OF FILE ON IOPRL
PRLOUT
EXIT
LOAD COMMON REQUESTED
NO
LODCOM
OBTAIN CLOCK VALUE FOR TRAJECTORY IDENTIFICATION
NO
ROLL BACK REQUESTED
ROLBCK
ROLBCK LOG OUT
YES
READ DATA AND UPDATE COMMON
DDREAD, UPDCOM
PRINT COMMON REQUESTED
NO
PRTCOM
DATA DECK LOG OUT
INITAL
RETURN

VMASS*RFJK.SETUP

```
C
C  PURPOSE
C
C     THIS PROGRAM READS THE INPUT ''DATA DECK'' AND SETS UP THE
C     PROGRAM FOR A COMPUTER RUN.
C
C  INPUTS
C
C     THE ''DATA DECK'' VIA FORTRAN IOCDR (USUALLY THE ON-LINE
C     CARD READER)
C     IDTRAJ   IDENT FOR PREVIOUSLY DET TRAJ TO BE PRINTED BY THIS RUN
C
C  OUTPUTS
C
C     IRTIME   REAL TIME CLOCK READING AS DA MON YR HR:MI:SE FOR TRAJ ID
C
      SUBROUTINE SETUP
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE EVARI
      INCLUDE EVARL
      INCLUDE EVARIO
C
C     FOR THE PRINT LATER OPTION (LPRLAT TRUE) PROVIDE SOFTWARE TERMINAL
C     RECORD (IEVCUR 0) FOR THE LAST EVENT OF EACH TRAJECTORY SEGMENT.
C     MVAR 0 INDICATES TRAJECTORY BEGINNING.
C
    5 IF (.NOT. LPRLAT) GO TO 7
         IF (MVAR .EQ. 0) GO TO 7
            IEVCUR = 0
            CALL PRLOUT
C
C     STOP PROGRAM IF ALL DATA DECKS HAVE BEEN PROCESSED (LSTDDK TRUE)
C
    7 IF (.NOT. LSTDDK) GO TO 9
            IF (LPRLAT) CALL PRLOUT
            CALL EXIT
C
C     PRELOAD COMMON FROM STORAGE (LLDCOM TRUE). THE CALLING PROGRAM,
C     USUALLY THE EVENTS LIST, SETS LLDCOM TRUE ON INITIAL CALL TO SETUP
C     LLDCOM, SET FALSE BY LODCOM, REMAINS FALSE THROUGH SUBSEQUENT
C     INITIALIZATIONS UNLESS CHANGED BY THE VMASS USER.
C
    9 IF (LLDCOM) CALL LODCOM
C
C     OBTAIN COMPUTER SYSTEM DATE-TIME GROUP FOR 18 CHARACTER IDENTIFI-
C     CATION (IRTIME) OF THE TRAJECTORY SEGMENT TO BE CALCULATED. SEE
C     ALSO IDTRAJ IN THE PRINT LATER PROGRAM PRTRAJ.
C
      CALL RTCLOK (IRTIME)
      WRITE (IOPRT, 20) IRTIME
   20 FORMAT ('1VIRTUAL MASS TRAJECTORY ', 3A6)
