NASA TECHNICAL NOTE

NASA TN D-1730

NASA TN D-1730

THE N-BODY CODE – A GENERAL FORTRAN CODE FOR THE NUMERICAL SOLUTION OF SPACE MECHANICS PROBLEMS ON AN IBM 7090 COMPUTER

by William C. Strack and Vearl N. Huff

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1963

TECHNICAL NOTE D-1730

THE N-BODY CODE - A GENERAL FORTRAN CODE FOR THE NUMERICAL SOLUTION OF SPACE MECHANICS PROBLEMS ON AN IBM 7090 COMPUTER

By William C. Strack and Vearl N. Huff

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

CONTENTS

APPENDIXES

THE N-BODY CODE - A GENERAL FORTRAN CODE FOR THE NUMERICAL SOLUTION OF SPACE MECHANICS PROBLEMS ON AN IBM 7090 COMPUTER

By William C. Strack and Vearl N. Huff

SUMMARY

A general astronomical integration code designed for a large class of problems in space mechanics that may be solved by numerical integration is described. The equations of motion provide for the effects of up to eight gravitating celestial bodies, oblateness and aerodynamic forces from the celestial body at the problem origin, propulsion system thrust, and rotation of the body at the origin. The coding in this report is intended for use on an IBM 7090. An earlier version was reported in NASA Technical Note D-1455 and was designed for use on an IBM 704.

INTRODUCTION

The general problems of space mechanics (i.e., n-bodies plus nonconservative forces such as thrust) cannot be solved analytically. Therefore, numerical integration through the use of computing machinery is usually employed.

Several codes have been written for the numerical solution of problems in orbit mechanics; for example, the Themis Code of reference 1 is a double-precision code intended primarily for close satellites or interplanetary coasting flight. Reference 2 describes a space-trajectory program of considerable merit. A listing of several other trajectory codes may be found in reference 3.

The general purpose code described herein has several distinctive features not all of which are found in any one of the previously available codes. As described herein, this code is designed to operate on an IBM 7090 computer that has a 32,000 word (32 K) memory. The fact that the program is written in FORTRAN should make it applicable to installations having other types of equipment that accept the FORTRAN language. An earlier version of this program designed for an IBM 704 with an 8-K core and at least 1 K of drum storage has been previously published in reference 4. This report has incorporated a number of improvements. The most important ones are: (1) only one core load is required, (2) all stage data for multistage vehicles may be loaded simultaneously, (3) the third harmonic term is included in the Earth's oblateness equations, and (4) there are additional program controls available that provide increased flexibility.

The program is compartmented into 25 subroutines to facilitate modifications for specific problems. The integration is carried out in either rectangular coordinates or orbit elements at the option of the user. A compact ephemeris that occupies about one-seventh of a reel of tape is utilized for positions and velocities of the planets (except Mercury) and the Moon. An atmosphere is included so that aerodynamic forces may be considered.

STATEMENT OF PROBLEM

The problem to be solved may be stated as follows: Given certain initial conditions, compute, using three degrees of freedom, the path of an object, such as a space vehicle, subject to any or all of the following forces:

Origin body gravitational field

Other celestial body gravitational fields

Propulsive thrust

Aerodynamic forces

Any other defined forces

Alternately, in equation form, with respect to a noninertial Cartesian coordinate system,

$$\ddot{\vec{r}} = \nabla U + \left[k^2 \sum_{i=1}^{n} m_i \nabla\left(\left|\frac{1}{\vec{r} - \vec{r}_i}\right| - \frac{\vec{r} \cdot \vec{r}_i}{r_i^3}\right)\right] + \frac{\vec{F}}{m} + \frac{\vec{L}}{m} + \frac{\vec{D}}{m} + \frac{\vec{X}}{m} \tag{1}$$

where n equals the number of perturbating bodies and ∇ denotes the del operator. (All symbols are defined in appendix A.)

Origin Body Gravitational Field and Oblateness Perturbations

The first term, ∇U, in the equation of motion (eq. (1)) represents the gravitational forces due to the origin body. When the origin body is spherical and made up of homogeneous layers, this term becomes simply $-\mu\vec{r}/r^3$. In the case of the Earth, however, the effect of oblateness may be important, and additional terms must be added to account for the oblateness effects. The expression for the gravitational potential U of an oblate spheroid may be written, according to reference 5, as

$$U = \frac{k^2 m_r}{r}\left\{1 + J\left(\frac{R_r}{r}\right)^2\left[\frac{1}{3} - \left(\frac{z}{r}\right)^2\right] + H\left(\frac{R_r}{r}\right)^3\left[\frac{3}{5} - \left(\frac{z}{r}\right)^2\right]\frac{z}{r} + \frac{\mathscr{D}}{35}\left(\frac{R_r}{r}\right)^4\left[3 - 30\left(\frac{z}{r}\right)^2 + 35\left(\frac{z}{r}\right)^4\right]\right\} \tag{2}$$

where the x,y plane lies in the equatorial plane. The components of gravitational acceleration are as follows:

$$
\begin{aligned}
U_x = \frac{\partial U}{\partial x} = \frac{k^2 m_r}{r^2} \Bigg\{ & -1 + J \left(\frac{R_r}{r}\right)^2 \left[5 \left(\frac{z}{r}\right)^2 - 1\right] + H \left(\frac{R_r}{r}\right)^3 \left[7 \left(\frac{z}{r}\right)^2 - 3\right] \frac{z}{r} \\
& + \mathscr{D} \left(\frac{R_r}{r}\right)^4 \left[- \frac{3}{7} + 6 \left(\frac{z}{r}\right)^2 - 9 \left(\frac{z}{r}\right)^4\right] \Bigg\} \frac{x}{r} \\
U_y = \frac{\partial U}{\partial y} = \frac{k^2 m_r}{r^2} \Bigg\{ & -1 + J \left(\frac{R_r}{r}\right)^2 \left[5 \left(\frac{z}{r}\right)^2 - 1\right] + H \left(\frac{R_r}{r}\right)^3 \left[7 \left(\frac{z}{r}\right)^2 - 3\right] \frac{z}{r} \\
& + \mathscr{D} \left(\frac{R_r}{r}\right)^4 \left[- \frac{3}{7} + 6 \left(\frac{z}{r}\right)^2 - 9 \left(\frac{z}{r}\right)^4\right] \Bigg\} \frac{y}{r} \\
U_z = \frac{\partial U}{\partial z} = \frac{k^2 m_r}{r^2} \Bigg\{ & -1 + J \left(\frac{R_r}{r}\right)^2 \left[5 \left(\frac{z}{r}\right)^2 - 3\right] + H \left(\frac{R_r}{r}\right)^3 \left[7 \left(\frac{z}{r}\right)^2 - 6 + \frac{3}{5} \left(\frac{r}{z}\right)^2\right] \frac{z}{r} \\
& + \mathscr{D} \left(\frac{R_r}{r}\right)^4 \left[- \frac{15}{7} + 10 \left(\frac{z}{r}\right)^2 - 9 \left(\frac{z}{r}\right)^4\right] \Bigg\} \frac{z}{r}
\end{aligned}
\tag{3}
$$

The first terms exist for a spherical planet composed of concentric layers of uniform density. The terms containing J, H, and $\mathscr{D}$ are frequently called the second, third, and fourth harmonic terms, respectively, while J, H, and $\mathscr{D}$ are known as the harmonic coefficients.

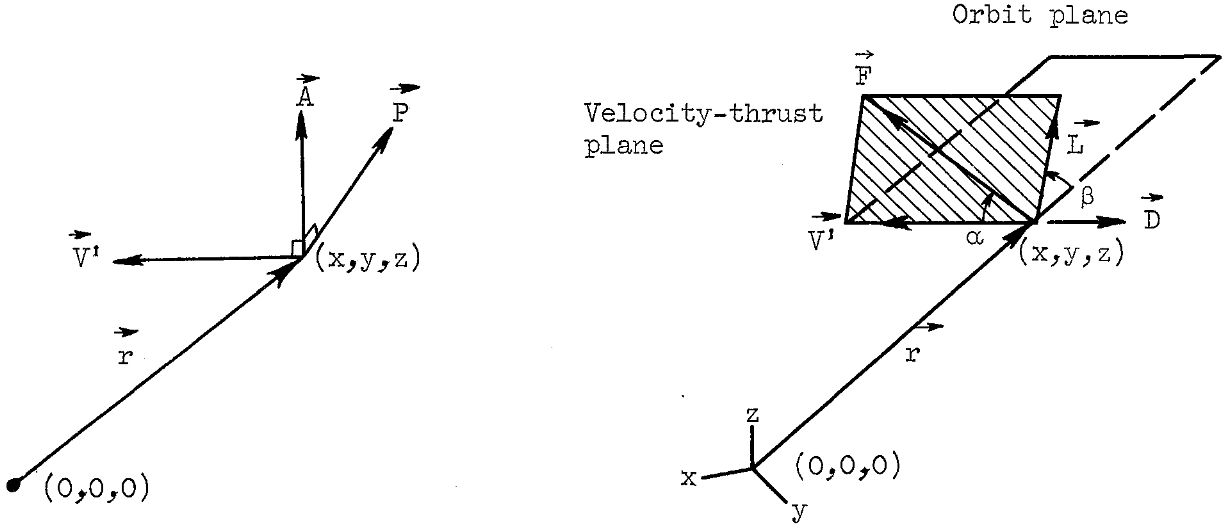
It is expected that oblateness perturbations need to be computed only for the origin body, since at large distances, such as that between celestial bodies, the gravitational field of an oblate body is essentially an inverse-square field. Consideration of oblate bodies other than the Earth requires only different values of J, H, and $\mathscr{D}$ if that body's rotation axis is parallel to the z-axis. When the body has triaxial asymmetry or when the z-axis cannot conveniently be alined with the rotation axis of the origin body, the equations must be extended if oblateness is to be included.

Celestial Body Perturbations

The presence of more than one gravitating body in addition to the object results in the inclusion of the second term of equation (1). The evaluation of this term requires a knowledge of the positions of the bodies as a function of time. The degree of precision desired determines the method to be used to obtain the positions such as elements of ellipses or an ephemeris.

Propulsive Thrust

The propulsive acceleration is completely specified by a direction and a magnitude. The thrust direction may be referred to the velocity vector by two angles: α, the angle between the velocity and the thrust vectors and β, the angle between the orbit plane and the velocity-thrust plane. The sense of each angle is indicated in sketch (a).



(a)

The velocity may be referenced with respect to one of several coordinate systems. If the computation refers to a takeoff of a rocket or winged vehicle, the coordinate system rotating with the Earth may be preferred. In such cases the relative velocity (i.e., the velocity of the object relative to the atmosphere) will serve to orient the thrust vector. Resolution of the thrust-vector components along the x,y,z axes is shown in appendix B.

The thrust magnitude for most types of problems is of the form

$$F = F(\dot{m}, I, t, \ldots)$$

Any such function (or even a more complicated relation) may quite easily be inserted into the program. For many space powerplants, the rocket engine thrust equation

$$F = -\dot{m}Ig_c - PA_e \tag{4}$$

is sufficient and is used as a standard in the present program.

Aerodynamic Forces

The aerodynamic forces are usually divided into the two components, lift and drag. The drag force is directed opposite to the relative wind vector, and the lift vector is perpendicular to the relative wind vector. The angles α and β, defined in the previous section, serve as the angles of attack and roll, respectively. Yaw effects are not considered. Resolution of the lift and drag vectors into components along the x,y,z axes is given in appendix B.

The magnitudes of the lift and drag forces may be conveniently determined through use of a tabular group of coefficients in relatively simple equations. The lift and drag magnitudes may then be expressed (as is usual in aerodynamics) as

$$L = C_L(\alpha, N_M)qS \tag{5}$$

$$D = C_D(C_L, N_M)qS \tag{6}$$

where

$\alpha = \alpha(t)$

$C_L = f_1(N_M)\sin\alpha$

$C_D = C_{D,o} + C_{D,i} = C_{D,o}(N_M) + f_2(N_M)C_L^2$

$q = \frac{1}{2}\rho(V')^2$

$\rho = \rho(P,T) = \rho(h)$

If $\alpha(t)$, $C_{D,o}(N_M)$, $f_1(N_M)$, and $f_2(N_M)$ are assumed to be quadratic functions and β is assumed to be constant, the expressions for α, β, C_L, $C_{D,o}$, and $C_{D,i}$ become

$$\alpha = a_{11} + a_{12}t + a_{13}t^2$$

$$\beta = \beta_0$$

$$C_L = \left(a_{21} + a_{22}N_M + a_{23}N_M^2\right)\sin\alpha$$

$$C_{D,o} = a_{31} + a_{32}N_M + a_{33}N_M^2$$

$$C_{D,i} = \left(a_{41} + a_{42}N_M + a_{43}N_M^2\right)C_L^2$$

where the quadratic constants $a_{i,j}$ may have different values for different regions of the independent variables t and N_M.

It should be remembered that these choices are arbitrary and are not restrictive because other functions may easily be used by simply changing the equations where they appear in the program. In fact, any propulsion system and aerodynamic configuration can presumably be incorporated by writing proper thrust and aerodynamic subroutines.

Pressure, temperature, and density are determined as functions of altitude in accordance with the U.S. Standard Atmosphere, 1962. The atmospheric data are in the form of a short table, which may be altered conveniently to account for a different atmospheric model.

Other Forces

The $\vec{X}$ forces may be any forces such as electrostatic, magnetic, or solar radiation pressure that affect the trajectory. While these forces are not considered further herein, their inclusion would usually be feasible and would be similar to thrust, lift, and drag.

METHOD OF SOLUTION

A description of several numerical integration techniques and their relative merits are contained in reference 6. A straightforward method for finding the position of the object as a function of time is to integrate the total acceleration of the object expressed in rectangular components. An example of this method is Cowell's method (ref. 6).

However, when the system under investigation consists of two nonoblate bodies (one of which is the object) with no forces other than gravitational attraction forces, an exact analytical solution for the motion of the body exists. Further, if the conditions of the actual problem are such as to approximate the two-body problem closely, another approach is to use the exact two-body solution as a basis and simply integrate the changes in the two-body parameters, since they should be slowly varying. This technique, sometimes called the "variation of parameters," will be referred to as "integration of orbit elements."

Since problems both remote and near to the exact two-body problem are encountered in orbit mechanics and since either type of problem is solved more efficiently by using the technique most suitably applicable, it was considered desirable to use either of the previously mentioned integration techniques at will. Accordingly, two methods of integration are provided in the program, namely, rectangular coordinates and orbit elements.

Integration Variables

To use either of these integration techniques, it is necessary to select a suitable set of variables. Because a differential equation may determine the mass of the object (i.e., spacecraft), mass has been selected as a variable to be integrated. Selection of the remaining parameters follows in the subsequent paragraphs.

Rectangular coordinates. - In the first technique, the total acceleration components $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are integrated to obtain x, y, and z where x, y, and z are the rectangular components of the origin to object radius $\vec{r}$. The positive x-axis points in the direction of the mean vernal equinox of 1950.0. The positive y-axis lies in the mean equator of 1950.0 and is perpendicular to and counterclockwise from the positive x-axis. The z-axis points north and completes the righthanded orthogonal set. The integration in rectangular coordinates involves numerical solution of three second-order differential equations; that is, a double integration is required for integrating the accelerations to obtain velocities and the velocities to obtain positions. The rectangular variables have advantages of complete generality and a minimum amount of computing per step.

Orbit elements. - In the variation-of-parameters technique, a set of six independent two-body parameters called orbit elements are integrated. These six parameters may be arbitrarily chosen from a host of possibilities. The set selected for this program is composed of the eccentricity e, the argument of pericenter ω, the equatorial longitude of ascending node Ω, the inclination of the orbit plane to the equatorial plane i, the mean anomaly M, and the semilatus rectum p. The transformation equations from orbit elements to rectangular coordinates are given in appendix C.

The integration of orbit elements requires the numerical solution of six first-order differential equations. The rather involved transformation by which the three second-order differential equations in $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are reduced to six first-order equations in $\dot{e}$, $\dot{\omega}$, $\dot{\Omega}$, $\dot{i}$, $\dot{M}$, and $\dot{p}$ is contained in reference 7. Integration in orbit elements is frequently advantageous because the smaller orbit-element derivatives may permit larger integration intervals that result in fewer steps. In the special case of two-body motion, the derivatives are zero (except $\dot{M}$, which is a constant).

Mathematical difficulties may arise occasionally with most sets of orbit elements. In particular, for the selected set, these occur when e approaches unity (parabolic trajectory), which causes a loss of numerical accuracy in the frequently used quantity $(1 - e^2)$, and when an asymptote is approached too closely, which causes numerical difficulties in the iterative solution for eccentric anomaly from Kepler's equation. The selected solution to these difficulties is to shift temporarily to rectangular-coordinate integration whenever the difficulty arises.

Integration Method

It is clear that regardless of the choice of integration technique, the magnitudes of the derivatives of the variables to be integrated may vary considerably along the trajectory. With fixed step size (constant intervals in time), the integration scheme will take unnecessary steps in the regions where the changes in the derivatives are small and thus will waste computing time and increase roundoff error. When the derivatives are large and change rapidly, a fixed step size will result in large truncation error (error due to excessive step size). Thus, in the interest of computing accuracy and economy, use of variable step size along the trajectory becomes desirable.

One of the integration schemes that allows variable step-size control to be incorporated easily is the Runge-Kutta scheme. For this and other reasons, it was decided to use a fourth-order Runge-Kutta method with variable step-size control.

Truncation error and step size may be controlled by examining the relative errors between the fourth-order Runge-Kutta integration scheme and a lower-order integration procedure. The arbitrarily chosen low-order integration scheme was an unequal-interval Simpson rule method. Details of the fourth-order Runge-Kutta integration method and the step-size control are given in appendix D. Roundoff error may be reduced by accumulating the integration variables in double precision.

Origin Translation

As noted previously, machine computing time and roundoff error may be minimized by maximizing the integration interval. The largest intervals are possible in orbit elements when the celestial body at the problem origin is the one that has the greatest influence on the vehicle motion. For this and sometimes other reasons, it may become desirable to translate the problem origin occasionally as the vehicle moves along its path.