C
C     ON INDICATION IEVREP NOT 0 RELOAD COMMON TO VALUES HELD ON THE
C     PROGRAM ROLL BACK FILE FOR THE EVENT GIVEN BY IEVREP. THE ROLL
```

```
      C      BACK FILE (IOROL) TRAJECTORY SEGMENTS MAY BE FROM THE CURRENT OR
      C      SOME PREVIOUS COMPUTER RUN.
      C
             IF (IEVREP(1) .EQ. 0) GO TO 30
                CALL ROLBCK (IEVREP)
                IF (.NOT. LOGOUT) GO TO 30
                   WRITE (IOMES, 22)
       22          FORMAT (' ROLBCK CALLED BY SETUP')
                   RETURN
      C
      C
       30    CALL DDREAD
             CALL UPDCOM
      C
      C      LPRNOW SET TRUE IN THE DATA DECK CAUSES PRINT OF COMMON AFTER
      C      INPUT READING. ROLL BACK OR PRELOADING OF COMMON CAN BE CONTROLL-
      C      ED FROM THE DATA DECK BY THE APPROPRIATE INDICATORS PLUS LOGOUT
      C      TRUE.
      C
             IF (LPRCOM) CALL PRTCOM
             IF (.NOT. LOGOUT) GO TO 100
                WRITE (IOMES, 40)
       40       FORMAT (' DDREAD CALLED BY SETUP')
                LOGOUT = .FALSE.
                GO TO 5
      C
      C      PROGRAM USER CAN EXTEND SETUP BY WRITING HIS OWN INITIAL ROUTINE
      C
       100   CALL INITAL
             RETURN
             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.

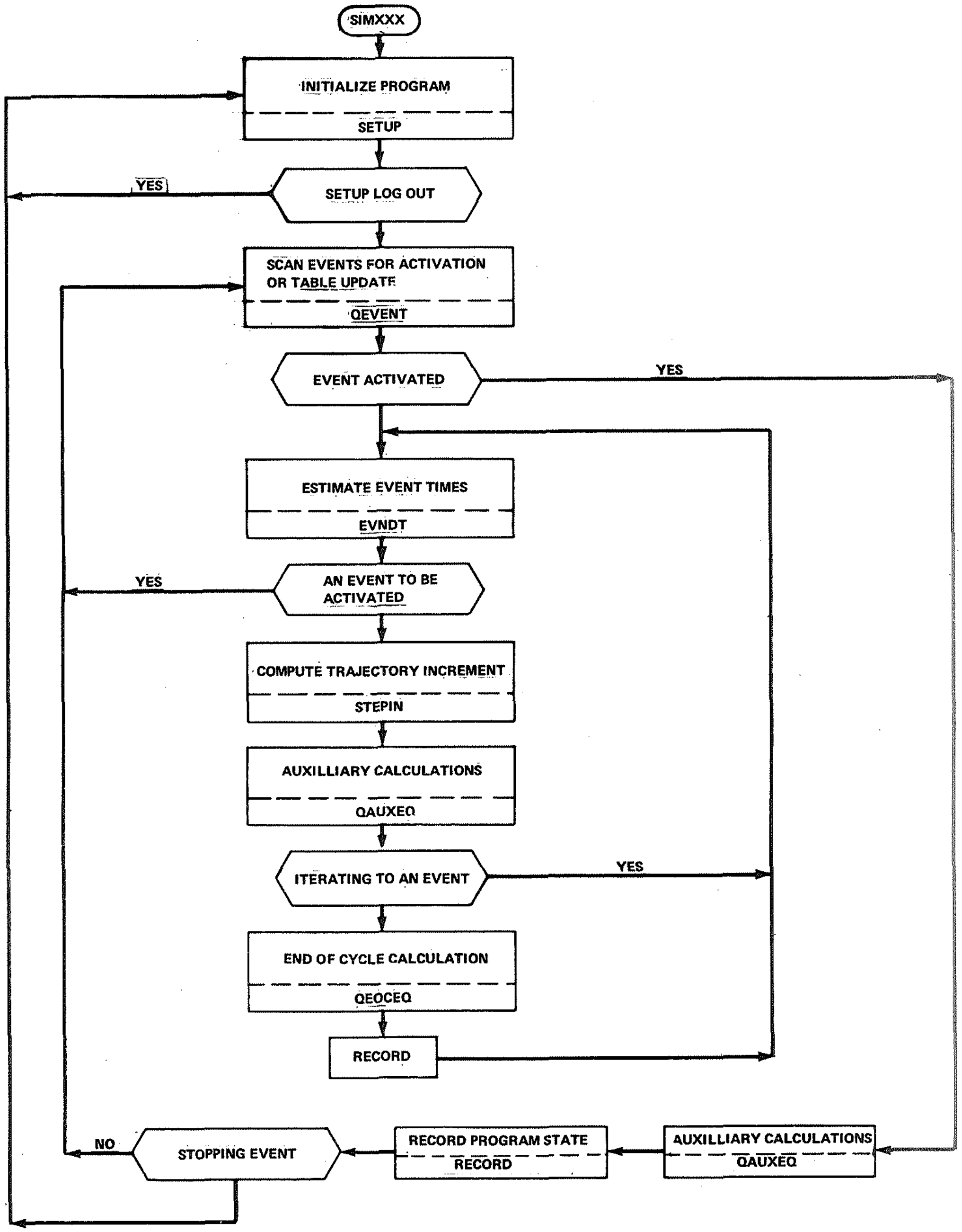
Called by: Separate computer execution.

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.


SIMXXX
INITIALIZE PROGRAM
SETUP
SETUP LOG OUT
YES
SCAN EVENTS FOR ACTIVATION OR TABLE UPDATE
QEVENT
EVENT ACTIVATED
YES
ESTIMATE EVENT TIMES
EVNDT
AN EVENT TO BE ACTIVATED
YES
COMPUTE TRAJECTORY INCREMENT
STEPIN
AUXILLIARY CALCULATIONS
QAUXEQ
ITERATING TO AN EVENT
YES
END OF CYCLE CALCULATION
QEOCEQ
RECORD
NO
STOPPING EVENT
RECORD PROGRAM STATE
RECORD
AUXILLIARY CALCULATIONS
QAUXEQ

VMASS*TPF$.SIMXXX