Such translations of the origin may be made when the object enters a body's "sphere of influence," that is, the sphere about a body within which the greatest influence upon the object is due to forces originating from that particular body. In this program, the orientation of the coordinate system is always alined with the system determined by the Earth's mean equator and equinox of 1950.0, as is standard in astronomy.

THE CODE AND ITS USAGE

The stated problem was programed in FORTRAN routines that are separately designed to accomplish one task but when combined form a complete program. This feature facilitates modifications.

The program is labeled as a general-purpose code, but an efficient general-purpose code cannot be a reality. As a result, this code is not especially general, but an attempt has been made to retain efficiency and to provide for easy modification of the routines to recover generality as needed. For example, the program is an "open system"; that is, it solves an initial-value problem. There is no link provided to obtain specific end conditions. Provision of this link is left to the user for his specific needs. In particular, when certain end conditions of a trajectory are to be met by determining the correct initial conditions (two-point boundary-value problem), the user may program an iteration scheme to compute initial conditions from end conditions of previous runs. Figure 1 is a simplified diagram that shows how the various major subprograms (and exits) are arranged.

In the following sections, the program is sometimes discussed in terms of the FORTRAN variables and routines. A glossary of these variables is given in appendix E.

Ephemerides

To determine the position of each celestial body, there is offered a choice between ellipses and a precision ephemeris. Any appropriate ellipse data may be used, and an example of such data is given in table I.

The precision-ephemeris tape that is used in the program was so made that position and velocity were obtainable through the use of a fifth-order polynomial whose coefficients are stored on tape. The details concerning the making of the tape and its structure are given in appendix F. This master tape is a merged ephemeris containing all the planets (except Mercury), the Moon, and the Earth-Moon barycenter from October 25, 1960 to about 2000 (except for the Moon, which has an ending date of 1970).

Direct use of the master merged ephemeris tape would, in general, waste computing time, since excess tape handling would occur in order to bypass data not required for the particular problem. To minimize tape handling during execution, a shorter merged ephemeris containing only that data needed for a specific problem is constructed at execution time. Several of these working ephemerides may be constructed before the integration of the problem. (Several problems may be loaded simultaneously with the same ephemeris, or each problem may require a distinct ephemeris, or several ephemerides may be desired for a single problem.)

Multistage Vehicles

The code is designed to handle the case of multistage vehicles in the following way: The stage data for all stages is intended to be loaded simultaneously. This results in several initial values of those parameters classified as stage parameters. Also, this may be expensive in terms of machine storage if either the number of stage parameters or number of stages is large. Therefore, only parameters of a basic group were defined to be arrays, and the number of stages was limited to 10. This group is composed of values for the initial mass, propellant flow rate, vacuum specific impulse, engine exit area, aerodynamic reference area, burning time, initial integration step size, and an input identification number. The input identification number is a provision that allows other parameters to be loaded just prior to integration of a particular stage.

Step-Size, Output, and Termination Controls

Truncation error and step size are controlled by computing the relative errors between the Runge-Kutta integration and the lower-order integration procedure. If the greatest relative error between the methods is greater than a maximum limit (ERLIMT), the integration step will be repeated after a smaller step size is computed. In either case, a new step size is computed from the relative errors of the previous steps and is intended to result in an error that is close to a reference value (EREF). Further, the step size may then be reduced by the output controls. In any case, a step can be no larger than three times the size of the previous successful step. (See appendix D.)

Output is sometimes desired at specific points along the trajectory, while at other times this is unimportant. This option is provided for the user so that he may choose output to occur at equal intervals in step number or equal time intervals (which places a constraint on the step size). Also, he may choose to change from one mode to another along the trajectory. These choices of output spacing are effected through the use of the FORTRAN variables MODOUT, DELMAX, STEPS, and TMIN, which is explained under the MODOUT entry of table II, a table of program control parameters.

In addition to the output control discussed in the previous paragraph, there is another facility that may be quite useful. The integration process may be interrupted at an arbitrary point along the trajectory where the point in question is not necessarily a specific time. For instance, it may be desirable to interrupt the flight at a specific altitude, velocity, dynamic pressure, and so forth. If the point is indeed attained along the path (it may not be), output occurs, input cards may be read in, and a decision is made whether to continue the stage, terminate the stage, or terminate the flight. The control of this facility is described under the entries in table II for LOOKX, XLOOK, LOOKSW, SWLOOK, and END.

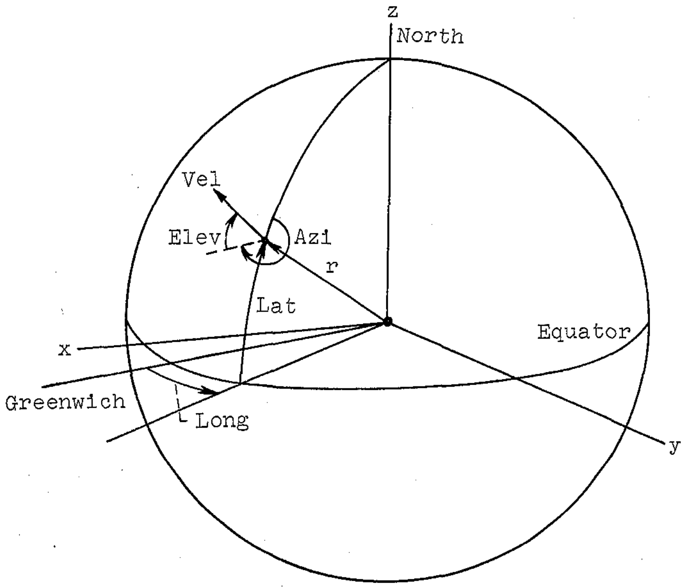
Computer Output

A basic output format was programed to serve as a basis for modification and is illustrated in table III. It is intended that a user of the code modify the output to suit his purpose. In addition to examining the normal output, it is sometimes desirable to examine the error-control data such as the relative errors in the integration variables along the path. These data are printed as a single block after completion of a stage if the sign of the input error reference value EREF is negative. The sign of EREF is irrelevant in the error-control portion of the program, since its absolute value is taken.

Computer Input

The user has a choice of three possible sets of input data that specify position and velocity: (1) six orbital elements, (2) three Cartesian components of velocity and position, and (3) latitude, longitude, azimuth, elevation, velocity, altitude, and time.

The third set mentioned is programed for the Earth only, where the latitude and longitude are the geocentric latitude and longitude measured from the equator and Greenwich, respectively. The azimuth angle is measured in a plane tangent to the sphere of radius r at the point on the sphere determined by the geocentric latitude and longitude, and relative to the local meridian, positive eastward from north. The elevation angle is then measured in a plane normal to the tangent plane, positive outward (sketch (b)). The tangent plane is taken to be horizontal with the effects of oblateness and rotation considered if these effects are "on." If oblateness and rotation are "off," the horizontal is perpendicular to the radial direction. This input option ignores the correction between universal time and ephemeris time and between the instantaneous equator and equinox and the mean equator and equinox of 1950.0.



Long = longitude measured from Greenwich in Earth's equatorial plane, positive east

Lat = latitude, measured positive north, geocentric

Azi = azimuth angle, measured east from north from local meridian

Elev = elevation angle, positive outward

Vel = vehicle's initial velocity

r = radius of vehicle from Earth's center

(b)

A list of input instructions is contained in appendix G along with an input check list.

The input routine described in reference 8 was used because of its simplicity; however, another input routine may be used if it is desired.

CODING

General

Appendix H contains the code listing of the program. A magnetic tape is available (from Lewis) so that the code listing in printed form and/or on cards can be reproduced. In addition, the tape contains the merged ephemeris data in proper format for use with the code.

Some of the FORMAT statements are of the G-type. These statements will print output in I, E, or F format depending on the nature of the variable. Fixed-point variables will take the I format, while floating-point variables will assume the F format unless the magnitude of the variable falls outside the useful F range, in which case the E format is used. FORTRAN facilities that do not accept the G-type format statements may easily substitute E-type formats.

A condensed description of the program supervisory control is shown in the flow diagram of figure 2. Table IV is a map of COMMON allocation (blanks are left for the user) and table II contains a description of the program control parameters. The elements of the integration variable array (XPRIM) are given in table V. The assumed values of the astronomical constants are given in table VI. These values are consistent with those given in reference 9.

Examples

Two examples of code usage are presented. The first example, which is described in the following paragraph, is a problem of raising a low-altitude satellite into a 24-hour orbit by using tangential, low acceleration. The other example is a more complex problem involving a ground-launched lunar probe with a three-stage rocket, which is described in appendix I. Both problems were selected to illustrate the usage of the program rather than to attempt a detailed analysis of the example problem.

For Example I for low-tangential thrust, the trajectory to be determined is that used to raise a 3850-kilogram package from an initial 300-statute-mile circular equatorial orbit to a 24-hour orbit using a 60,000-watt nuclear electric system with a specific impulse of 2540 seconds and an overall efficiency of 40 percent. The required engine parameters may be calculated as follows:

Thrust force:

$$F = \frac{2P_W\eta}{Ig_c} = \frac{2 \times 60,000 \times 0.4}{2540 \times 9.80665} = 1.927 \text{ newtons}$$

Initial acceleration:

$$\frac{F}{m_0} = \frac{1.927}{3850} = 5.0051948\times10^{-4} \text{ m/sec}^2$$

Propellant flow rate:

$$-\dot{m} = \frac{F}{Ig_c} = \frac{1.927}{2540 \times 9.80665} = 7.7361935\times10^{-5} \text{ kg/sec}$$

A detailed account is given in the following paragraphs for the solution of this problem by the prescribed program. Only those features of the program that have a direct bearing on this particular problem are discussed. Additional program features are discussed in the account of the second example problem. It may prove beneficial to refer to figure 2 during these two discussions. Also, all statement numbers referred to in the following text correspond to the program listing in appendix H.

It is assumed in the program that all memory data stores are cleared (set equal to zero) before operation begins. Control begins when the main program is entered. Then a set of so-called "standard data" is "initialized" by executing SUBROUTINE STDATA. Before initializing, STDATA clears that area of COMMON C no longer needed.

The next step is calling for input at statement 8. The following list of parameters constitutes the input:

Parameter	FORTRAN name	Value
Initial mass, m_0, kg	RMASS	3850
Semilatus rectum, p, m	P	6.86×10^6
Specific impulse, I, sec	SIMP	2540
Flow rate, $-\dot{m}$, kg/sec	FLOW	7.7361935×10^{-5}
Time limit, sec	TB	^{a}42590.2
Initial step size, sec	DELT	^{a}1500
Step number limit, steps	STEPMX	^{a}2000
Frequency of output, steps/output	STEPS	^{a}200

^{a}Assumed value.

Variables such as eccentricity and mean anomaly that are initially zero are not included in this list, since all memory data stores are initially zero.

In accordance with the input routine of reference 8, the input cards may appear as

```
$DATA=1,$TABLE,83=RMASS,718=P,103=SIMP,                $$  IDENTIFICATION AND
93=FLOW,143=DELT,73=TB,26=STEPMX,27=STEPS/             $$  TABLE DEFINITION
                                                       $$
RMASS=3850,SIMP=2540,FLOW=7.7361935E-5 $$  VEHICLE MASS, ISP, MASS FLOW
P=6.86E6,TB=42590.2,STEPMX=2000      $$  SEMILATUS-RECTUM, TIME LIMIT, STEP LIMIT
DELT=1500,STEPS=200                $$  INITIAL STEP SIZE, OUTPUT EVERY 200TH STEP
```

where the entries between the $TABLE and slash (/) reference the subsequent entries to the second argument C of the calling statement. Thus, for example, STEPS is equivalent to C(27), the 27th location from the beginning of COMMON C.

Part 11 of SUBROUTINE ORDER computes the gravitational constants μ and $\sqrt{\mu}$. Next, SUBROUTINE STAGE is called where the stage data for the first (and only) stage are moved into the proper stores for use in the SUBROUTINE NBODY. The vacuum value, PUSHO, for the thrust is computed, and then SUBROUTINE NBODY is called to integrate the path.

The next sequence is that of integrating the first two steps. These two steps are of equal size and are integrated before an error check is made. If the first two steps are satisfactory (determined by statement 25), the remaining steps are integrated while the relative error is being checked at the end of each step. Parts 1 and 5 of NBODY are concerned solely with this starting phase. Part 1 sets up the starting sequence and causes the initial conditions to appear on the output sheet. Parts 2 to 4 accomplish the Runge-Kutta integration for a single step.

The derivatives used in the integration are obtained from SUBROUTINE EQUATE. The first half of this subroutine finds the Cartesian coordinates and velocities through use of Kepler's equation. SUBROUTINE THRUST is called to determine the components of the thrust acceleration in the Cartesian coordinate system. (After control is returned to SUBROUTINE EQUATE, the thrust acceleration is resolved into circumferential, radial, and normal components.) Finally, the derivatives of the orbit elements are calculated, and a return is made to NBODY.

After the Runge-Kutta integration is performed, the error check is made in part 5B (part 6 after the starting sequence) by computing the difference between the Runge-Kutta integration and the low-order integration. SUBROUTINE ERRORZ is called to determine the largest of the relative errors. If the largest of the relative errors is greater than the limit value, ERLIMT (set in STDATA), part 8, which computes a smaller step size for the same interval, is entered and control is returned to part 1. If the greatest relative error is smaller than the limit value, part 7, which advances the variables of integration, is entered and calls SUBROUTINE STEP to compute the next step size and print out the variables of the first step. Part 7 also counts the revolutions past the x-axis and adjusts the argument of pericenter and mean anomaly to within $\pm\pi$ to retain accuracy in the sine-cosine routines. If the step size exceeds 1/2 revolution, the revolution count may be short by an integral number. Control is finally transferred to part 1 to begin computation of the next step.

The problem is terminated when the time limit TB is reached. This check is done in SUBROUTINE STEP. Had the problem exceeded the step number limit STEPMX, it would have terminated at that point. In either case, control is returned to the main program to begin the next problem. When no data for another problem are given, the execution is terminated (i.e., control is returned to the monitor by SUBROUTINE INPUT as a result of an end-of-file on tape 7). The output of the last step is

```
STEP=  822. + 46.   ECCENTRICITY= 2.37439758E-04 OMEGA= 1.57424484
TIME= 42590.200     SEMILATUS R.= 6898546.50      TRU A= 1.57107785
JDAY= 2440000.4927  MEAN ANOMALY= 1.57060298       NODE= 0
ALFA= 0               PATH ANGLE= 1.36042929E-02   INCL= 0

 V= 7601.36401      R= 6898546.94        REFER=EARTH  ORBIT  1
VX= 26.5485928      X=-6898498.81        RMASS= 3846.70511
VY=-7601.31769      Y=-25731.9050        REVS.= 7.50059360
VZ=-0               Z=-0                  DELT= 312.139160
```

The time histories of several trajectory parameters for this example are shown as solid lines in figure 3. The oscillations of the eccentricity and mean anomaly cause a rather small step size, as noted in the figure. To indicate how exercising care in selecting the input can increase the computational efficiency, the same problem may again be run with the following initial values (according to ref. 10) of eccentricity and mean anomaly:

$$e_0 = \frac{2(F/m_0)p^2}{\mu}$$

$$M_0 = \frac{\pi}{2} - 3e_0 - \frac{e_0 V_0}{2Ig_c}$$

The input cards for this case make use of the algebraic properties of the input routine to compute the desired value of these parameters. The cards are

```
$DATA=1,$TABLE,83=RMASS,718=P,103=SIMP,              $$  IDENTIFICATION AND
93=FLOW,143=DELT,73=TB,26=STEPMX,27=STEPS/           $$  TABLE DEFINITION
                                                     $$
RMASS=3850,SIMP=2540,FLOW=7.7361935E-5 $$  VEHICLE MASS, ISP, MASS FLOW
P=6.86E6,TB=42590.2,STEPMX=2000     $$  SEMILATUS-RECTUM, TIME LIMIT, STEP LIMIT
DELT=1500,STEPS=200               $$  INITIAL STEP SIZE, OUTPUT EVERY 200TH STEP
$TABLE,713=E,717=MA,714=OMEGA/ E=2*5.0051948E-4*P*P/3.983667E14$$ ECCENTRICITY
MA=-7620.429/SIMP/9.80665-6*E+3.1415926/2,STEPS=5 $$ MEAN ANOMALY,OUTPUT CONTROL
OMEGA=-2*E-MA  $$   ADJUST OMEGA (TO START PATH ON THE X-AXIS)
```

The dashed lines in figure 3 show the time histories of the same trajectory parameters when initial values of e and M given immediately preceding are used. The increase in average step size is 15 to 1. To compare the accuracy of this approximation with the exact case ($e_0 = M_0 = 0$), the final time was chosen when the corresponding orbit positions were identical (when the true anomalies were equal). At t = 42,590.2 seconds, the orbit positions are nearly identical, and, at this time, the values of position and velocity may be compared as follows:

	Case A: $e_0 = M_0 = 0$	Case B: e_0 and $M_0 \neq 0$
Radius, m	6898546.94	6898546.94
Velocity, m/sec	7601.36401	7601.36407
Number of steps	822	55

For most purposes the two answers would be accepted as equivalent, and case B would be preferred because less computer time is required.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, July 12, 1963

APPENDIX A

SYMBOLS

$\vec{A}$	relative angular momentum per unit mass, $\vec{r} \times \vec{V}'$ (appendix B)
A_e	engine exit area, m^2
$a_{i,j}$	coefficients for quadratic functions
C_D	total drag coefficient
$C_{D,i}$	induced drag coefficient
$C_{D,0}$	zero angle-of-attack drag coefficient
C_L	lift coefficient
D	drag force, newtons
$\mathcal{D}$	fourth harmonic coefficient in oblateness equations
E	eccentric anomaly, radians
e	eccentricity
F	thrust force, newtons
f_1,f_2	functions of Mach number
g_c	gravitational conversion factor, 9.80665 m/sec^2 (sometimes referred to as standard Earth gravity)
H	third harmonic coefficient in oblateness equations
h	altitude above Earth's surface, m
I	vacuum specific impulse, sec
i	orbit inclination to mean equator of 1950.0, radians
J	second harmonic coefficient in oblateness equations
k^2	universal gravitational constant, $1.32452139 \times 10^{20}$, $m^3/(sec^2)$(sun mass units)
L	lift force, newtons
M	mean anomaly, radians
m	object mass, kg

m_i	mass of i^{th} perturbating body, sun mass units
m_r	mass of reference body plus m, sun mass units
N_M	Mach number
P	atmospheric pressure, newtons/m^2
$\vec{P}$	$\vec{V}' \times \vec{A}$ (appendix B)
P_w	power, w
p	semilatus rectum, m
q	dynamic pressure, $\frac{1}{2}\rho(V')^2$, newtons/m^2
R_r	radius of reference body, m
r	radius from origin to object, m
r_i	radius from origin to i^{th} perturbating body, m
S	aerodynamic reference area, m^2
T	temperature, $^{\circ}K$
t	time, sec
U	gravitational potential
U_x,U_y,U_z	x,y,z accelerations due to gravity, m/sec^2
V	absolute velocity, m/sec
V'	relative velocity, m/sec
v	true anomaly, radians
X	forces acting on object other than gravity, thrust, lift, drag, and perturbations due to perturbating bodies
x,y,z	components of r, m
α	angle between thrust and velocity vectors (sketch (a)), deg
β	angle of rotation of thrust out of orbit plane (sketch (a)), deg
η	power efficiency factor
μ	k^2m_r
ρ	atmospheric density, kg/m^3

ω	argument of pericenter, radians
Ω	equatorial longitude of ascending node, radians

Subscript:

0	initial value

APPENDIX B

VECTOR RESOLUTION

Relative Velocity

The relative velocity is defined as the velocity of the object with respect to the origin body. If the origin body is assumed to rotate about the z-axis, this velocity is given by

$$\vec{V}' = \vec{V} - \vec{\omega} \times \vec{r} \tag{B1}$$

In x,y,z component form,

$$V'_x = V_x + \omega y \tag{B2a}$$

$$V'_y = V_y - \omega x \tag{B2b}$$

$$V'_z = V_z \tag{B2c}$$

In the following sections, the atmosphere of the origin body is assumed to rotate as a solid body at the rate $\vec{\omega}$.