```
C       SIMXXX   R JESSUP - BELLCOMM - 06 JAN 69
C
C PURPOSE
C
C       COMPUTE A VIRTUAL MASS TRAJECTORY (PROPAGATE A SPACE CRAFT INITIAL
C       STATE FROM A STARTING TO A STOPPING EVENT)
C
C  INPUTS
C
C       (LODCOM) VALUES FOR FLAGS AND CONSTANTS REFLEC ING PROGRAMMED
C                CONVENTIONS FOR TYPICAL RUNS
C       (SETUP)  THE DATA-DECK GIVING THE INITIAL SPAC+ CRAFT STATE AND
C                THE STARTING AND STOPPING EVENTS (IEVSTR AND IEVSTP)
C       (STEPIN) COMPUTATIONS FOR INCREMENTAL STEPS IN SC STATE PROGATION
C       (EVNACC) EVENTS ACCOUNTING AND RECOGNITION
C       (AUXEQ)  EVENT ASSOCIATED AUXILLIARY COMPUTATIONS
C       (EOCEQ)  EV ASSOC AUX EQ FOR NORMAL COMPUTATION STEPS ONLY
C
C  OUTPUTS
C
C       (RECORD) PRINTOUT - GRAPHS - ''PRINT LATER'' TAPES
C
        INCLUDE QTYPE
        INCLUDE QVAR
        INCLUDE QFAC
        INCLUDE QEPH
        INCLUDE EVARI
        INCLUDE EVARL
        INCLUDE EVARS
        INCLUDE EVARV
        INCLUDE EVARIO
        INCLUDE QSTORE
        INCLUDE QTEM
C
C       ON FIRST DATA DECK INITIALIZE FOR RUN WITH PRELOAD OF COMMON.
C       INITIALIZE IN ACCORDANCE WITH DATA DECKS AND USERS CODING THERE-
C       AFTER.
C
        LLDCOM = .TRUE.
  9000  CALL SETUP
        IF (.NOT. LOGOUT) GO TO 9020
           WRITE (IOMES, 9010)
  9010     FORMAT (' SETUP CALLED BY MAIN PROGRAM')
           CALL PRTCOM
           GO TO 9000
  9020  CONTINUE
  9300  CONTINUE
        INCLUDE QEVENT, LIST
C
C       AFTER UPDATE OF EVENTS TABLE CHECK FOR NEXT EVENT AND RESUME
C       CALCULATIONS
C
  9911  CALL EVNDT
        IF (LXECEV) GO TO 9300
C
C       OBTAIN INCREMENTAL STEP IN TRAJECTORY SEGMENT
```