Thrust Resolution Along x,y,z Axes

The thrust direction is specified with respect to the relative velocity vector $\vec{V}'$ by the angles α and β, as shown in sketch (a) (p. 4). For resolution of thrust vector into x,y,z components, it is convenient to define vectors $\vec{A}$ and $\vec{P}$ normal to and within the $\vec{r},\vec{V}'$ plane, respectively, such that $\vec{V}'$, $\vec{A}$, and $\vec{P}$ form an orthogonal set. Thus,

$$\vec{A} \equiv \vec{r} \times \vec{V}' = \text{relative angular momentum per unit mass} \tag{B3}$$

$$\vec{P} \equiv \vec{V}' \times \vec{A} \tag{B4}$$

The thrust vector can then be resolved in the $\vec{V}',\vec{A},\vec{P}$ set as:

$$\vec{F} \cdot \vec{V}' = FV' \cos \alpha \tag{B5a}$$

$$\vec{F} \cdot \vec{A} = FA \sin \alpha \sin \beta \tag{B5b}$$

$$\vec{F} \cdot \vec{P} = FP \sin \alpha \cos \beta \tag{B5c}$$

Solving for $\vec{F}$ yields

$$\vec{F} = \frac{F}{P^2} \left(V' \cos \alpha \, \vec{A} \times \vec{P} + A \sin \alpha \sin \beta \, \vec{P} \times \vec{V}' + P \sin \alpha \cos \beta \, \vec{P}\right) \tag{B6}$$

or, in x,y,z component form,

$$F_x = \frac{F}{P^2}\left[V' \cos \alpha(A_y P_z - A_z P_y) + A \sin \alpha \sin \beta(P_y V'_z - P_z V'_y) + P \sin \alpha \cos \beta\, P_x\right] \tag{B7a}$$

$$F_y = \frac{F}{P^2}\left[V' \cos \alpha(A_z P_x - A_x P_z) + A \sin \alpha \sin \beta(P_z V'_x - P_x V'_z) + P \sin \alpha \cos \beta\, P_y\right] \tag{B7b}$$

$$F_z = \frac{F}{P^2}\left[V' \cos \alpha(A_x P_y - A_y P_x) + A \sin \alpha \sin \beta(P_x V'_y - P_y V'_x) + P \sin \alpha \cos \beta\, P_z\right] \tag{B7c}$$

Aerodynamic Lift and Drag Resolution Along x,y,z Axes

The drag vector $\vec{D}$ is alined with the relative velocity vector $\vec{V}'$ and is therefore given in x,y,z components as

$$\vec{D} = -D\frac{V'_x}{V'} - D\frac{V'_y}{V'} - D\frac{V'_z}{V'} \tag{B8}$$

The lift vector $\vec{L}$ may be resolved into components along the previously defined orthogonal set $\vec{V}'$, $\vec{A}$, and $\vec{P}$ by the following relations:

$$\vec{L} \cdot \vec{V}' = 0 \tag{B9a}$$

$$\vec{L} \cdot \vec{A} = LA \sin \beta \tag{B9b}$$

$$\vec{L} \cdot \vec{P} = LP \cos \beta \tag{B9c}$$

Solving for $\vec{L}$ yields

$$\vec{L} = \frac{L}{P^2}(A \sin \beta\, \vec{P} \times \vec{V}' + P \cos \beta\, \vec{P}) \tag{B10}$$

or, in x,y,z component form,

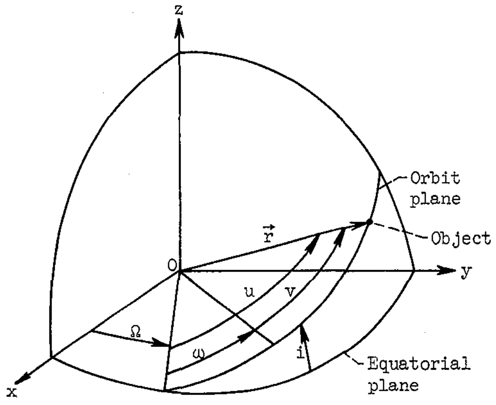
$$L_x = \frac{L}{P^2}\left[A \sin \beta(P_y V'_z - P_z V'_y) + P \cos \beta\, P_x\right] \tag{B11a}$$

$$L_y = \frac{L}{P^2}\left[A \sin \beta(P_z V'_x - P_x V'_z) + P \cos \beta\, P_y\right] \tag{B11b}$$

$$L_z = \frac{L}{P^2}\left[A \sin \beta(P_x V'_y - P_y V'_x) + P \cos \beta\, P_z\right] \tag{B11c}$$

APPENDIX C

TRANSFORMATION EQUATIONS FROM ORBIT ELEMENTS TO RECTANGULAR COORDINATES



(c)

From spherical trigonometry used in reference to the celestial sphere shown in sketch (c), the following relations may be derived for the position coordinates:

$$x = r(\cos \Omega \cos u - \sin \Omega \sin u \cos i) \tag{C1a}$$

$$y = r(\sin \Omega \cos u + \cos \Omega \sin u \cos i) \tag{C1b}$$

$$z = r(\sin u \sin i) \tag{C1c}$$

where

$$r = \frac{p}{1 + e \cos v} \tag{C2a}$$

$$u = \omega + v \tag{C2b}$$

and v can be obtained from

$$\cos v = \frac{\cos E - e}{1 - e \cos E} \tag{C2c}$$

and

$$M = E - e \sin E \tag{C2d}$$

The velocity components may be obtained by differentiating the position equations using the two-body relations $\dot{u} = \dot{v} = \frac{\sqrt{\mu p}}{r^2}$ and $\dot{r} = \sqrt{\frac{\mu}{p}}\, e \sin v$:

$$\dot{x} = -\sqrt{\frac{\mu}{p}}\,(N \cos i \sin \Omega + Q \cos \Omega) \qquad \text{(C3a)}$$

$$\dot{y} = \sqrt{\frac{\mu}{p}}\,(N \cos i \cos \Omega - Q \sin \Omega) \qquad \text{(C3b)}$$

$$\dot{z} = \sqrt{\frac{\mu}{p}}\,(N \sin i) \qquad \text{(C3c)}$$

where

$$N = e \cos \omega + \cos u \qquad \text{(C4a)}$$

$$Q = e \sin \omega + \sin u \qquad \text{(C4b)}$$

APPENDIX D

RUNGE-KUTTA AND LOW-ORDER INTEGRATION SCHEMES WITH ERROR CONTROL

The Runge-Kutta formula used is of fourth-order accuracy in step size h. It is of the form

$$X\Big]_1^2 \equiv X_2 - X_1 = \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{D1}$$

where

$$X = \text{a dependent variable}$$

$$X\Big]_1^2 = \text{increment in the dependent variable}$$

$$h_2 = \text{increment in the independent variable } t$$

$$k_1 = h_2\dot{X}_2(t_1, X_1)$$

$$k_2 = h_2\dot{X}_2\left(t_1 + \frac{h_2}{2},\ X_1 + \frac{k_1}{2}\right)$$

$$k_3 = h_2\dot{X}_2\left(t_1 + \frac{h_2}{2},\ X_1 + \frac{k_2}{2}\right)$$

$$k_4 = h_2\dot{X}_2(t_1 + h_2,\ X_1 + k_3)$$

A lower-order formula may be found by utilizing the three derivatives at $t = t_0$, t_1, and t_2. If $h_1 = t_1 - t_0$ and $h_2 = t_2 - t_1$, the following Lagrangian interpolation formula gives the derivative at any time $t_0 \leq t \leq t_2$:

$$\dot{X} \equiv \dot{X}_0 \frac{(t - t_1)(t - t_2)}{h_1(h_1 + h_2)} - \dot{X}_1 \frac{(t - t_0)(t - t_2)}{h_1 h_2} + \dot{X}_2 \frac{(t - t_0)(t - t_1)}{h_2(h_1 + h_2)} \tag{D2}$$

Integration of this equation from t_1 to t_2 yields

$$X'\Big]_1^2 = \frac{1}{6}\left[\left(\frac{h_2}{h_1}\right)^2\left(\frac{-h_2}{1 + \frac{h_2}{h_1}}\right)\dot{X}_0 + \frac{h_2}{h_1}\,(h_2 + 3h_1)\dot{X}_1 + \left(2h_2 + \frac{h_2}{1 + \frac{h_2}{h_1}}\right)\dot{X}_2\right] \tag{D3}$$

The difference in the increments over the interval h_2 between the Runge-Kutta scheme and the low-order scheme may be divided by a nominal value of the dependent variable $\overline{X}$ to obtain the relative error δ_2. Thus,

$$\delta_2 = \left| \frac{X'\Big]_1^2 - X\Big]_1^2}{\overline{X}} \right| \tag{D4}$$

The error is expected to vary as approximately the fifth power of h, which leads to

$$\delta = Ah^5 \tag{D5a}$$

(where A is a suitable coefficient) or in the logarithmic form

$$\log \delta = A' + 5 \log h \tag{D5b}$$

where

$$A' = \log A \tag{D6a}$$

Let it be assumed that A' will vary linearly with t, the variable of integration. Then A' at a time corresponding to t_3 can be found from A' at two previous points t_1 and t_2 as

$$A_3' = A_2' + \frac{A_2' - A_1'}{t_2 - t_1} (t_3 - t_2) \tag{D6b}$$

and if $h_3 = (t_3 - t_2)$ and $h_2 = (t_2 - t_1)$,

$$A_3' = A_2' + (A_2' - A_1') \frac{h_3}{h_2} \tag{D6c}$$

and on this basis δ_3 would be predicted to be

$$\log \delta_3 = A_3' + 5 \log h_3 \tag{D7}$$

It is desired that δ_3 should approximate $\overline{\delta}$, the reference error; therefore,

$$\log h_3 = \frac{1}{5} (\log \overline{\delta} - A_3') \tag{D8}$$

Each dependent variable has an associated relative error and would lead to computation of a different step size for each variable; however, the maximum relative error of all variables may be selected for δ. Obviously, inaccurate predictions of step size can occur when the maximum relative error shifts from one variable to another or when any sudden change occurs. When a step size produces an excessively large error ($\delta > \delta_{limit}$), a reduced step size must be used. It may be obtained from the reference error $\bar{\delta}$ as

$$h_3 = \exp\left[\frac{1}{5}\left(\log \bar{\delta} - A_2^1\right)\right] \tag{D9}$$

Starting the integration. - The Runge-Kutta scheme is simple to start, since integration from X_n to X_{n+1} requires no knowledge of X less than X_n. Since the error control coefficient A has no value at $t = 0$, a prediction of the second step size is difficult. To overcome this difficulty, two equal size first steps may be made before checking the error. The A for the first step may be arbitrarily set equal to the A for the second step so that h_3 may be predicted. The low-order integration scheme equation in this case becomes, with $h_2 = h_1$,

$$X'\Big]_1^2 = \frac{h_1}{3}\left(\dot{X}_0 + 4\dot{X}_1 + \dot{X}_2\right) \tag{D10}$$

Failures. - Should two consecutive predictions of the same step fail to produce an error δ less than δ_{limit}, a return to the starting procedure will be made with a third prediction on step size, which is no larger than one-half of the second estimate. The step-size control described here will operate stably with nearly constant error per step only for a well-behaved function. For most problems it will repeat a step occasionally to reduce a large error, and on sharp corners it will restart. This action is not regarded as objectionable. The objective is to attain a desired level of accuracy with a minimum total number of steps.

APPENDIX E

GLOSSARY OF VARIABLES

VARIABLE	COMMON LOCATION	DEFINITION
A (700)	C(11)	ARRAY CONTAINING THE INITIAL DATA AND THE PROGRAM CONTROL VARIABLES
A1	B(10)	ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(1)
A2	B(11)	ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(2)
ACOEF1	B(12)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
ACOEF2	B(13)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
ACOEF3	B(14)	INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)
AEXIT1 (10)	A(103)	ENGINE EXIT AREAS FOR AT MOST 10 STAGES, M**2
AEXIT	B(3)	AEXIT1(NSTAGE)
AK (3)	A(51)	RUNGE KUTTA COEFFICIENTS,SET IN STDATA
ALPHA	A(49)	ANGLE BETWEEN VELOCITY AND THRUST VECTORS,SEE SKETCH (A)
ALT	A(4)	VEHICLE ALTITUDE ABOVE EARTH, M
AM	B(90)	TOTAL VEHICLE ANGULAR MOMENTUM PER UNIT MASS, M**2/SEC
AMASS (30)	A(347)	PERMANENT LIST OF BODY MASSES IN ORDER OF PNAME LIST, SET IN STDATA,MASSES FROM ELIPS DATA BEGIN AT AMASS (21), SUN MASS UNITS
AMC (3)	B(87)	X,Y,Z COMPONENTS OF ANGULAR MOMENTUM PER UNIT MASS, **2)/SEC
AMSQRD	B(91)	SQUARE OF TOTAL ANGULAR MOMENTUM PER UNIT MASS,M**4/SEC**2
AREA1 (10)	A(113)	AERODYNAMIC REFERENCE AREAS FOR AT MOST 10 STAGES, M**2
AREA	B(6)	AREA1(NSTAGE)
ASYMPT	A(7)	SEE TABLE II
ATMN	A(21)	SEE TABLE II
AU	A(29)	ASTRONOMICAL UNIT,M
AW (4)	A(55)	RUNGE KUTTA COEFFICIENTS,SET IN STDATA
AZI	A(35)	INITIAL AZIMUTH ANGLE,USED WHEN IMODE = 4, SEE SKETCH (B),DEGREES

B (800)	C(1111)	ARRAY CONTAINING INTERNAL PARAMETERS NOT UNDER USER CONTROL
BETA	A(50)	ANGLE BETWEEN VELOCITY-THRUST PLANE AND ORBIT PLANE, SEE SKETCH(A)
BMASS (8)	B(137)	BODY MASSES SELECTED FROM AMASS LIST IN SEQUENCE CORRESPONDING TO BNAME LIST
BNAME (8)	B(122)	ORDERED LIST OF BCD BODY NAMES
BODYCD (10)	A(143)	ORIGINAL UNORDERED LIST OF BCD BODY NAMES READ IN AT INPUT
BODYL (10)	B(153)	AUXILIARY ORDERED LIST OF BCD BODY NAMES
CD	A(165)	TOTAL DRAG COEFFICIENT
CDI	A(163)	INDUCED DRAG COEFFICIENT
CHAMP	B(25)	SMALLEST CRITICAL RADIUS (RBCRIT(J)) WITHIN WHICH OBJECT LIES
CINCL	B(55)	COSINE OF INCLINATION
CIRCUM	B(82)	CIRCUMFERENTIAL COMPONENT OF TOTAL PERTURBATIVE ACCELERATION,M/SEC**2
CL	A(164)	LIFT COEFFICIENT
CLEAR	C(3)	SEE TABLE II
COEFN (192)	A(407)	STORAGE ARRAY FOR COEFFICIENTS USED TO COMPUTE ALPHA, CL,CDI,CD OR OTHER PARAMETERS
COMPA (3)	B(63)	COMPONENTS OF TOTAL PERTURBATIVE ACCELERATION ALONG X ,Y,Z AXES.
CONSU	A(31)	SEE TABLE II
CONSTU	A(32)	SEE TABLE II
COSALF	B(48)	COSINE OF ALPHA
COSBET	B(49)	COSINE OF BETA
COSTRU	B(53)	COSINE OF TRU
COSV	B(57)	COSINE OF THE ARGUMENT OF LATITUDE
D (1100)	C(2111)	ARRAY WHERE SAVED DATA IS STORED FOR LATER USE. ARRAYS A,XPRIM,AND XPRIMB MAY BE SAVED.
DELMAX	A(19)	SEE TABLE II
DEL	A(43)	OUTPUT CONTROL PARAMETER USED IN STEP
DELT1 (10)	A(133)	INITIAL STEP SIZES FOR AT MOST 10 STAGES,SEC
DELT	B(1)	DELT1(NSTAGE)
DNSITY	B(29)	ATMOSPHERIC DENSITY,KG/M**3
DONE	B(39)	CONTROL PARAMETER FROM STEP WHICH INFORMS NBODY TO STOP INTEGRATING