```
C
 9913 CALL STEPIN
      IF (.NOT. LOGOUT) GO TO 9915
         WRITE (IOMES, 9914)
 9914    FORMAT (' STEPIN CALLED FROM EVENTS LIST')
         CALL PRTVAR
         GO TO 9000
C
C     AFTER EACH INCREMENTAL STEP, WHICH MAY INCLUDE ACTIVATION AND ONE
C     TIME CALCULATIONS OF AN EVENT, PERFORM USERS SUPPLIED AUXILLIARY
C     EQUATIONS.
C
 9915 CALL AUXEQ
      IF (.NOT. LXECEV) GO TO 9920
C
C        ON ACTIVATION OF AN EVENT RECORD TRAJECTORY STATE AND, UNTIL
C        STOPPING EVENT IS REACHED, UPDATE EVENTS TABLE AND CONTINUE
C        INCREMENTAL STEPS.
C
         CALL RECORD
         LXECEV = .FALSE.
         IEVFLG = 1
         IF (IEVCUR .NE. IEVSTP) GO TO 9300
         GO TO 9000
C
C     AFTER INCREMENTAL STEPS NOT UNDER ADJUSTMENT TO REACH AN EVENT
C     (LIMPEV TRUE) PERFORM USERS SUPPLIED END CYCLE EQUATIONS.  PROCEED
C     TO NEXT STEP.
C
 9920 IF (LIMPEV) GO TO 9911
      CALL EOCEQ
      CALL RECORD
      GO TO 9911
C
C
      SUBROUTINE AUXEQ
C
C  PURPOSE
C
C     USERS SUPPLIED AUXILLIARY EQUATIONS
C
      INCLUDE QAUXEQ, LIST
      RETURN
C
C
      ENTRY EOCEQ
C
C  PURPOSE
C
C     USERS SUPPLIED END OF CYCLE EQUATIONS
C
      INCLUDE QEOCEQ, LIST
C
      RETURN
      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:

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

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

$$TTRE = TTRB + TRCI$$

$$RSCDVM_k = |RSCEVM_k|$$

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

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

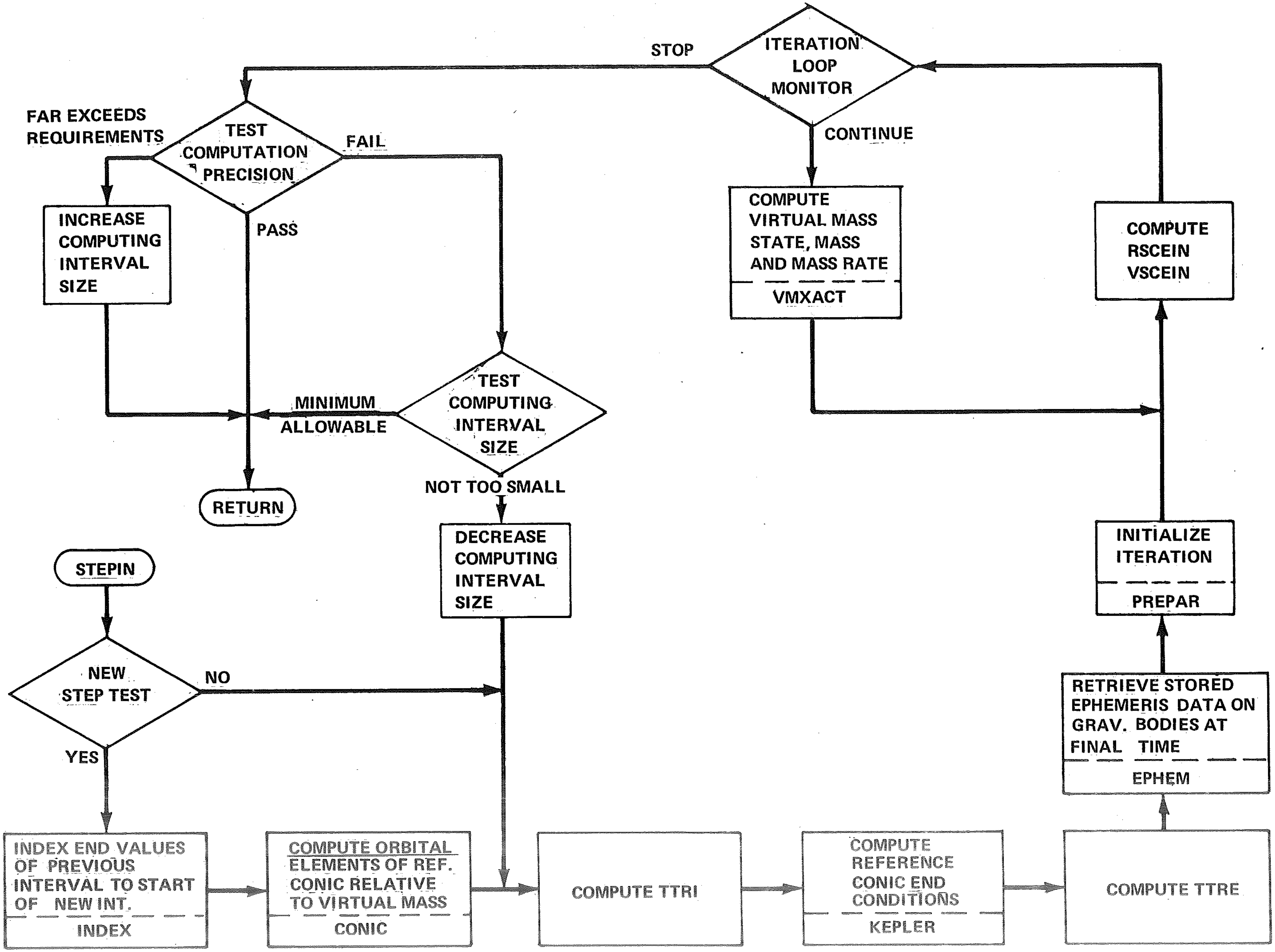
$$V_{2,k} = SV_{3,k-1}$$

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

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

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

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


ITERATION LOOP MONITOR
STOP
CONTINUE
FAR EXCEEDS REQUIREMENTS
TEST COMPUTATION PRECISION
FAIL
PASS
INCREASE COMPUTING INTERVAL SIZE
COMPUTE VIRTUAL MASS STATE, MASS AND MASS RATE
VMXACT
COMPUTE RSCEIN VSCEIN
MINIMUM ALLOWABLE
TEST COMPUTING INTERVAL SIZE
NOT TOO SMALL
RETURN
DECREASE COMPUTING INTERVAL SIZE
INITIALIZE ITERATION
PREPAR
STEPIN
NEW STEP TEST
NO
YES
RETRIEVE STORED EPHEMERIS DATA ON GRAV. BODIES AT FINAL TIME
EPHEM
INDEX END VALUES OF PREVIOUS INTERVAL TO START OF NEW INT.
INDEX
COMPUTE ORBITAL ELEMENTS OF REF. CONIC RELATIVE TO VIRTUAL MASS
CONIC
COMPUTE TTRI
COMPUTE REFERENCE CONIC END CONDITIONS
KEPLER
COMPUTE TTRE