DRAG (3)	B(69)	X,Y,Z COMPONENTS OF THE DRAG ACCELERATION,M/SEC**2
DTOFFJ	A(23)	JULIAN DATE OF TAKEOFF
E2	B(18)	LARGEST OF THE RELATIVE ERRORS BETWEEN R-K AND LOW-ORDER INTEGRATION METHODS,EQ. (D4)
EFMRS (7)	B(130)	LIST OF BCD BODY NAMES WHOSE POSITIONS ARE TO BE DETERMINED FROM TAPE DATA
ELEV	A(36)	INITIAL ELEVATION ANGLE,USED WHEN IMODE=4, SKETCH(B), DEGREES
ELIPS (12,10)	A(167)	ELLIPSE DATA FOR PERTURBATING BODIES,READ FROM CARDS, 12 PIECES OF DATA PER BODY
EMONE	B(28)	ECCENTRICITY -1
END	A(5)	SEE TABLE II
EPAR	B(26)	SQUARE ROOT OF (ECCENTRICITY SQUARED -1)
EREF	A(13)	SEE TABLE II
ERLIMT	A(14)	SEE TABLE II
ERLOG	B(17)	NATURAL LOGARITHM OF EREF
ETOL	A(30)	SEE TABLE II
EXITA	B(392)	AEXIT(NSTAGE)/100, NEWTONS/MB
EXMODE	B(27)	ECCENTRICITY CALCULATED WHEN IMODE=3
FILE	B(22)	SEE TABLE II
FLOW1 (10)	A(83)	RATE OF PROPELLENT FLOW, KG/SEC
FLOW	B(5)	FLOW1(NSTAGE)
FORCE (3)	B(66)	X,Y,Z COMPONENTS OF THRUST ACCELERATION, M/SEC**2
GASFAC	A(46)	DEFINED IN SUBROUTINE AERO, SET IN STDATA
GEOH	B(32)	GEOPOTENTIAL, M
GK2M	B(36)	GRAVITATIONAL CONSTANT,MU,OF THE SYSTEM,M**3/SEC**2
GKM	B(37)	SQUARE ROOT OF GK2M
H2	B(15)	VALUE OF DELT FOR PREVIOUS STEP
IBODY (8)	B(177)	DEFINED IN SUBROUTINE ORDER
ICC (10)	A(153)	SEE TABLE II
IDENT (10)	A(123)	INPUT IDENTIFICATION NUMBERS ASSOCIATED WITH EACH STAGE
IMODE	A(1)	SEE TABLE II
IND (3)	A(60)	SET OF INDICES, SET IN STDATA
INDERR	B(51)	NUMBER OF SETS OF ERROR DATA, SET IN ERRORZ FOR USE IN NBODY

INLOOK	A(599)	INPUT IDENTIFICATION NUMBER FOR INPUT AFTER FINDING C (LOOKX) = XLOOK
KSUB	B(19)	INDEX OF RUNGE-KUTTA SUBINTERVALS
LAT	A(33)	INITIAL GEOCENTRIC LATITUDE, USED WHEN IMODE=4,SKETCH (B), DEGREES
LONG	A(34)	INITIAL LONGITUDE RELATIVE TO GREENWICH, USED WHEN IMODE=4, SKETCH(B), DEGREES
LOOKX	A(8)	SEE TABLE II
LOOKSW	A(9)	SEE TABLE II
LSTAGE	A(38)	TOTAL NUMBER OF STAGES INTEGRATED BEFORE RETURNING TO THE MAIN PROGRAM
MBODYS	B(42)	NUMBER OF PERTURBATING BODIES (NBODYS-1)
MODOUT	A(20)	SEE TABLE II
NBODYS	B(41)	TOTAL NUMBER OF BODIES, EXCLUDING THE VEHICLE
NCASES	A(600)	SAVED VALUE OF NCASE
NCASE	C(1)	CASE NUMBER, RAISED ONCE EACH TIME CONTROL PASSES THROUGH THE MAIN PROGRAM
NEFMRS (8)	B(185)	DEFINED IN SUBROUTINE ORDER
NEQ	A(2)	NUMBER OF EQUATIONS TO BE INTEGRATED, SET TO 8 IN STDATA
NSAVE	C(4)	SEE TABLE II
NSTAGE	A(3)	THE INDEX INDICATING THE PARTICULAR STAGE CURRENTLY BEING INTEGRATED
NSTART	B(24)	INTERNAL CONTROL IN NBODY AND EQUATE
OBLATJ	A(26)	OBLATENESS COEFFICIENT OF SECOND HARMONIC
OBLATD	A(27)	OBLATENESS COEFFICIENT OF FOURTH HARMONIC
OBLATH	A(28)	OBLATENESS COEFFICIENT OF THIRD HARMONIC
OBLATN	A(40)	SEE TABLE II
OBLAT (3)	B(75)	X,Y,Z COMPONENTS OF OBLATENESS ACCELERATION, M/SEC**2
OLDDEL	B(9)	VALUE OF DELT FOR PREVIOUS GOOD STEP
ORBELS (6)	B(116)	ARRAY OF OUTPUT VARIABLES,EITHER RECTANGULAR OR ORBIT ELEMENTS
OUTPOT	B(399)	CAUSES ABSENCE OF OUTPUT WHEN NONZERO
P (3)	B(84)	DEFINED IN EQ. (B4)

PAR (3)	B(60)	DEFINED BY EQUATIONS IN SUBROUTINE THRUST
PMAGN	B(50)	DEFINED IN EQUATION FORM BY SUBROUTINE THRUST
PNAME (30)	A(287)	PERMANENT LIST OF BODY NAMES MADE FROM PNAA LIST IN SUBROUTINE ORDER, ELIPS NAMES BEGIN AT PNAME(21)
PRESS	B(33)	ATMOSPHERIC PRESSURE, MB
PSI	B(30)	PATH ANGLE, ANGLE BETWEEN PATH AND LOCAL HORIZONTAL, DEGREES
PSIR	B(398)	RELATIVE PATH ANGLE, TAKEN RELATIVE TO A ROTATING ORIGIN BODY, DEG
PUSH	A(166)	THRUST FORCE, NEWTONS
PUSHO	B(391)	VACUUM THRUST FORCE, NEWTONS
Q	B(59)	DYNAMIC PRESSURE, NEWTONS/M**2
QMAX	B(44)	MAXIMUM VALUE OF Q DEVELOPED DURING A SINGLE TRAJECTORY (SET TO ZERO WHEN CONTROL PASSES THROUGH SUBROUTINE EXTRA)
QX (3)	B(78)	X,Y,Z COMPONENTS OF PERTURBATIVE ACCELERATION DUE TO PERTURBATING BODIES, M/SEC**2
R (8)	B(102)	DISTANCES OF ALL BODIES FROM OBJECT, IN ORDER OF BNAME LIST, M
RADIAL	B(81)	RADIAL COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, POSITIVE OUTWARD, M/SEC**2
RAMC (5)	B(393)	RELATIVE ANGULAR MOMENTUM PER UNIT MASS COMPONENTS, TOTAL RELATIVE ANGULAR MOMENTUM PER UNIT MASS, AND ITS SQUARE, M**2/SEC
RATM	A(22)	RADIUS OF ATMOSPHERE,M
RATMOS	B(23)	SET EQUAL TO RATM WHEN ATMN EQUALS THE REFERENCE BODY NAME, BNAME(1)
RATIO	B(58)	RATIO OF ADJACENT STEP SIZES, DELT
RB (3,8)	B(193)	X,Y,Z COMPONENTS OF DISTANCE FROM ALL BODIES TO THE OBJECT,M
RBCRIT (8)	B(145)	LIST OF SPHERE-OF-INFLUENCE RADII OF ALL BODIES IN BNAME LIST, M
RCRIT (30)	A(377)	PERMANENT LIST OF SPHERE-OF-INFLUENCE RADII CORRESPONDING TO PNAME LIST OF BODY NAMES. RADII FROM ELIPS DATA BEGIN AT RCRIT(21), M
RE	A(25)	RADIUS OF EARTH EQUATOR, M
RECALL	C(5)	SEE TABLE II
REFER (30)	A(317)	LIST OF REFERENCE BODIES CORRESPONDING TO PNAME LIST, REFERENCE BODIES FROM ELIPS DATA BEGIN AT REFER(21)
RESQRD	B(7)	SQUARE OF RE
RETURN	B(400)	CAUSES CONTROL NOT TO RETURN TO MAIN PROG. IF NONZERO

REVS	A(48)	REVOLUTION COUNTER, USED ONLY FOR OUTPUT
REVOLV	B(21)	ROTATION RATE OF REFERENCE BODY WHEN ATMN=BNAME(1), RAD/SEC
RMASS1 (10)	A(73)	INITIAL MASSES FOR AT MOST 10 STAGES,KG
ROTATE	A(39)	ROTATION RATE OF A REFERENCE BODY, RAD/SEC
RSQRD	B(45)	RADIUS SQUARED OF OBJECT TO ORIGIN, M**2
SIGNAL	B(31)	SEE TABLE II
SIMP1 (10)	A(93)	SPECIFIC IMPULSES FOR AT MOST 10 STAGES, SEC
SIMP	B(2)	SIMP1(NSTAGE)
SINALF	B(46)	SINE OF ALPHA
SINBET	B(47)	SINE OF BETA
SINTRU	B(52)	SINE OF TRU
SINCL	B(54)	SINE OF INCLINATION
SINV	B(56)	SINE OF THE ARGUMENT OF LATITUDE
SPACES	B(16)	NUMBER OF EQUAL TIME UNITS UNTIL NEXT OUTPUT
SPD	A(44)	SECONDS PER DAY, SET IN STDATA, SEC/DAY
SQRDK1	A(47)	GRAVITATIONAL CONSTANT OF THE SUN, AU**3/DAY**2
SQRDK	B(35)	GRAVITATIONAL CONSTANT OF THE SUN, M**3/SEC**2
STEPMX	A(16)	SEE TABLE II
STEPS	A(17)	SEE TABLE II
STEPGO	A(41)	COUNT OF SUCCESSFUL INTEGRATION STEPS
STEPNO	A(42)	COUNT OF UNSUCCESSFUL INTEGRATION STEPS (THOSE WHICH DO NOT PASS ERROR CONTROL TEST)
SWLOOK	A(10)	SEE TABLE II
TABLT	B(20)	TIME MEASURED RELATIVE TO THE JULIAN DATE OF TAKEOFF, DAYS
TABLE (200)	C(1911)	ARRAY OF INPUT PARAMETERS AND THEIR COMMON STORE LOCATIONS
TAPE3	C(2)	SEE TABLE II
TB (10)	A(63)	FLIGHT TIMES FOR AT MOST 10 STAGES, SEC

TDATA (6,3,7)	B(265)	COEFFICIENTS FROM EPHEMERIDES TAPE TO BE USED IN DETERMINING POSITIONS AND POSSIBLY VELOCITIES OF PERTURBATING BODIES, ONE SET FOR EACH OF 7 BODIES
TDEL (7)	B(170)	ONE-HALF OF TIME SPACING BETWEEN TWO ADJACENT ENTRIES OF LIKE BODY NAME ON EPHEMERIDES TAPE, READ FROM TAPE FOR EACH BODY
TFILE	A(6)	SEE TABLE II
TIM (7)	B(163)	TIME FOR SET OF EPHEMERIS DATA, READ FROM EPHEMERIDES TAPE, ONE FOR EACH BODY
TKICK	A(15)	INITIAL STEP SIZE OF A TRAJECTORY TO BE COMPUTED IN CLOSED-FORM, FOR USE WHEN IMODE=4, WHICH FACILITATES STARTING OF SOME TYPES OF TRAJECTORIES
TM	B(34)	ATMOSPHERIC TEMPERATURE TIMES THE RATIO OF MOLECULAR TO ACTUAL MOLECULAR WEIGHT, DEGREES KELVIN
TMAX	B(4)	SEE TABLE II
TMIN	A(18)	SEE TABLE II
TOFFT	A(24)	FRACTIONAL PART OF JULIAN DATE OF TAKEOFF, DAYS
TRSFER	B(8)	SEE TABLE II
TRU	B(40)	TRUE ANOMALY, RAD
TTEST	A(54)	SEE TABLE II
TTOL	A(45)	TIME TOLERANCE WITHIN WHICH PROBLEM TIME MINUS TMAX MUST LIE TO END STAGE
U	A(59)	ECCENTRIC ANOMALY, RAD
V	B(95)	VELOCITY OF OBJECT RELATIVE TO THE ORIGIN, M/SEC
VATM (3)	B(97)	X,Y,Z COMPONENTS OF THE RELATIVE VELOCITY, VQ,M/SEC
VEFM (3,8)	B(241)	X,Y,Z COMPONENTS OF OBJECT VELOCITY RELATIVE TO ALL BODIES, M/SEC
VEL	A(37)	INITIAL RELATIVE VELOCITY, USED WHEN IMODE=4, SKETCH (B), M/SEC
VMACH	B(38)	MACH NUMBER OF OBJECT
VQ	B(100)	VELOCITY OF OBJECT RELATIVE TO ATMOSPHERE, M/SEC
VQSQRD	B(101)	SQUARE OF VQ, M**2/SEC**2
VSQRD	B(96)	SQUARE OF V, M**2/SEC**2
VX	B(92)	X COMPONENT OF VELOCITY, M/SEC
VY	B(93)	Y COMPONENT OF VELOCITY, M/SEC
VZ	B(94)	Z COMPONENT OF VELOCITY, M/SEC
X (100)	B(401)	WORKING SET OF INTEGRATION VARIABLES
XDOT (100)	B(501)	TIME DERIVATIVES OF THE SET X

XIFT (3)	B(72)	X,Y,Z COMPONENTS OF LIFT ACCELERATION, M/SEC**2
XINC (100)	B(601)	INCREMENTS OF THE INTEGRATION VARIABLES PER STEP
XLOOK	A(12)	SEE TABLE II
XP (3,8)	B(217)	X,Y,Z COMPONENTS OF PERTURBATING BODY POSITIONS RELATIVE TO ORIGIN
XPRIM (100,2)	C(711)	TWO 100-ELEMENT SETS, THE FIRST SET CONTAINS VALUES OF THE INTEGRATION VARIABLES AT THE PREVIOUS GOOD STEP,THE SECOND SET IS UNDER THE INTEGRATION PROCESS, SEE TABLE V
XPRIMB (100,2)	C(911)	LEAST SIGNIFICANT HALF OF DOUBLE PRECISION INTEGRATION VARIABLES XPRIM
XTOL	A(11)	TOLERANCE ON THE DISCIMINATION C(LOOKX)-XLOOK TO BE SATISFIED
XWHOLE (6)	B(110)	RECTANGULAR COORDINATES AND VELOCITIES, SET ASIDE FOR USE IN ORIGIN TRANSLATIONS
ZN	B(43)	MEAN ANGULAR MOTION OF OBJECT, RAD/SEC
ZORMAL	B(83)	Z COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, M/SEC**2

APPENDIX F

LEWIS RESEARCH CENTER EPHEMERIS

General Description

The ephemeris data initially available on magnetic tape were from the Themis code prepared by the Livermore Laboratory, evidently from U.S. Naval Observatory data. Later, an ephemeris was obtained from the Jet Propulsion Laboratory assembled as a joint project of the Jet Propulsion Laboratory and the Space Technology Laboratory. These data are given relative to the mean vernal equinox and equator of 1950.0 and are tabulated with ephemeris time as the argument.

An ephemeris was desired for certain uses in connection with the IBM 7090 computer that would be shorter than the original ephemeris tapes mentioned and would be as accurate as possible consistent with the length. A short investigation of the various possibilities led to adoption of fitted equations. In particular, fifth-order polynomials were simultaneously fitted to the position and velocities of a body at three points. This procedure provides continuity of position and velocity from one fit to the next, because the exterior points are common to adjacent fits. Polynomials were selected rather than another type of function, because they are easy to evaluate. Three separate polynomials are used for the x, y, and z coordinates, respectively.

Procedure Used to Fit Data

The process of computing the fitting equations is as follows:

(1) A group of 50 sets of the components of planetary position was read into the machine memory for a single planet together with differences as they existed on the original magnetic tape. The differences were verified by computation (in double precision because some data required it); and any errors were investigated, corrected, and verified. Published ephemeris data were adequate to correct all errors found.

(2) The components of velocity v_x, v_y, and v_z were computed and stored in the memory for each of the 50 positions by means of a numerical differentiation formula using ninth differences; namely,

$$\dot{X} = (T_1 - T_{-1})\left[\frac{\Delta I_{-1} + \Delta I_{+1}}{2} - \frac{\Delta III_{-1} + \Delta III_{+1}}{12} + \frac{\Delta V_{-1} + \Delta V_{+1}}{60} - \frac{\Delta VII_{-1} + \Delta VII_{+1}}{280} + \frac{\Delta IX_{-1} + \Delta IX_{+1}}{1260}\right] \tag{F1}$$

(See ref. 11, pp. 42 and 99 for notation.) Double-precision arithmetic was used for differences, but velocities were tabulated with single precision.

(3) Coefficients C, D, E, and F in the fifth-order polynomial

$$X = X_0 + \dot{X}_0(T - T_0) + C(T - T_0)^2 + D(T - T_0)^3 + E(T - T_0)^4 + F(T - T_0)^5 \qquad \text{(F2)}$$

and its derivative

$$\dot{X} = \dot{X}_0 + 2C(T - T_0) + 3D(T - T_0)^2 + 4E(T - T_0)^3 + 5F(T - T_0)^4 \qquad \text{(F3)}$$

were found to fit a first point (which was far enough from the beginning point to have all differences computed) and two equally spaced points for each component of position and velocity. (The initial spacing is not important, as will be seen later.) Spacing is defined as the number of original data points fitted by one equation. Single-precision arithmetic was used.

(4) The coefficients C, D, E, and F in step (3) were then used in equations (F2) and (F3) to calculate components of all positions and velocities given in the original data and lying within the interval fitted. These values were checked with the original data. Radius R and velocity V were computed at the times tabulated in the original data. If any component of the position differed from the original data by more than $R \times 10^{-7}$ or if any velocity differed from the original by more than $V \times 10^{-6}$, the fit was considered unsatisfactory.

(5) If the fit was considered unsatisfactory, this fact was recorded, and the spacing was reduced by two data points. Steps 2 to 4 were then repeated. If the fit was considered satisfactory, this fact was recorded, and the spacing was increased by two spaces. Steps 2 to 4 were repeated. The largest satisfactory fit was identified when a certain spacing was satisfactory and the next larger fit was not satisfactory.

(6) The coefficients that corresponded to the largest satisfactory fit were recorded on tape, in binary mode as follows:

Word number	Data	Mode	Definitions and/or units
1	Planet name	BCD	Six characters (first six)
2	Julian date	Floating point	Date of midpoint of fit, Julian date
3	Delta T	↓	Number of days on each side of midpoint
4	F_x		[a]AU/day^5
5	E_x		[a]AU/day^4
6	D_x		[a]AU/day^3
7	C_x		[a]AU/day^2
8	$\dot{x}$		[a]AU/day
9	x		[a]AU
10	F_y		[a]AU/day^5
11	E_y		[a]AU/day^4
12	D_y		[a]AU/day^3
13	C_y		[a]AU/day^2
14	$\dot{y}$		[a]AU/day
15	y		[a]AU
16	F_z		[a]AU/day^5
17	E_z		[a]AU/day^4
18	D_z		[a]AU/day^3
19	C_z		[a]AU/day^2
20	$\dot{z}$		[a]AU/day
21	z		[a]AU

[a] Except for Moon data, which are in Earth radii and days.

(7) As soon as a set of coefficients was selected for an interval, additional data were read from the source ephemeris tape and used to replace the points already fitted (except the last point). These data were processed as described in steps 1 and 2 so that the next 50 points were ready to be fitted. Steps 3 to 6 were then used to find the next set of coefficients, and steps 1 to 6 were repeated until all data for all planets were fitted.

Data Treated

The preceding process was applied to all data available at the time. For the Moon, the technique usually led to the use of every point in the fitted interval (i.e., only three points were fitted). Thus, a check of accuracy was not available. The error in the attempt to fit the next greater interval (five points) was not excessive, however, and it is judged that the accuracy obtained from these fits is about equal to that held on the other bodies.

Merged Ephemeris Tape

Once all the positions and velocities of all the bodies then available were fitted, the coefficients were merged in order of the starting date of each fit. The resulting tape was written in binary mode with 12 sets of fits per record.

The detail of this record is as follows:

1st word: FORTRAN compatible
2nd word: file number, fixed point in decrement

Set 1
3rd word: planet name, code in BCD, first six characters
4th word: Julian date, floating point
- etc., according to list in paragraph 6
- 21 words
23rd word: z

Set 2
24th word: planet name, code in BCD, first six characters
25th word: Julian date, floating point
-
-
44th word: z

Successive sets follow one another with a total of 12 sets.

Set 12 (last set)
234th word: planet name
235th word: Julian date, floating point
254th word: z
End-of-record gap

One record contains 254 words, the first is for FORTRAN compatibility, the second is a file number used for identification in the system. It is a fixed point 2. The third is the beginning of the first set of data, and 12 sets follow each with 21 words. The last word is the 254th word (counting the FORTRAN compatible word) followed by an end-of-record gap. The remaining records are compiled in the same manner with an end-of-file recorded as a terminating mark.

Because of the merging operation, all bodies are given in one list in a random order according to the starting date of the interval. The starting date is the Julian day (word 2) minus the half interval (word 3) (see procedure, paragraph 6). The entire ephemeris occupies about one-seventh reel of tape. A summary of data is given in table VII.

APPENDIX G

INPUT-DATA REQUIREMENTS

The procedure needed to run actual problems with the aid of this routine is described herein. It is intended to permit the user with a specific problem in mind to make a complete list of data required and to select desirable operating alternatives from those available. The details of this procedure are contained in the following instructions:

(1) Provision has been made for two types of ephemeris data to specify the locations of celestial bodies that perturb the vehicle. They are ellipse data and ephemeris-tape data. If the problem does not involve perturbing bodies (except a reference body) or if elliptic data are used for all the perturbing bodies, skip to instruction 5.

(2) If the perturbing-body data are to be taken from an ephemeris tape, list the names of the ephemerides and Julian dates to be covered along with the following auxiliary information:

1st card: $DATA = 300, $TABLE, 2 = TAPE 3, 17 = ELIST, 29 = TBEGIN, 30 = TEND/

Other cards: TAPE 3 = 0

TBEGIN = ephemeris beginning Julian date

TEND = ephemeris ending Julian date

ELIST = (names of perturbing bodies in "ALF" format, see example in text)

The ephemerides of all planets except Earth bear the name of the planet. The ephemeris giving the distance from Earth to the Sun is called "sun," as is astronomical practice.

(3) If successive files on the ephemeris tape are to be made, punch the corresponding sets as follows:

$DATA = 300, TAPE 3 = 0, TBEGIN = , TEND = , ELIST =

As many similar sets as are needed may be appended.

(4) If ellipse data are to be loaded from cards, they are prepared later under instruction 11.

(5) On the first execution after loading the routine, the common area is cleared whether an ephemeris tape is constructed or not. It is now necessary to load a table of variable names. Once loaded, this table will not be cleared again (except if the control variable TAPE 3 is set equal to zero). These names are for use on the input cards. If a different name is desirable for any

variable, it may be changed in the table and where it appears on the input card (ref. 7). The cards are:

```
$DATA=1,$TABLE, 33=DTOFFJ, 34=TOFFT, 711=TIME, 716=X, 717=Y, 718=Z,
713=VX, 714=VY, 715=VZ, 11.=IMODE, 713=E, 714=OMEGA, 715=NODES,
716=INCL, 717=MA, 718=P, 43=LAT, 44=LONG, 45=AZI, 46=ELEV, 14=ALT,
47=VEL, 16=TFILE, 28=TMIN, 153=BODYCD, 177=ELIPS, 30.=MODOUT,
27=STEPS, 29=DELMAX, 26=STEPMX, 23=EREF, 24=ERLIMT, 4.=NSAVE,
5=RECALL, 3=CLEAR, 18.=LOOKX, 22=XLOOK, 19.=LOOKSW, 20=SWLOOK,
609.=INLOOK, 15=END, 31=ATMN, 32=RATM, 49=ROTATE, 417=COEFN,
163.=ICC, 60=BETA, 50=OBLATN, 73=TB, 93=FLOW, 103=SIMP, 123=AREA,
143=DELT, 83=RMASS, 113=AEXIT, 133.=IDENT, 48.=LSTAGE, 25=TKICK /
```

(6) The initial position and velocity of the vehicle may be given in any one of the three coordinate systems. If the initial data are given in orbit elements, skip to instruction (8). If the initial data are given in rectangular coordinates, skip to instruction (7). If the initial data are given in Earth-centered spherical coordinates, the following variables should be punched:

LAT = latitude, deg, positive north of equator

LONG = longitude, relative to Greenwich, deg

ALT = altitude above sea level, m

AZI = azimuth angle, east from north, deg

ELEV = elevation angle, horizontal to path, deg

VEL = initial relative velocity, m/sec

TKICK = size of initial vertical, nondrag step to facilitate starting, sec

If the Earth is assumed to be rotating but aerodynamic forces are not to be considered, set
ROTATE = Earth rotation rate, 7.29211585×10^{-5} radian/sec

If integration in rectangular coordinates is desired set
IMODE = 4
or else if integration in orbit elements is desired set
IMODE = -4

Skip to instruction (9).

(7) If the initial data are in rectangular coordinates, set the following variables:

X = x-component of position in x,y,z coordinate system, m

Y = y-component of position in x,y,z coordinate system, m

Z = z-component of position in x,y,z coordinate system, m

VX = x-component of velocity in x,y,z coordinate system, m/sec

VY = y-component of velocity in x,y,z coordinate system, m/sec

VZ = z-component of velocity in x,y,z coordinate system, m/sec

If integration in rectangular coordinates is desired set
IMODE = 2
or else, if integration in orbit elements is desired set
IMODE = -2

Skip to instruction (9).

(8) If the initial data are in orbit elements, set the following variables:

E = eccentricity

OMEGA = argument of pericenter, radians

NODES = longitude of ascending node (to mean vernal equinox of 1950.0), radians

INCL = orbit inclination to mean equator of 1950.0, radians

MA = mean anomaly, radians

P = semilatus rectum, m

If integration in orbit elements is desired set
IMODE = 1
or else, if integration in rectangular coordinates is desired set
IMODE = -1

(9) To specify takeoff time, set the following variables:

DTOFFJ = Julian day number

TOFFT = fraction of day

TIME = time from previously set Julian date, sec

Takeoff occurs at the instant (ephemeris time) corresponding to the sum of the last three quantities. If a specific date or time is not required, these variables may be skipped. In that case, the SUBROUTINE STDATA sets DTOFFJ to 2440 000.

(10) To specify the origin and any perturbing bodies, list them as BODYCD = (list of body names in "ALF" format, see text example). The first body in the list is taken to be the reference body. The distances between the bodies in

this list must be computable from either ellipse data (instruction (11)) or ephemeris-tape data (instruction (2)). There may be no more than eight names in the list. Also, if the ephemeris tape is being used, the correct file must be found on it. For this purpose, set TFILE = desired ephemeris-tape file. The ephemeris files were numbered in sequence when written in instruction (2). If TFILE is not given, it will be set equal to 1.0 by the SUBROUTINE STDATA.

(11) For each body whose path is represented by an ellipse, a 12-element set of data must be loaded. A 12-element set consists of:

1. Body name in "ALF" format (maximum of six characters)
2. Reference body name in "ALF" format (maximum of six characters)
3. Mass of body, sun mass units
4. Radius of sphere of influence, m
5. Semilatus rectum, AU
6. Eccentricity
7. Argument of pericenter, radians
8. Longitude of ascending node (to mean vernal equinox of 1950.0), radians
9. Orbit inclination (to mean equator of 1950.0), radians
10. Julian day at perihelion
11. Fraction of day at perihelion
12. Period, mean solar days

It is convenient to punch a 12-element set in sequence and to separate the elements by commas on as many cards as are required. Several sets may then be loaded consecutively. The order of the sets is immaterial. Ellipse data, if present, take precedence over ephemeris-tape data. The sets are loaded consecutively, in any order, as follows:

ELIPS = set 1, set 2, set 3, . . ., set n; $n \leq 10$ (see example in appendix I)

(12) If oblateness effects of the Earth are to be included, set

OBLATN = (ALF5)EARTH

(13) Provision has been made to fly multistage vehicles with up to 10 stages. At least one stage must be loaded. There are eight parameters for each stage with provision for input-controlled modifications of other variables. The 10 values of each parameter are stored in an array corresponding to the

10 stages. Input cards are as follows:

TB = burning time for 1st stage, 2nd stage, etc., sec

FLOW = propellant flow rate for 1st stage, 2nd stage, etc., kg/sec

SIMP = vacuum specific impulse of 1st stage, 2nd stage, etc., sec

AREA = aerodynamic reference area of 1st stage, 2nd stage, etc., m^2

AEXIT = engine exit area for 1st stage, 2nd stage, etc., m^2

RMASS = initial mass or jettison mass for 1st stage, 2nd stage, etc., kg

DELT = initial integration step size for 1st stage, 2nd stage, etc., sec

IDENT = input identification number 1st stage, 2nd stage, etc.

TB must be loaded for as many stages as are to be flown. Others may be omitted if zero is appropriate. If RMASS(i) is not positive, the i^{th} stage begins with the final mass of the previous stage reduced by the fixed amount RMASS(i). In the case of DELT, zero will result in use of TB/100. IDENT of a nonzero value will cause any data cards of that identification number to be read in after the stage is set up and before integration begins. This permits the user to make almost any change desired. The order of data cards is discussed in instruction (24).

(14) The thrust orientation must be specified by setting

BETA = angle β, deg (see sketch (a) (p. 4))

COEFN (I) = angle-of-attack schedule, $\alpha = \alpha(t)$ (see instruction (16))

ICC = fixed-point integer (see instruction (16))

For the special case of tangential thrust, none of the last three variables need be set.

(15) If aerodynamic forces are present, set in addition to AREA in instruction (13):

ATMN = name of body that has atmosphere, in "ALF" format, (Earth)

RATM = radius above which atmospheric forces are not to be considered, m

ROTATE = atmospheric-rotation rate, radians/sec (7.29211585×10^{-5} for Earth)

BETA = angle β, deg (see sketch (a))

COEFN (I) = angle-of-attack schedule, $\alpha = \alpha(t)$, $C_L/\sin\alpha$, $C_{D,O}$, and $C_{D,i}/C_L^2$ curves (see instruction (16))

ICC = fixed-point integers (see instruction (16))

(16) If neither thrust nor aerodynamic forces are present, skip to instruction (18). The relations $\alpha(t)$, $C_L/\sin\alpha$, $C_{D,O}$, and $C_{D,i}/C_L^2$ are assumed to be quadratic functions that involve coefficients, which are located in the COEFN(J) array. The arrangement of these coefficients is best explained by an example. Suppose the function $\alpha(t)$ is as follows:

$$\alpha = \begin{cases} a_{11} + a_{12}t + a_{13}t^2 & (t_1 \leq t \leq t_2) \\ a_{21} + a_{22}t + a_{23}t^2 & (t_2 \leq t \leq t_3) \\ a_{31} + a_{32}t + a_{33}t^2 & (t_3 \leq t \leq t_4) \\ \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot \\ \text{etc.} & \text{etc.} \end{cases}$$

The coefficients $a_{i,j}$ should then be loaded into the COEFN(J) array as:

$$\text{COEFN(J)} = t_1, a_{11}, a_{12}, a_{13}, t_2, a_{21}, a_{22}, a_{23}, t_3, a_{31}, a_{32}, a_{33}, t_4, \cdot \cdot \cdot, t_n$$

Furthermore, additional sets of coefficients for the other functions may simply be added to the COEFN(J) array, which results in a string of sets of coefficients, and can be represented, for example, as:

$$\begin{aligned} \text{COEFN(J)} &= \alpha \text{ coefficients, } C_L/\sin\alpha \text{ coefficients, } C_{D,O} \text{ coefficients, etc.} \\ &= t_1, a_{11}, a_{12}, \cdot \cdot \cdot, t_n, N_{M,1}, b_{11}, b_{12}, \cdot \cdot \cdot, N_{M,k}, \text{ etc.} \end{aligned}$$

The starting point in the COEFN(J) array of each function must also be loaded to identify the correct region of coefficients. To this end, the following array must also be loaded:

ICC(1) = fixed-point value of J where α coefficients begin

ICC(2) = fixed-point value of J where $C_L/\sin\alpha$ coefficients begin

ICC(3) = fixed-point value of J where $C_{D,i}/C_L^2$ coefficients begin

ICC(4) = fixed-point value of J where $C_{D,O}$ coefficients begin

For this purpose, all values in the COEFN(J) array are called coefficients (i.e., the t's and the N_M's are coefficients). The sequence of the sets is arbitrary, since changing the sequence requires only a change in the ICC(I) array. (See appendix I for Example II, the lunar orbiting probe.)

(17) The size of the integration steps is determined primarily by the error control variables. These are loaded as:

EREF = error reference value; $\bar{\delta}$ in appendix D

ERLIMT = maximum value of δ that is acceptable on any particular step

EREF is always treated as a positive number; however, if it is loaded with a minus sign, this will cause error information to be printed at the completion of the problem. If no error control data is loaded, SUBROUTINE STDATA will set EREF = 1×10^{-6}, ERLIMT = 3×10^{-6}.

(18) The output control offers a choice on the frequency of output data as follows:

If MODOUT = 1, output will occur every n^{th} step (n = STEPS) until t = TMIN, and then MODOUT is set equal to 2 by the program

If MODOUT = 2, output occurs at equal time intervals of DELMAX until t = TMAX

If MODOUT = 3, output occurs at equal time intervals of DELMAX until t = TMIN, then MODOUT is set equal to 4 by the program

If MODOUT = 4, output occurs every n^{th} step (n = STEPS) until t = TMAX

STEPMX = maximum step limit before problem is completed

DELMAX = time interval between outputs

STEPS = number of steps between outputs

TMIN = time when MODOUT changes

Note that output control may, at times, strongly influence the integration step size especially if MODOUT is 2 or 3 and DELMAX is small. STDATA will put MODOUT = 4 and STEPS = 1.

Note that TMAX = time at start of a stage, plus the stage time, TB(NSTAGE), and is computed internally.

(19) Provision has been made to interrupt the integration procedure when an arbitrary value of an arbitrary parameter is attained. By interrupt it is meant that an output will occur at this point, input is permissible, and a decision is made whether to continue the stage, terminate the stage, or terminate the flight. Skip to instruction (20) if this facility is not desired. To cause an interrupt, set

LOOKX = COMMON C location of arbitrary parameter

XLOOK = value of C(LOOKX) where an interrupt is desired

INLOOK = input identification number for interrupt

END = a negative number if flight should be terminated, zero if stage should continue, or a positive number if stage should be terminated

If the interrupt is not desired the first time C(LOOKX) = XLOOK, set

LOOKSW = COMMON C location of a second arbitrary parameter

SWLOOK = value of C(LOOKSW), which must be equaled or exceeded before an interrupt may occur (interrupt occurs if C(LOOKX) = XLOOK and C(LOOKSW) $\geq$ SWLOOK)

Typically, time may be the second arbitrary parameter; thus, STDATA sets LOOKSW = 711, the COMMON location of time. INLOOK of a nonzero value will cause any data cards of that identification number to be read-in prior to the interrogation of END. The order of the cards is discussed in instruction (24).