MASS*TPF$.STEPIN

```
C     STEPIN   D NOVAK AND R JESSUP     BELLCOMM       03 FEB 69
C
C  PURPOSE
C
C     TO ORGANIZE A GROUP OF SUBROUTINES FOR PROGATING A SPACECRAFT
C     TRAJECTORY ACROSS ONE COMPUTATIONAL INTERVAL.
C     THIS TRAJECTORY SATISFIES THE VIRTUAL MASS EQUATIONS OF MOTION
C     ALONG WITH VARIOUS HIGHER DERIVATIVES THEREOF.  ALSO
C
C  INPUTS
C
C     LIMPEV   RECOMPUTE INTERVAL WITH PRECISION ITERATION TO PROJECTED
C              END POINT NOW THE TIME OF SOME EVENT
C     TEVI     ESTIMATE OF TIME INCREMENT TO NEXT EVENT
C     TRCI     REFERENCE CONIC TIME INCREMENT FOR INTERVAL
C     TTRB     TRAJECTORY TIME ORIGIN OF INTERVAL
C     DTAIDN   MULTIPLIER FOR DECREASING THE TRUE ANOMALLY STEP-SIZE
C     DTAIUP   MULTIPLIER FOR INCREASING THE TRUE ANOMALLY STEP-SIZE
C
C  OUTPUTS
C
C     RSCEIN   POST INTERVAL SPACE CRAFT POS REL TO INERTIAL COORD FRAME
C     VSCEIN              ''          VEL         ''
C     TTRE     TRAJECTORY TIME TERMINATION OF INTERVAL
C     DTAI     TRUE ANOMALLY STEP-SIZE DEFINING COMPUTING INTERVAL
C     LINTDN   COMPUTING INTERVAL IS DOWN FROM EARLIER VALUE
C
      SUBROUTINE STEPIN
C
      INCLUDE QTYPE
      INCLUDE QVAR
      INCLUDE QFAC
      INCLUDE QTEM
      INCLUDE EVARL
      INCLUDE EVARIO
      INCLUDE EVARS
      INCLUDE EVARV
C
      DIMENSION ICALL(4)
      DATA ICALL / '        CALLED BY STEPIN ' /
C
      LINTDN = .FALSE.
      IF (LIMPEV) GO TO 10
         CALL INDEX
         CALL CONIC
         TTO    = RTOPRF * FTTO * RRCAVM / VALUE (VRCBVM)
C
C     INCR   ***
C
C  PURPOSE
C
C     ESTABLISHES THE NORMAL COMPUTATION TIME INCREMENT ON THE BASIS OF
C     A TRUE ANOMOLY INCREMENT.  COMPARES THIS STEP SIZE WITH THE
C     ESTIMATED TIME TO THE NEXT EVENT AND SETS THE SMALLER OF THE TWO
C     AS THE DESIRED TIME INTERVAL.  ALSO SETS COUNTER ITRAT TO ZERO.
C
```

```
C  INPUTS
C
C     DTAI      REFERENCE CONIC TRUE ANOMOLY ANGLE OVER INTERVAL
C     TEVI      ESTIMATED TIME INTERVAL TO NEXT EVENT
C     TTO       TOLERANCE ON ACCURACY OF COMPUTED TIME INCREMENT
C
C  OUTPUTS
C
C     TEPI      DESIRED COMPUTING TIME INTERVAL
C     ITRAT     COUNTER CONTROLLING THE COMPUTATION INTERVAL CHANGES
C
 10   TTRI = MIN (TEVI, TTO * DTAI)
      ITRAT = 0
 20   CALL KEPLER
      IF (.NOT. LOGOUT) GO TO 30
         ICALL(1) = 'KEPLER'
         GO TO 500
 30   TTRE = TTRB + TRCI
      CALL EPHEM
      IF (.NOT. LOGOUT) GO TO 40
         ICALL(1) = 'EPHEM'
         GO TO 500
 40   CALL PREPAR
C
C     UPDATE *** S LOCATIONS FROM PREPAR
C
 60   T1     = DOT (RSCEVM, RSCEVM)
      RSCDVM = SQRT (T1)
      T2     = RSCDVM * T1
      T3 = (S3 * QVME - GVME * (S1 + S4 * DOT(RSCEVM,VSCEVM) / T1)) / T2
      T4 = S3 * GVME / T2
      T5 = S5 * GVME / T2
      T6 = F0
      DO 62 I = 1, 3
         TV1(I) = T3 * RSCEVM(I)  +  T4 * VSCEVM(I)
         TV2(I) = SV3(I)
         SV3(I) = S2 * TV1(I)  +  T5 * RSCEVM(I)
         T6     = T6  +  (TV2(I) - SV3(I)) ** 2
         RSCEIN(I) = SV1(I)   +  SV3(I)
 62      VSCEIN(I) = SV2(I)   + TV1(I)
      RTOPCM = SQRT (T6) / RRCDVM
      IF (ITRAT .GE. 2) GO TO 70
         ITRAT = ITRAT + 1
         CALL VMXACT
         GO TO 60
 70   IF (RTOPCM .GT. RTOPRF) GO TO 80
         IF (RTOPCM * FRTO .LT. RTOPRF) DTAI = DTAI * DTAIUP
         RETURN
 80   IF (DTAI .LE. DTAIMN) RETURN
         DTAI =  DTAI * DTAIDN
         LINTDN = .TRUE.
         GO TO 10
 500  WRITE (IOPRT, 510) ICALL
 510  FORMAT (1X, 4A6)
      RETURN
      END
```

Program: UNITV (a, b)

Purpose: Calculate b the unit vector of a.