(20) Provision has been made to save a block of initial conditions and program control parameters prior to the integration of the n^{th} stage. This allows the flight to be flown again from the n^{th} stage onward with prescribed alterations. Skip to instruction (21) if this facility is not desired. To save the program control variable array, A, and the integration variable array, XPRIM + XPRIMB, just prior to integration of the n^{th} stage, set

NSAVE = the number of the n^{th} stage

The saved data, stored in the D array, will be returned to the A and XPRIM + XPRIMB arrays after the flight is completed if

RECALL = any nonzero number

It is intended that changes in the succeeding flight will be made at the main input station ($DATA=1). NSAVE and RECALL are not contained in the array A and are therefore unaffected by the save-recall sequence. The correct sequence of these controls is not always simple and an understanding of the main program and input stationing is quite desirable.

(21) If the standard set of data contained in the SUBROUTINE STDATA is not desired, set

CLEAR = any nonzero number

It is intended that this control shall be set nonzero by the $DATA = 99 input station at the beginning of the main program. It is not affected by the save-recall sequencing explained in instruction (20).

(22) If the number of stages to be flown is not equal to the number of consecutive nonzero flight times, TB, set

LSTAGE = number of last stage to be flown

(23) When a transfer of origin occurs, provision has been made to read input into the program. This is done with the aid of $DATA = 101, followed by the data statements desired.

(24) The sequencing of the input cards is not always simple and no rigid rules may be written down. Inspection of the program may be necessary to answer some questions. However, in general, the first input cards belong to the $DATA = 300 group if an ephemerides tape is required. This group is followed by the $DATA = 1 group, which consists of the main input for a single flight. Following this are the in-flight input cards, if any, which may be any combination of $DATA = 101, $DATA = INLOOK, or $DATA = IDENT (NSTAGE) groups. The order of these groups of cards matches the order of the time sequence of events in the flight itself. For multiple flights, sets of the above groups may be added in tandem. It is usually desirable in this case, however, to read all the $DATA = 300 sets at the same time (as in instruction (3)) to avoid excessive tape handling.

(25) Following is an input check list that may be helpful at execution time:

INPUT CHECK LIST[a]

Takeoff time	Position and velocity (completely fill in one and only one block)			Reference and perturbing bodies	
				BODYCD =	
	Rectangular	Orbit elements	Spherical	Tape	Elliptic
DTOFFJ = TOFFT = TIME =	X = Y = Z = VX = VY = VZ = IMODE = 2	E = OMEGA = NODES = INCL = MA = P = IMODE = 1	LAT = LONG = AZI = ELEV = ALT = VEL = TKICK IMODE = 4	[b]TAPE 3 = 0 [b]TBEGIN = [b]TEND = [b]ELIST = TFILE =	ELIPS =

Output control	Error control	Restart feature	Parameter search	Atmosphere and coefficients	Oblateness - rotation	Stage data
TMIN = MODOUT = STEPS = DELMAX = STEPMX =	EREF = ERLIMT =	NSAVE = RECALL = CLEAR =	LOOKX = XLOOK = LOOKSW = SWLOOK = INLOOK = END =	ATMN = RATM = COEFN = ICC = BETA =	OBLATN = ROTATE =	TB = FLOW = SIMP = AREA = DELT = RMASS = AEXIT = IDENT = LSTAGE =

[a] The following standard data are loaded by SUBROUTINE STDATA:

DTOFFJ = 2440 000.0
IMODE = 1
BODYCD(1) = (ALF5)EARTH
RMASS(1) = 1.0

MODOUT = 4
STEPS = 1.0
STEPMX = 100.0
LOOKSW = 711

EREF = 1×10^{-6}
ERLIMT = 3×10^{-6}
TFILE = 1.0

[b] At input 300, setting TAPE 3 = 0 is necessary to make an ephemeris tape.

APPENDIX H

PROGRAM LISTING

```
C
C      THIS MAIN PROGRAM IS THE SUPERSTRUCTURE ABOVE ALL SUBPROGRAMS.
C      SUBROUTINE TAPE CLEARS COMMON 1 THRU 4000 AND MAY CONSTRUCT AN
C      EPHEMERIS TAPE, ALSO, IT ALWAYS SETS TAPE3 =0. SUBROUTINE STDATA
C      LOADS A STANDARD SET OF DATA. IF RECALL DOES NOT EQUAL ZERO, A
C      PREVIOUSLY SAVED SET OF DATA(FROM STAGE) IS MOVED TO THE INITIAL
C      DATA LOCATION. THE MAIN INPUT STATION IS STATEMENT 8(INPUT1)
C      WHERE THE VEHICLE DATA FOR ALL STAGES MAY BE LOADED. SUBROUTINE ORDER IS
C      CALLED TO ORDER THE LIST OF BODIES, DETERMINE THE GRAVITATIONAL CONSTANT,
C      ORIGIN ROTATION RATE, ATMOSPHERIC RADIUS, RELOCATE ELLIPTIC EPHEMERIS DATA
C      AND POSITION THE EPHEMERIS TAPE.
C
       COMMON C
C
       DIMENSION A(600), B(700), C(4000),
      1   TB(10), D(1100)
C
       EQUIVALENCE
      1(A     ,C     (  11)),(B     ,C     (1111)),(CLEAR ,C     (   3)),
      2(D     ,C     (2111)),(LSTAGE,A     (  38)),(NCASE ,C     (   1)),
      3(NCASES,A     ( 600)),(NSTAGE,A     (   3)),(RECALL,C     (   5)),
      4(TB    ,A     (  63)),(TAPE3 ,C     (   2)),(TABLE ,C     (1911))
C
     1 CALL INPUT (99,C,TABLE)
       IF (TAPE3) 3,2,3
     2 CALL TAPE
     3 NCASE = NCASE + 1
       WRITE OUTPUT TAPE 6,12,NCASE
    12 FORMAT(12H1CASE NUMBERI3,1H.)
       IF (CLEAR) 5,4,5
     4 CALL STDATA
     5 IF (RECALL) 6,8,6
     6 DO 7 J=1,1100
     7 A(J) = D(J)
       WRITE OUTPUT TAPE 6,16,NSTAGE,NCASES
    16 FORMAT(33H RECALLED INITIAL DATA FROM STAGEI2,8H OF CASEI4,1H.)
     8 CALL INPUT (1,C,TABLE)
       IF (SENSE SWITCH 6) 13,14
    13 WRITE OUTPUT TAPE 6,15
    15 FORMAT(19H0EXIT VIA SENSE SW6)
       CALL EXIT
    14 IF (LSTAGE) 11,9,11
     9 DO 10 LSTAGE=1,10
       IF (TB(LSTAGE+1)) 10,11,10
    10 CONTINUE
       LSTAGE = 10
    11 CALL ORDER
    17 CALL STAGE
       GO TO 1
       END
```

```
                                                  SUBROUTINE AERO
C      SUBROUTINE AERO COMPUTES THE LIFT AND DRAG ACCELERATIONS.  AS IN SUBROUT-
C      INE THRUST, THESE VECTORS ARE REFERENCED TO THE RELATIVE WIND VELOCITY.
C      COEFFICIENTS OF LIFT, INDUCED DRAG, AND DRAG AT ZERO ANGLE OF ATTACK ARE
C      ASSUMED TO BE FUNCTIONS OF MACH NUMBER AND ANGLE OF ATTACK.  TABLES OF
C      CDI/CL**2, CL/SIN(ALPHA), AND CDO ARE ASSUMED AS FITTED QUADRATIC EQUAT-
C      IONS IN THE COEFN ARRAY.  GASFAC IS THE SQRTF(SPECIFIC HEAT RATIO * STAND-
C      ARD ACCELERATION OF GRAVITY * UNIVERSAL GAS CONSTANT).  FOR EARTH, GASFAC=
C      20.064881 (METERS / SEC**2 / KELVIN DEGREE)**1/2.
C
       COMMON C
C
       DIMENSION A(600), B(700), C(4000),
      1VATM(3), P(3), XIFT(3), DRAG(3), PAR(3), X(100)
C
       EQUIVALENCE
      1(A     ,C     (  11)),(ALPHA ,A     (  49)),(AREA  ,B     (   6)),
      2(B     ,C     (1111)),(BETA  ,A     (  50)),(CD    ,A     ( 165)),
      3(CDI   ,A     ( 163)),(CL    ,A     ( 164)),(COSALF,B     (  48)),
      4(COSBET,B     (  49)),(DNSITY,B     (  29)),(DRAG  ,B     (  69)),
      5(GASFAC,A     (  46)),(P     ,B     (  84)),(PAR   ,B     (  60)),
      6(PMAGN ,B     (  50)),(Q     ,B     (  59)),(R     ,B     ( 102)),
      7(SINALF,B     (  46)),(SINBET,B     (  47)),(TM    ,B     (  34)),
      8(VATM  ,B     (  97)),(VMACH ,B     (  38)),(VQ    ,B     ( 100)),
      9(VQSQRD,B     ( 101)),(X     ,B     ( 401)),(XIFT  ,B     (  72))
C
       Q = 0.5*DNSITY*VQSQRD
       QVAL = Q*AREA/X(2)
       VMACH=SQRTF(VQSQRD/TM)/GASFAC
C
C      COMPUTE THE X,Y,Z COMPONENTS OF LIFT.
       IF (ALPHA) 2,1,2
     1 CL = 0.0
       CDI=0.0
       GO TO 4
     2 CL = QUAD(VMACH,2)*SINALF
       AA = QVAL*CL/PMAGN
       AB = SINBET/VQ
       DO 3 K=1,3
     3 XIFT(K) = AA*(AB*PAR(K)+COSBET*P(K))
     7 CDI=QUAD(VMACH,3)*CL*CL
C
C      COMPUTE THE X,Y,Z COMPONENTS OF DRAG.
     4 CD = CDI+QUAD(VMACH,4)
       AC = -CD*QVAL/VQ
       DO 5 K=1,3
     5 DRAG(K) = AC*VATM(K)
     6 RETURN
       END
```

```
                                          FUNCTION ARCTAN (Y,X)
C     THE FORTRAN II LIBRARY ATANF(+ OR - Z=TAN(THETA)) USES A SINGLE
C     ARGUMENT WITH ITS SIGN TO GIVE THETA IN THE FIRST (+Z) OR FOURTH
C     (-Z) QUADRANT.
C
C     THE ARCTAN FUNCTION MAY BE USED IF + OR - Z IS DERIVED FROM A
C     FRACTION SO THAT ARCTAN (Y,X) = TAN-1 ((+OR-Y=SIN(THETA))/(+OR-X=
C     COS(THETA))). THUS THE ARCTAN (Y,X) GIVES THETA IN ITS PROPER
C     QUADRANT FROM -180 DEGREES TO +180 DEGREES.
C
      IF (X) 2,1,2
    1 ARCTAN=SIGNF(1.57079632,Y)
      GO TO 4
    2 ARCTAN=ATANF(Y/X)
      IF(X) 3,1,4
    3 ARCTAN=ARCTAN+SIGNF(3.14159265,Y)
    4 RETURN
      END
```

```
                                        SUBROUTINE CONVT1 (V,AMC)
C     THIS ROUTINE COMPUTES -- (1) ANGULAR MOMENTUM, AMC(4)
C                              (2) ANGULAR MOMENTUM SQUARED, AMC(5)
C                              (3) X,Y,Z COMPONENTS OF ANG. MOM., AMC(J)
C                              (4) VELOCITY, V(4)
C                              (5) VELOCITY SQUARED, V(5)
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   AMC(3), V(5), RB(3), IND(3)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(B     ,C      (1111)),(IND   ,A      (  60)),
     2(RB    ,B      ( 193))
C
      DO 1 J1=1,3
      J2=IND(J1)
      J3=IND(J2)
    1 AMC(J3) = RB(J1)*V(J2)-RB(J2)*V(J1)
      AMC(5) = AMC(1)**2+AMC(2)**2+AMC(3)**2
      AMC(4) = SQRTF(AMC(5))
      V(5) = V(1)**2+V(2)**2+V(3)**2
      V(4) = SQRTF(V(5))
      RETURN
      END
```

```
                                                SUBROUTINE CONVT2
C     THIS ROUTINE CONVERTS RECTANGULAR COORDINATES INTO ORBIT ELEMENTS.
C     RECTANGULAR COORDINATES- POSITION COMPONENTS,X,AND VELOCITY COMPONENTS,VX.
C     THE ORBIT ELEMENTS ARE IN THE ORBELS ARRAY-
C     (1) ECCENTRICITY            (4) INCLINATION
C     (2) ARGUMENT OF PERICENTER           (5) MEAN ANOMALY
C     (3) LONGITUDE OF ASCENDING NODE            (6) SEMILATUS RECTUM
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   AMC(3), ORBELS(6), RB(3)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(AM    ,B      (  90)),(AMSQRD,B      (  91)),
     2(AMC   ,B      (  87)),(B     ,C      (1111)),(COSTRU,B      (  53)),
     3(EPAR  ,B      (  26)),(GK2M  ,B      (  36)),(ORBELS,B      ( 116)),
     4(R     ,B      ( 102)),(RB    ,B      ( 193)),(SINTRU,B      (  52)),
     5(TRU   ,B      (  40)),(V     ,B      (  95)),(VSQRD ,B      (  96)),
     6(VX    ,B      (  92)),(VY    ,B      (  93)),(VZ    ,B      (  94))
C
      ORBELS(6)=AMSQRD/GK2M
      R=SQRTF(RB(1)**2+RB(2)**2+RB(3)**2)
      TRU=ARCTAN(AM/GK2M*(RB(1)*VX+RB(2)*VY+RB(3)*VZ),ORBELS(6)-R)
      IF(AMC(1)) 2,1,2
    1 ORBELS(3)=0.
      GO TO 3
    2 ORBELS(3)=ARCTAN(AMC(1),-AMC(2))
    3 ORBELS(4)=ARCTAN(SQRTF(AMC(1)**2+AMC(2)**2),AMC(3))
      SNODE=SINF(ORBELS(3))
      CNODE=COSF(ORBELS(3))
      AA=RB(1)*CNODE+RB(2)*SNODE
      AB=RB(3)*SINF(ORBELS(4))+COSF(ORBELS(4))*(RB(2)*CNODE-RB(1)*SNODE)
      ORBELS(2)=ARCTAN(AB,AA)-TRU
      ORBELS(1)=SQRTF(ABSF(1.+ORBELS(6)*(VSQRD/GK2M-2./R)))
      EPONE=SQRTF(1.+ORBELS(1))
      E2M1=1.-ORBELS(1)**2
      EPAR=SQRTF(ABSF(E2M1))
      SINTRU=SINF(TRU)
      COSTRU=COSF(TRU)
      EPAS=SQRTF(ABSF(1.-ORBELS(1)))*SINTRU/(1.+COSTRU)
      ETHETA=ORBELS(1)*SINTRU/(1.+ORBELS(1)*COSTRU)*EPAR
    4 IF(E2M1) 5,6,6
    5 ORBELS(5)=LOGF((EPONE+EPAS)/(EPONE-EPAS))-ETHETA
      GO TO 7
    6 ORBELS(5)=2.*ARCTAN(EPAS,EPONE)-ETHETA
    7 RETURN
      END
```

```
                                                    SUBROUTINE ERRORZ
C    THIS SUBROUTINE COMPUTES THE RELATIVE ERRORS BETWEEN THE R-K AND LOW-ORDER
C    INTEGRATION SCHEMES.  IT ALSO COMPUTES THE ERROR COEFFICIENT, A, AND SAVES
C    THE ERROR DATA WHEN EREF HAS A - SIGN.  THE BRANCH ON IMODE DETERMINES
C    WHICH SET OF NORMALIZING FACTORS ARE TO BE USED.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   RELERR (8),   XPRIM (200),  XINC (100)
C
      EQUIVALENCE
     1(A     ,C     (  11)),(A1    ,B     (  10)),(A2    ,B     (  11)),
     2(B     ,C     (1111)),(DELT  ,B     (   1)),(E2    ,B     (  18)),
     3(EREF  ,A     (  13)),(IMODE ,A     (   1)),(INDERR,B     (  51)),
     4(R     ,B     ( 102)),(STEPGO,A     (  41)),(STEPNO,A     (  42)),
     5(V     ,B     (  95)),(XINC  ,B     ( 601)),(XPRIM ,C     ( 711))
      EQUIVALENCE (RELERR,XINC)
C
      E2 = 0.
      RELERR(2) = XINC(2)/XPRIM(2)
      IF (IMODE-1) 2,1,2
C
C     COMPUTE THE NORMALIZED INTEGRATION ERRORS FOR THE ORBIT ELEMENTS.
    1 RELERR(3)=XINC(3)/(XPRIM(3)+1.)/10.
      RELERR(8)=XINC(8)/XPRIM(8)/10.
      DO 10 J=1,4
   10 RELERR(J+3)=XINC(J+3)/62.831853
      GO TO 3
C
C     COMPUTE THE NORMALIZED INTEGRATION ERRORS IN RECTANGULAR VARIABLES.
    2 V1 = V+100.
      DO 20 J=1,3
      RELERR(J+2)=XINC(J+2)/V1
   20 RELERR(J+5)=XINC(J+5)/R
C
C     SELECT MAXIMUM ERROR, COMPUTE ERROR COEFFICIENT, POSSIBLY SAVE ERROR DATA.
    3 DO 5 J=2,8
      IF (ABSF(RELERR(J))-E2) 5,5,4
    4 K=J
      E2 = ABSF(RELERR(J))
    5 CONTINUE
      E2 = E2 + 2E-8
      A1 = A2
      A2 = LOGF(E2)-5.*LOGF(ABSF(DELT))
      IF (EREF) 6,7,7
    6 WRITE TAPE 4,STEPGO,STEPNO,XPRIM(1),DELT,A2,E2,(RELERR(J),J=2,8),K
      INDERR = INDERR + 1
    7 RETURN
      END


                                                    SUBROUTINE EQUATE
C     THIS SUBROUTINE IS CALLED FROM NBODY  TO EVALUATE THE DERIVATIVES OF THE
C     VARIABLES OF INTEGRATION.  EITHER RECTANGULAR COORDINATES OR ORBIT ELE-
C     MENTS MAY BE USED AS THE VARIABLES OF INTEGRATION, BUT IN THE CASE OF THE
C     LATTER, THE CORRESPONDING RECTANGULAR COORDINATES MUST FIRST BE FOUND.
C     THIS IS DONE AT THE BEGINNING THRU THE USE OF KEPLERS EQUATION.  THE
C     PERTURBATING ACCELERATIONS ARE FOUND BY CALLING VARIOUS OTHER SUBROUTINES
C     AND THEIR SUM RESOLVED ALONG THE X,Y,Z AXIS.  FINALLY, THE DERIVATIVES
C     ARE CALCULATED.  IN THE CASE OF ORBIT ELEMENTS, THE X,Y,Z PERTURBATING
C     ACCELERATION COMPONENTS MUST FIRST BE RESOLVED INTO CIRCUMFERENTIAL,RADIAL
C     AND NORMAL COMPONENTS.  THIS ROUTINE ALSO CHANGES THE INTEGRATION VARI-
C     ABLES FROM ORBIT ELEMENTS TO RECTANGULAR VARIABLES IF THE ECCENTRICITY
C     APPROACHES UNITY.  THE X,XPRIM, AND XDOT ARRAYS ARE AS FOLLOWS.
C
C
C     X            ORBIT ELEMENTS          RECTANGULAR COORDINATES
C
C     1            TIME                     TIME
C     2            MASS                     MASS
C     3            ECCENTRICITY             X-VELOCITY
C     4            ARGUMENT OF PERICENTER   Y-VELOCITY
C     5            ARGUMENT OF ASC. NODE    Z-VELOCITY
C     6            INCLINATION              X
C     7            MEAN ANOMALY             Y
C     8            SEMILATUS RECTUM         Z
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1    XPRIM (100,2),        VX (3),             QX (3),
     2       RB (3),        NEFMRS (8),             X (100),
     3   XPRIMB (100,2),     FORCE (3),           XIFT (3),
     4     DRAG (3),         OBLAT (3),          COMPA (3),
     5     XDOT (100)
C
```

```
      EQUIVALENCE
     1(A     ,C      (  11)),(AMSQRD,B      (  91)),(ASYMPT,A      (   7)),
     2(B     ,C      (1111)),(BNAME ,B      ( 122)),(CINCL ,B      (  55)),
     3(CIRCUM,B      (  82)),(COMPA ,B      (  63)),(CONSTU,A      (  32)),
     4(COSTRU,B      (  53)),(COSV  ,B      (  57)),(DRAG  ,B      (  69)),
     5(EMONE ,B      (  28)),(EPAR  ,B      (  26)),(ETOL  ,A      (  30)),
     6(EXMODE,B      (  27)),(FLOW  ,B      (   5)),(FORCE ,B      (  66)),
     7(GK2M  ,B      (  36)),(GKM   ,B      (  37)),(IMODE ,A      (   1)),
     8(KSUB  ,B      (  19)),(MBODYS,B      (  42)),(NEFMRS,B      ( 185)),
     9(NSTART,B      (  24)),(OBLATN,A      (  40)),(OBLAT ,B      (  75))
      EQUIVALENCE
     1(PRESS ,B      (  33)),(PUSHO ,B      ( 391)),(QX    ,B      (  78)),
     2(RADIAL,B      (  81)),(RATMOS,B      (  23)),(RB    ,B      ( 193)),
     3(R     ,B      ( 102)),(RSQRD ,B      (  45)),(SINCL ,B      (  54)),
     4(SINTRU,B      (  52)),(SINV  ,B      (  56)),(SPD   ,A      (  44)),
     5(TABLT ,B      (  20)),(TOFFT ,A      (  24)),(TRSFER,B      (   8)),
     6(TTEST ,A      (  54)),(U     ,A      (  59)),(V     ,B      (  95)),
     7(VSQRD ,B      (  96)),(VX    ,B      (  92)),(XDOT  ,B      ( 501)),
     8(XIFT  ,B      (  72)),(XPRIM ,C      ( 711)),(XPRIMB,C      ( 911)),
     9(X     ,B      ( 401)),(ZORMAL,B      (  83)),(ZN    ,B      (  43))
C
      TABLT=X(1)/SPD+TOFFT
      IMODE=IMODE
    1 GO TO (2,16,16),IMODE
C
C     STATEMENTS 2 TO 16 FIND THE RECTANGULAR POSITION AND VELOCITY FROM ORBIT
C     ELEMENTS AND TRUE ANOMALY.  THE TRUE ANOMALY IS FOUND FROM ITERATIVE
C     SOLUTION OF KEPLERS EQUATION.
    2 E2 = X(3)**2
      E2M1 = 1.-E2
      EMONE = X(3)-1.
      EPAR=SQRTF(ABSF(E2M1))
      VCIRCL=GKM/SQRTF(X(8))
C
C     COMPUTE SINE AND COSINE OF TRUE ANOMALY.
C     PART A. E=1
    3 IF (EMONE) 10,4,5
    4 SINTRU = 0.
      COSTRU = 1.
      GO TO 14
C
C     PART B. E IS GREATER THAN 1
    5 DO 7 J=1,100
      DELM = X(7)-U+X(3)*SINHF(U)
      ECOSU = X(3)*COSHF(U)
      DELU = DELM/(1.0-ECOSU)
      U = U+DELU
    6 IF (ABSF(DELM)-CONSTU) 9,9,7
    7 CONTINUE
      ASYMPT = 1.0
      IF (MBODYS) 8,23,8
    8 CALL EPHMRS
      GO TO 23
    9 COSU = COSHF(U)
      DEM1 = 1.-X(3)*COSU
      COSTRU = (COSU-X(3))/DEM1
      SINTRU =-EPAR*SINHF(U)/DEM1
      GO TO 14
C
C     PART C. E IS LESS THAN 1
   10 DO 12 J=1,15
      DELM = X(7)-U+X(3)*SINF(U)
      ECOSU = X(3)*COSF(U)
      DELU = DELM/(1.0-ECOSU+0.01*ECOSU**3)
      U = U+DELU
   11 IF (ABSF(DELM)-CONSTU) 13,13,12
   12 CONTINUE
      WRITE OUTPUT TAPE 6,55,U,DELU
      CALL EXIT
   13 COSU = COSF(U)
      DEM1 = 1.-X(3)*COSU
      COSTRU = (COSU-X(3))/DEM1
      SINTRU = EPAR*SINF(U)/DEM1
   14 POVR = 1.+X(3)*COSTRU
C
C     COMPUTE POSITION AND VELOCITY FROM ORBIT ELEMENTS AND TRUE ANOMALY.
C     ALSO, CLEAR THE PERTURBATING ACCELERATIONS.
   15 SOMEGA = SINF(X(4))
      COMEGA = COSF(X(4))
      SNODE = SINF(X(5))
      CNODE = COSF(X(5))
      SINCL = SINF(X(6))
      CINCL = COSF(X(6))
      SINV=SINTRU*COMEGA+COSTRU*SOMEGA
      COSV=COSTRU*COMEGA-SINTRU*SOMEGA
      AR=COSV*CNODE-SINV*SNODE*CINCL
      B1=SINV*CNODE+COSV*SNODE*CINCL
      C1=COSV*SNODE+SINV*CNODE*CINCL
      D1=SINV*SNODE-COSV*CNODE*CINCL
      E1 = X(3)*SOMEGA+SINV
      F1 = X(3)*COMEGA+COSV
      AS=E1*CNODE+F1*SNODE*CINCL
      B2=F1*CNODE*CINCL-E1*SNODE
      R = X(8)/POVR
      RSQRD = R*R
      SINVY=SINV*SINCL
      RB(1)    = R*AR
      RB(2)    = R*C1
      RB(3)    = R*SINVY
      VX(1)=-VCIRCL*AS
      VX(2)=VCIRCL*B2
      VX(3)=VCIRCL*F1*SINCL
      GO TO 18
```

```
C
   16 DO 17 K=1,3
      VX(K) = X(K+2)
   17 RB(K) = X(K+5)
      RSQRD = RB(1)*RB(1) + RB(2)*RB(2) + RB(3)*RB(3)
      R=SQRTF(RSQRD)
   18 VSQRD=VX(1)*VX(1)+VX(2)*VX(2)+VX(3)*VX(3)
      V = SQRTF(VSQRD)
      DO 19 I=1,15
   19 FORCE(I) = 0.
C
C     TEST FOR PRESENCE OF PERTURBING BODIES.
      IF (MBODYS) 20,21,20
   20 CALL EPHMRS
   21 IF (XABSF(IMODE)-1) 26,22,26
C
C     TEST FOR CHANGE FROM ORBIT ELEMENTS TO TEMPORARY RECTANGULAR
C     COORDINATES IF E IS TOO NEAR TO UNITY.
   22 IF (ETOL-ABSF(EMONE)) 26,23,23
   23 IF (IMODE) 54,24,24
   24 IMODE=-3
      IF (NSTART) 25,54,25
   25 TTEST = X(1)
   27 CALL TESTTR
C
C     TEST FOR OBLATENESS PERTURBATION COMPUTATION.
   26 IF(OBLATN-BNAME)30,29,30
   29 CALL OBLATE
C
C     TEST FOR PRESENCE OF THRUST.
   30 XDOT(2) = -FLOW
      IF (R-RATMOS) 31,31,32
   31 CALL ICAO
      GO TO 33
   32 PRESS=0.
   33 IF (PUSHO) 37,36,37
   36 ASSIGN 40 TO NDONE
      GO TO 38
   37 CALL THRUST
      ASSIGN 41 TO NDONE
C
C     TEST FOR EXISTENCE OF ATMOSPHERE.  FIND AERODYNAMIC FORCES.
   38 IF (PRESS ) 39,42,39
   39 GO TO NDONE, (40,41)
   40 CALL THRUST
   41 CALL AERO
C
C     SUM COMPONENTS OF THE PERTURBING ACCELERATION.
   42 DO 43 J=1,3
   43 COMPA(J) = -QX(J)+OBLAT(J)+FORCE(J)+XIFT(J)+DRAG(J)
   44 GO TO (47,45,45),IMODE
C
C     COMPUTE DERIVATIVES FOR THE RECTANGULAR VARIABLES OF INTEGRATION.
   45 AA = GK2M/R/RSQRD
      DO 46 K=1,3
      XDOT(K+5) = X(K+2)
   46 XDOT(K+2) = COMPA(K)-AA*X(K+5)
      GO TO 54
C
C     COMPUTE THE DERIVATIVES OF THE ORBIT ELEMENTS.  (AFTER RESOLVING
C     PERTURBATING ACCELERATION INTO CIRCUMFERENTIAL, RADIAL, NORMAL COMPONENTS)
   47 CIRCUM=COMPA(3)*COSV*SINCL-COMPA(1)*B1-COMPA(2)*D1
      RADIAL=COMPA(1)*AR+COMPA(2)*C1+COMPA(3)*SINVY
      ZORMAL=COMPA(1)*SNODE*SINCL-COMPA(2)*CNODE*SINCL+COMPA(3)*CINCL
      ZN=VCIRCL*E2M1*EPAR/X(8)
      RDVPP1 = 1./PDVR + 1.
      RDVA = E2M1/PDVR
      XDOT(8) = 2.*R/VCIRCL*CIRCUM
      IF (X(3)) 48,48,49
   48 CSQRD  = CIRCUM*CIRCUM
      RASQRD = RADIAL*RADIAL
      DEM1 = (4.*CSQRD+RASQRD)*VCIRCL
C