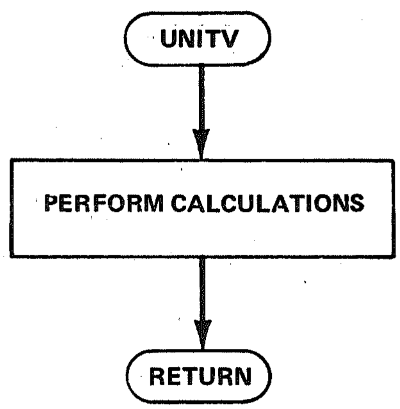
Called by: Kepler

Calls: None

Errors and Diagnostics: None

Language: Fortran subroutine

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



VMASS*RFJK.UNITV

```
      C
      C  PURPOSE
      C
      C     CALCULATE B THE UNIT VECTOR OF A
      C
            SUBROUTINE UNITV (A, B)
      C
            INCLUDE QTYPE
            INCLUDE QTEM
            DIMENSION A(3), B(3)
      C
            T1 = VALUE (A)
            DO 10 I = 1, 3
         10    B(I) = A(I) / T1
      C
            RETURN
            END
```

<u>Program</u>: s = VALUE (a)

<u>Purpose</u>: Calculate s the magnitude of the vector a.

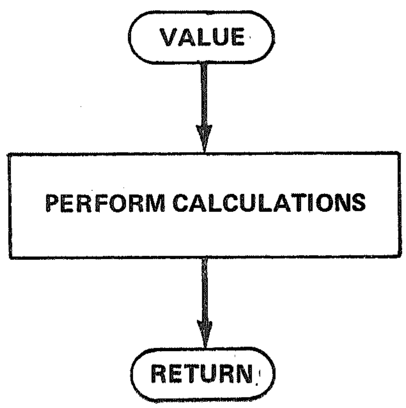
<u>Called by</u>: CONIC, KEPLER, STEPIN, UNITV

<u>Calls</u>: None

<u>Error and Diagnostics</u>: None

<u>Language</u>: Fortran function

<u>Calculations</u>: $s = a_1^2 + a_2^2 + a_3^2$



VMASS SUBROUTINE VALUE

```
VMASS*RFJK.VALUE
      C
      C   PURPOSE
      C
      C      CALCULATE MAGNITUDE OF A 3 COMPONENT VECTOR
      C
             FUNCTION VALUE (A)
      C
             INCLUDE QTYPE
             DIMENSION A(3)
      C
             VALUE = SQRT (A(1) ** 2  +  A(2) ** 2  +  A(3) ** 2)
             RETURN
             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:

$$RSCEOO_k = RSCEIN - ROOEIN_k$$

$$RSCDOO_k = |RSCEOO_k|$$

$$VSCEOO_k = VSCEIN - VOOEIN_k$$

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

$$VVMEIN = \frac{1}{t_1} \left(\sum_i \frac{GOO_k}{|RSCEDO_i|^3} (VOOEIN_i - T_6 \cdot ROOEIN_i) \right)$$

$$RSCEVM = RSCEIN - RVMEIN$$

$$VSCEVM = VSCEIN - VVMEIN$$

$$RSCDVM = |RSCEVM|$$

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

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

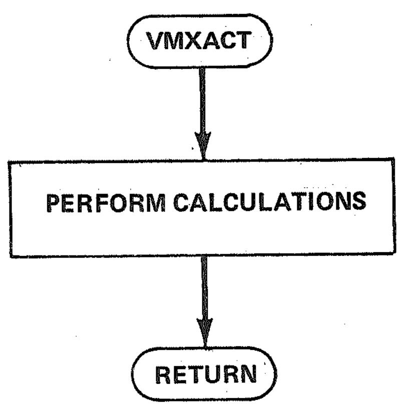
Where

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

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

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

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



VMASS*RFJK.VMXACT

```
C
C   PURPOSE
C
C      COMPUTES THE VIRTUAL MASS POSITION, MASS, VELOCITY, AND MASS RATE,
C      GIVEN THE INERTIAL POSITIONS AND VELOCIIIES OF THE SPACE CRAFT AND
C      ALL THE GRAVITATING MASS POINTS.  OUTPUTS THE VIRTUAL MASS DATA IN
C      INERTIAL COORDINATES AND GIVES THE SPACECRAFT POSITION AND
C      VELOCITY RELATIVE TO THE VIRTUAL MASS.
C
C   INPUTS
C
C      JK          LIST OF POINT MASS SUBSCRIPTS
C      JKN         NUMBER OF SUBSCRIPTS IN LIST JK
C      GJK         PRODUCT OF UNIVERSAL GRAVITATION CONSTANT AND MASS OF THE
C                  GRAVITATING POINT JK
C      RJKEIN      POSITION VECTOR OF MASS POINT JK RELATIVE TO AN INERTIAL
C                  COORDINATE FRAME
C      VJKEIN      INERTIAL VELOCITY VECTOR OF MASS POINT JK
C      RSCEIN      INERTIAL POSITION VECTOR OF SPACECRAFT
C      VSCEIN      INERTIAL VELOCITY VECTOR OF SPACECRAFT
C
C   OUTPUT
C
C      RVMEIN      POSITION VECTOR OF THE VIRTUAL MASS RELATIVE TO AN
C                  INERTIAL COORDINATE FRAME
C      VVMEIN      INERTIAL VELOCITY VECTOR OF THE VIRTUAL MASS
C      RSCEVM      POSITION OF SPACECRAFT RELATIVE TO THE VIRTUAL MASS
C      VSCEVM      VELOCITY OF SPACECRAFT RELATIVE TO THE VIRTUAL MASS
C      GVME        PRODUCT OF UNIVERSAL GRAVITATION CONST AND VIRTUAL MASS
C      QVME        PRODUCT OF UNIVERSAL GRAVITATION CONST AND RATE OF CHANGE
C                  OF THE VIRTUAL MASS MAGNITUDE
C      RSCDVM      DISTANCE OF SPACECRAFT FROM VIRTUAL MASS POINT
C      RSCEJK      POS OF SPACECRAFT RELATIVE TO MASS POINT JK
C      VSCEJK      VEL     ''
C      RSCDJK      DIST FROM SPACECRAFT TO MASS POINT JK
C
       SUBROUTINE VMXACT
C
       INCLUDE QTYPE
       INCLUDE QVAR
       INCLUDE QFAC
       INCLUDE QPHC
       INCLUDE QTEM
       INCLUDE QEPH
       INCLUDE EVARS
       INCLUDE EVARV
       INCLUDE EVARI
C
       T1  = F0
       T4 = F0
       DO 2 M = 1, 3
           TV1(M) = F0
 2         TV2(M) = F0
       DO 10 I = 1, JKN
           K = JKLOC (JK(I))
                DO 4 M = 1, 3
```

```
                          RSCEOO(K+M-1) = RSCEIN(M) - ROOEIN(K+M-1)
    4                     VSCEOO(K+M-1) = VSCEIN(M) - VOOEIN(K+M-1)
                   T2           = DOT  (RSCEOO(K), RSCEOO(K))
                   RSCDOO(K) = SQRT (T2)
                   T3           = GOO(K) / (T2 * RSCDOO(K))
                   T1           = T1 + T3
                   T6           = F3 * DOT (RSCEOO(K), VSCEOO(K)) / T2
                   T4           = T4 - T6 * T3
                   DO 6 M = 1, 3
                      TV1(M) = TV1(M) + ROOEIN(K+M-1) * T3
    6                 TV2(M) = TV2(M)
         1                   + T3 * (VOOEIN(K+M-1) - ROOEIN(K+M-1) * T6)
   10              CONTINUE
          DO 12 M = 1, 3
             RVMEIN(M) = TV1(M) / T1
             VVMEIN(M) = (TV2(M) - RVMEIN(M) * T4) / T1
             RSCEVM(M) = RSCEIN(M) - RVMEIN(M)
   12        VSCEVM(M) = VSCEIN(M) - VVMEIN(M)
          T5     = DOT (RSCEVM, RSCEVM)
          RSCDVM = SQRT (T5)
          GVME   = T1 * RSCDVM * T5
          QVME   = GVME * (F3 * DOT (RSCEVM, VSCEVM) / T5 + T4 / T1)
C
          RETURN
          END
```

@ 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. Arenstorf/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

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. Hinners
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