C     TEST FOR IN-PLANE PERTURBATION.
      IF (DEM1) 57,56,57
   56 XDOT(3) = 0.
      XDOT(4) = 0.
      XDOT(7) = 0.
      GO TO 50
   57 VDV2R=VCIRCL/R/2.
      XDOT(3) = SQRTF(4.*CSQRD+RASQRD)/VCIRCL
      XDOT(4) = VDV2R+(2.*CSQRD+RASQRD)/DEM1*RADIAL
      XDOT(7) = ZN-VDV2R+(6.*CSQRD+RASQRD)/DEM1*RADIAL
      GO TO 50
   49 XDOT(3) = (SINTRU*RADIAL+(PDVR-RDVA)/X(3)*CIRCUM)/VCIRCL
      XDOT(4)=(SINTRU/X(3)*RDVPP1*CIRCUM-COSTRU*RADIAL/X(3))/VCIRCL
      XDOT(7)=ZN+EPAR/VCIRCL*((COSTRU/X(3)-2./PDVR)*RADIAL-(SINTRU/X(3)*
     1RDVPP1*CIRCUM))
   50 IF(SINCL) 51,52,51
   51 XDOT(5) = SINV/SINCL*ZORMAL/VCIRCL/PDVR
      GO TO 53
   52 XDOT(5) = 0.
   53 XDOT(6) = COSV*ZORMAL/PDVR/VCIRCL
   54 RETURN
   55 FORMAT(41H0KEPLERS EQUATION CONVERGENCE FAILURE, U=G15.8,7H  DELU=
     1G15.8)
      END
```

```
                                                      SUBROUTINE EPHMRS
C    SUBROUTINE EPHMRS IS CALLED TO COMPUTE THE POSITIONS OF THE PERTURBING
C    BODIES RELATIVE TO THE VEHICLE AND, FROM THESE, THEIR PERTURBING ACCELERA-
C    TIONS UPON THE VEHICLE.  OCCASIONALLY THIS ROUTINE IS CALLED FOR THE PURPOSE
C    OF TRANSLATING THE ORIGIN IN WHICH CASE (TRSFER=1) THE RELATIVE VELOCITIES
C    ARE ALSO CALCULATED.  IF A BODYS POSITION IS TO BE COMPUTED FROM AN ELLIPTIC
C    APPROXIMATION SUBROUTINE ELIPSE IS CALLED. OTHERWISE, THE POSITION WILL BE
C    CALCULATED IN EPHMRS FROM THE PRECISION TAPE EPHEMERIS.  THE DO 19 LOOP
C    ENCOMPASSES ALMOST THE ENTIRE EPHMRS SUBROUTINE AND ,IN EFFECT, ELIPSE TOO.
C
      COMMON  C
C
      DIMENSION A(600), B(700), C(4000),
     1   QX(3), IBODY(8), EFMRS(7), XP(3,8), RB(3,8), R   (8), TIM(7),
     2   NEFMRS(8), TDATA(6,3,7), TDEL(7), BMASS(8), VEFM(3,8), DATA(21)
     3   , TDAT(18,7)
C
      EQUIVALENCE
     1(A     ,C      (   11)),(AU    ,A      (   29)),(B     ,C      (1111)),
     2(BMASS ,B      (  137)),(DTOFFJ,A      (   23)),(EFMRS ,B      ( 130)),
     3(IBODY ,B      (  177)),(MBODYS,B      (   42)),(NEFMRS,B      ( 185)),
     4(QX    ,B      (   78)),(R     ,B      ( 102)),(RB    ,B      ( 193)),
     5(SQRDK ,B      (   35)),(SPD   ,A      (   44)),(TABLT ,B      (  20)),
     6(TDATA ,B      ( 265)),(TDEL  ,B      ( 170)),(TIM   ,B      ( 163)),
     7(TRSFER,B      (    8)),(VEFM  ,B      ( 241)),(XP    ,B      ( 217))
      EQUIVALENCE (IBF,FIB), (TDAT,TDATA)
C
C     PART 2.  SET INDEXS, FIND POSITION IF ELLIPSE IS USED (NEFMRS = 20 OR UP).
      DO 19 JB=1,MBODYS
      JB1 = JB+1
      IBF = IBODY(JB1)
      IB = XABSF(IBF)
      IF (NEFMRS(JB)-20) 2,2,1
    1 CALL ELIPSE (JB1)
      IF (TRSFER) 12,12,17
C
C     PART 3.  TAPE EPHEMERIS IS TO BE USED.  FIND DIFFERENCE (DT) BETWEEN
C     CURRENT PROBLEM TIME (DTOFFJ+TABLT) AND MIDPOINT TIME (TIM) OF CURRENTLY
C     STORED TAPE DATA.  THEN SEE IF CURRENT DATA IS OKAY.  TDEL = TIME INTERVAL
C     ON EITHER SIDE OF TIM FOR WHICH CURRENT DATA IS GOOD.
    2 DT = TABL T - (TIM(JB) -DTOFFJ)
      IF (ABSF(DT)-TDEL(JB)) 10,10,3
C
C     PART 4A.  CURRENT DATA NOT OKAY.  READ IN NEXT DATA SET.  IF DT IS -,
C     BACK UP THE TAPE 2 RECORDS BEFORE READING.
    3 IF (DT) 4,5,5
    4 BACKSPACE 3
      BACKSPACE 3
    5 READ TAPE 3, (DATA(I), I=1,21)
C
C     PART 4B. IF THIS DATA IS FOR A BODY IN THE BNAME LIST, STORE IT.
C     (IF NOT STORED, WE MIGHT HAVE TO RETURN FOR IT.)  IF ELLIPSE DATA IS
C     PROVIDED FOR THE BODY FOUND, BY-PASS THE TAPE DATA AND READ IN NEXT SET.
      DO 7  J = 1,MBODYS
B     IF ((DATA(1)+EFMRS(J))*(-(DATA(1)*EFMRS(J)))) 7,6,7
    6 IF (NEFMRS(J)-20) 8,8,3
    7 CONTINUE
      GO TO 3
C
C     PART 4C.  MOVE THE DATA INTO PLACE AND THEN GO BACK AND SEE IF IT IS OKAY.
    8 TIM(J) = DATA(2)
      TDEL(J) = DATA(3)
      DO 9 JJ=1,18
      TDAT(JJ,J) = DATA(JJ+3)
    9 CONTINUE
      GO TO 2
C
C
C     PART 5.  CURRENT DATA IS OKAY.  GET POSITION FROM THE POLONOMIAL
C     P = A + BX + CX**2 + DX**3 + EX**4 + FX**5.
   10 DO 11 K=1,3
      XP(K,JB1) = TDATA(1,K,JB)
      DO 11 KT=2,6
      XP(K,JB1) = XP(K,JB1)* DT +TDATA(KT,K,JB)
   11 CONTINUE
      IF (TRSFER) 12,12,15
C
C     PART 6. COMPUTE DISTANCE FROM REFERENCE AND FROM ROCKET .
   12 DO 13 K=1,3
      XP(K,JB1) = XP(K,IB) +XP(K,JB1)*SIGNF(AU,FIB)
   13 RB(K,JB1)= RB(K,1) - XP(K,JB1)
C
C     PART 7. COMPUTE PERTURBING ACCELERATIONS (QX).  4194304=2**22 IS REMOVED
C     TO PREVENT OVERFLOW.  2048=2**11 AND 8589934592=2**33 RESTORE THE SCALE.
      PRSQRD = (RB(1,JB1)**2 + RB(2,JB1)**2 + RB(3,JB1)**2)/4194304.
      RRELL = SQRTF(PRSQRD)
      RSQRD = ( XP(1,JB1)**2 + XP(2,JB1)**2 + XP(3,JB1)**2)/4194304.
      RCUBE = RSQRD * SQRTF(R SQRD)
      PRCUBE = PRSQRD * RRELL
      R(JB1) = RRELL*2048.
      DO 14 K=1,3
   14 QX(K)=SQRDK * BMASS(JB1) * ((XP(K,JB1)/RCUBE) + RB(K,JB1)/PRCUBE)/
     1 8589934592. + QX(K)
      GO TO 19
C
C     PART 8.  COMPUTE VELOCITY FROM V = B + 2CX + 3DX**2 + 4EX**3 + 5FX**4
C     AND FROM REFERENCE BODY VELOCITY (VEFM(IB)).
   15 DO 16 K=1,3
      VEFM(K,JB1) = 0.
      DO 16 KT=1,5
   16 VEFM(K,JB1) = (VEFM(K,JB1)*DT+TDATA(KT,K,JB)*FLOATF(-KT+6))
   17 DO 18 K=1,3
   18 VEFM(K,JB1) = VEFM(K,IB) + VEFM(K,JB1)*SIGNF(AU/SPD,FIB)
      GO TO 12
   19 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EXTRA
C
C     THIS ROUTINE IS EXECUTED BETWEEN FLIGHTS AND MAY THEREFORE BE EXPANDED TO
C     DO ADDITIONAL COMPUTATION BETWEEN SUCCESSIVE FLIGHTS.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(B     ,C     (1111)),(QMAX  ,B     (  44)),
     2(SIGNAL,B      (  31))
C
      SIGNAL = 0.
      QMAX = 0.
      RETURN
      END
```

```
                                                  SUBROUTINE EXTRAS
C     THIS ROUTINE IS EXECUTED BETWEEN STAGES AND MAY THEREFORE BE EXPANDED TO
C     DO CALCULATIONS BETWEEN SUCCESSIVE STAGES OF A FLIGHT.
C
      RETURN
      END
```

```
                                                  SUBROUTINE ELIPSE (JB1)
C    THIS SUBROUTINE IS CALLED FROM EPHMRS TO COMPUTE THE POSITION OF A BODY
C    USING APPROXIMATE ELLIPTIC DATA.  THE VELOCITY IS ALSO COMPUTED IF THE
C    ORIGIN IS BEING TRANSLATED (TRSFER=1.0).  THE ELLIPSE DATA IS READ FROM
C    INPUT CARDS AND ORGANIZED IN SUBROUTINE ORDER. TPD IS TIME SINCE PERIHELION
C    PASSAGE, ZM IS MEAN ANOMALY, U IS ECCENTRIC ANOMALY.
C      TDATA ARRAY -  (K)    SEMILATUS RECTUM             (K+7) PERIOD
C                     (K+1)  ECCENTRICITY                 (K+8) SIN OMEGA
C                     (K+2)  OMEGA                        (K+9) SIN NODE
C                     (K+3)  NODE                         (K+10) SIN INCLINATION
C                     (K+4)  INCLINATION                  (K+11) COS OMEGA
C                     (K+5)  JD OF PERIHELION             (K+12) COS NODE
C                     (K+6)  FRACTIONAL PART OF (K+5)     (K+13) COS INCLINATION
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   XP(3,8), VEFM(3,8), TDATA(121)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(B     ,C     (1111)),(CONSU ,A     (  31)),
     2(DTOFFJ,A      (  23)),(TABLT ,B     (  20)),(TDATA ,B     ( 265)),
     3(TRSFER,B      (   8)),(VEFM  ,B     ( 241)),(XP    ,B     ( 217))
C
      K = 18*(JB1-2)+1
      TPD = (DTOFFJ-TDATA(K+5))+(TABLT-TDATA(K+6))
      ZN = 6.28318533/TDATA(K+7)
      ZM = ZN*MODF(TPD,TDATA(K+7))
C
C     GET THE SINE(SINTRU) AND THE COSINE (COSTRU) OF THE TRUE ANOMALY
C     BY ITERATING KEPLERS EQUATION. THEN COMPUTE X,Y,Z (XP).
      U = ZM+TDATA(K+1)*SINF(ZM)+.5*TDATA(K+1)**2*SINF(2.*ZM)
      DO 1 J=1,10
      DELM = ZM-U+TDATA(K+1)*SINF(U)
      DELU = DELM/(1.-TDATA(K+1)*COSF(U))
      U = U+DELU
      IF (ABSF(DELM)-CONSU) 2,2,1
    1 CONTINUE
    2 COSU = COSF(U)
      DENOM = 1.-TDATA(K+1)*COSU
      COSTRU = (COSU-TDATA(K+1))/DENOM
      R = TDATA(K)/(1.+TDATA(K+1)*COSTRU)
      SINTRU = SQRTF(1.-TDATA(K+1)**2)*SINF(U)/DENOM
      SINV = SINTRU*TDATA(K+11)+COSTRU*TDATA(K+8)
      COSV = COSTRU*TDATA(K+11)-SINTRU*TDATA(K+8)
      XP(1,JB1) = R*(COSV*TDATA(K+12)-SINV*TDATA(K+9)*TDATA(K+13))
      XP(2,JB1) = R*(COSV*TDATA(K+9)+SINV*TDATA(K+12)*TDATA(K+13))
      XP(3,JB1) = R*SINV*TDATA(K+10)
      IF (TRSFER) 3,4,3
C
C     COMPUTE THE VELOCITIES FOR THE TRSFER OF ORIGIN.
    3 EX = TDATA(K+1)*TDATA(K+8)+SINV
      FX = TDATA(K+1)*TDATA(K+11)+COSV
      CFACT = ZN*TDATA(K)/(SQRTF((1.-TDATA(K+1)**2)**3))
      AX = EX*TDATA(K+12)+FX*TDATA(K+9)*TDATA(K+13)
      BX = FX*TDATA(K+12)*TDATA(K+13)-EX*TDATA(K+9)
      VEFM(1,JB1) = -AX*CFACT
      VEFM(2,JB1) = BX*CFACT
      VEFM(3,JB1) = FX*CFACT*TDATA(K+10)
    4 RETURN
      END
```

```
       REM    SUBROUTINE EXADD (A,B,C)
       REM THIS ROUTINE WILL ADD IN DOUBLE PRECISION A QUANTITY C TO THE DOUBLE
       REM PRECISION VARIABLE A+B WHERE A IS THE MOST SIGNIFICANT PART AND B IS
       REM THE LEAST SIGNIFICIANT PART.
       ENTRY   EXADD
       COMMON  -206
Q1     COMMON  1
Q2     COMMON  1
TEMP1  COMMON  1
TEMP2  COMMON  1
       HTR
       BCI     1,EXADD
       SXD     *-4,1
       SXD     *-4,2
EXADD  SXD     *-4,4
       CLA*    1,4
       FAD*    3,4
       STQ     Q1
       FAD*    2,4
       STQ     Q2
       FAD     Q1
       STQ     Q1
       STO     TEMP1
       CLA     Q1
       FAD     Q2
       STO     TEMP2
       FAD     TEMP2
       FAD     TEMP1
       STQ     Q1
       FSB     TEMP2
       STO*    1,4
       STQ     Q2
       CLA     Q1
       FAD     Q2
       STO*    2,4
       TRA     4,4
       END
```

```
                                                  SUBROUTINE ICAO
C    SUBROUTINE ICAO DETERMINES THE ATMOSPHERIC TEMPERATURE, PRESSURE, AND
C    DENSITY AS A FUNCTION OF ALTITUDE ABOVE THE EARTH IN ACCORDANCE WITH
C    THE 1962 U.S. STANDARD ATMOSPHERE (ICAO TO 20 KM.). A SHORT FAP
C    PROGRAM FOLLOWS ICAO WHICH PROVIDES A MEANS OF LOADING DATA INTO MACHINE.
C    IT MUST BE LOADED DIRECTLY AFTER ICAO.  IF THE LENGTH OF ICAO IS CHANGED,
C    THE DATA MUST BE RELOCATED.
C
C              R IS DISTANCE TO CENTER OF EARTH IN METERS.
C            ALT IS VEHICLE ALTITUDE ABOVE EARTH IN METERS.
C        TABLE H IS METERS OF ALTITUDE FROM THE EARTHS SURFACE AND IS
C                   THE ARGUMENT OF ATMOSPHERE PROPERTY TABLE.
C            ALM IS THE MEAN SLOPE OF THE TABLE H VS. TM CURVE AT TABLE H.
C            TMR IS TM AT TABLE H.
C          REF P IS THE PRESSURE IN MILLIBARS AT TABLE H.
C             TM IS THE TEMPERATURE TIMES STD. MOLECULAR WEIGHT / ACTUAL
C                   MOLECULAR WEIGHT.  DEGREES KELVIN.
C          PRESS IS PRESSURE IN  MILLIBARS.
C         DNSITY IS DENSITY IN  KILOGRAMS PER CUBIC METER.
C         HEIGHT IS EITHER GEOPOTENTIAL ALTITUDE OR GEOMETRIC ALTITUDE IN METERS.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   TABLEH(23), TMR(23), REFP(23), ALM(23) , RB(3)
C
      EQUIVALENCE
     1(A     ,C      (   11)),(ALT   ,A      (    4)),(B     ,C      (1111)),
     2(DNSITY,B      (   29)),                        (OBLATN,A      (   40)),
     3(PRESS ,B      (   33)),(R     ,B      (  102)),(RB    ,B      (  193)),
     4(RE    ,A      (   25)),(TABLT ,B      (   20)),(TM    ,B      (   34)),
     5(RESQRD,B      (    7))
      EQUIVALENCE (TABLEH(24),TMR),(TABLEH(47),ALM),(TABLEH(70),REFP)
C
      IF (OBLATN) 102,101,102
  101 ALT = R - RE
      GO TO 103
  102 ALT = R-6356783.28/SQRTF(.9933065783+.006693421685(RB(3)/R)**2)
  103 IF (ALT-90000.) 105,104,104
  104 HEIGHT = ALT
      GO TO 106
  105 HEIGHT = ALT/(1.0+ALT/6356766.)
  106 K=K
C
C     FIND THE              HEIGHT IN A TABLE OF BASE DATA. DATA ARE
C     ARRANGED IN DECENDING  ALT  WITH 21 REGIONS.  ABOVE THAT, PRESSURE AND
C     DENSITY ARE SET = 0.  TEMPERATURE IS SET TO 3000.
    1 IF (K-22) 2,6,6
    2 IF (HEIGHT-TABLEH(K+1)) 5,3,3
    3 K = K+1
      GO TO 1
    4 K = K-1
    5 IF (K) 7,7,6
    6 HINC = HEIGHT - TABLEH(K)
      IF (H INC) 4,8,8
    7 K = 1
    8 IF (ALM(K)) 9,100,9
C
```

```
C     CONTROL COMES HERE FOR NONISOTHERMAL LAYERS
    9 TM = TMR(K) + ALM(K)*H INC
      IF (ALT-90000.) 107,107,108
  107 PRESS = REFP(K)*(TMR(K)/TM)**(.0341631947/ALM(K))
      GO TO 10
  108 IF (K-KC) 109,110,109
  109 KC = K
      C1 = RE+TABLEH(K)
      C2 = TMR(K)/ALM(K)
      C3 = 1./(C1-C2)
      C4 = -.0341631947*RESQRD*C3/ALM(K)
  110 PRESS = REFP(K)*EXPF(C4*(C3*LOGF(C1*(HINC/C2+1.)/(RE+HEIGHT))-
     1        HINC/C1/(RE+HEIGHT)))
   10 DNSITY = PRESS/TM/2.87053072
      GO TO 13
C
C     CONTROL COMES HERE FOR ISOTHERMAL LAYERS
  100 IF (K-22) 11,12,12
   11 TM = TMR(K)
      PRESS = REFP(K)*EXPF(-.0341631947*HINC/TMR(K))
      GO TO 10
C
C     CONTROL COMES HERE FOR EXTREME ALTITUDES
   12 PRESS = 0.0
      DNSITY = 0.0
      TM = 3000.
   13 RETURN
      END

       REM   THIS IS THE FAP PROGRAM WHICH LOADS ICAO DATA INTO MACHINE.
       REM THE 256 IN ORG 256 WAS FOUND BY SUBTRACTING 22 FROM THE DEC LOCATION
       REM OF REF P (FROM FAP LISTING OF ICAO, THIS WAS FOUND TO BE 278).
       REM THUS, 278-22=256. DISCARD THE FIRST TWO BINARY CARDS AFTER ASSEMBLY
       REM AND PLACE REMAINING CARDS IMMEDIATELY BEHIND ICAO BINARY DECK.
       REM
       REM   A1 IS REF P(23)
       REM   A2 IS ALM(23)
       REM   A3 IS TMR(23)
       REM   A4 IS TABLE H(23)
       REM
       ORG 256
 A1    DEC 0.,1.1918E-9,3.4502E-9,1.0957E-8,4.0304E-8,1.8838E-7
       DEC 6.9604E-7,1.6852E-6,2.7926E-6,3.6943E-6,5.0617E-6,2.5217E-5
       DEC 7.3544E-5,3.0075E-4,1.6438E-3,.010377,.182099,.590005
       DEC 1.10905,8.68014,54.7487,226.320,1013.25
 A2    DEC 0.,0.,.0011,.0017,.0026,.0033,.004,.005,.007,.01,.015,.02,.01
       DEC .005,.003,0.,-.004,-.002,0.,.0028,.001,0.,-.0065
 A3    DEC 0.,2700.65,2590.65,2420.65,2160.65,1830.65,1550.65,1350.65
       DEC 1210.65,1110.65,960.65,360.65,260.65,210.65,180.65,180.65
       DEC 252.65,270.65,270.65,228.65,216.65,216.65,288.15
 A4    DEC 1E30,7E5,6E5,5E5,4E5,3E5,2.3E5,1.9E5,1.7E5,1.6E5,1.5E5,1.2E5
       DEC 1.1E5,1E5,.9E5,79000.,61000.,52000.,47000.,32000.,20000.
       DEC 11000.,0.
       END                                                                      1
                                                 SUBROUTINE NBODY
C
C     NBODY COMPUTES THE TRAJECTORY IN EITHER ORBIT ELEMENTS OR RECTANGULAR
C     COORDINATES USING THE RUNGE-KUTTA TECHNIQUE. A LOWER ORDER INTEGRATION
C     TECHNIQUE IS ALSO PERFORMED TO FACILITATE AUTOMATIC STEP SIZE CONTROL.
C     THE X,XPRIM,XDOT,XINC,ETC.  ARRAYS ARE AS FOLLOWS.
C
C     X            ORBIT ELEMENTS          RECTANGULAR COORDINATES
C
C     1            TIME                     TIME
C     2            MASS                     MASS
C     3            ECCENTRICITY             X-VELOCITY
C     4            ARGUMENT OF PERICENTER   Y-VELOCITY
C     5            ARGUMENT OF ASC. NODE    Z-VELOCITY
C     6            INCLINATION              X
C     7            MEAN ANOMALY             Y
C     8            SEMILATUS RECTUM         Z
C
C     IMODE     VARIABLES
C       1   ORBIT ELEMENTS
C       2   RECTANGULAR
C       3   RECTANGULAR TEMPORARY
C       4   EARTH SPHERICAL--CHANGE TO RECTANGULAR
C      -1   ORBIT ELEMENTS--CHANGE TO RECTANGULAR
C      -2   RECTANGULAR--CHANGE TO ORBIT ELEMENTS
C      -3   ORBIT ELEMENTS--CHANGE TO TEMPORARY RECTANGULAR
C      -4   EARTH SPHERICAL -- CHANGE TO ORBIT ELEMENTS
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1    XPRIM (100,2),      XPRIMB (100,2),      XDOTPM (100,2),
     2        X (100),          XINC (100),        OLDINC (100),
     3     XDOT (100),            RB (3),              XK (100),
     4      AMC (3),              AK (3),              AW (4),
     5   XWHOLE (6),              VX (3),             BEX (14)
C
```

```
      EQUIVALENCE
     1(A     ,C      (  11)),(A1    ,B      (  10)),(A2    ,B      (  11)),
     2(ACOEF1,B      (  12)),(ACOEF2,B      (  13)),(ACOEF3,B      (  14)),
     3(AK    ,A      (  51)),(AMC   ,B      (  87)),(AMSQRD,B      (  91)),
     4(AM    ,B      (  90)),(ASYMPT,A      (   7)),(AW    ,A      (  55)),
     5(B     ,C      (1111)),(CONSTU,A      (  32)),(DELT  ,B      (   1)),
     6(DONE  ,B      (  39)),(E2    ,B      (  18)),(EMONE ,B      (  28)),
     7(ERLIMT,A      (  14)),(ETOL  ,A      (  30)),(EXMODE,B      (  27)),
     8(GK2M  ,B      (  36)),(H2    ,B      (  15)),(IMODE ,A      (   1)),
     9(INDERR,B      (  51)),(KSUB  ,B      (  19)),(MBODYS,B      (  42))
      EQUIVALENCE
     1(NEQ   ,A      (   2)),(NSTART,B      (  24)),(OLDDEL,B      (   9)),
     2(QMAX  ,B      (  44)),(RATIO ,B      (  58)),(RB    ,B      ( 193)),
     3(REVS  ,A      (  48)),(R     ,B      ( 102)),(STEPMX,A      (  16)),
     4(STEPGO,A      (  41)),(STEPNO,A      (  42)),(TRSFER,B      (   8)),
     5(TRU   ,B      (  40)),(TTEST ,A      (  54)),(VSQRD ,B      (  96)),
     6(VX    ,B      (  92)),(XDOT  ,B      ( 501)),(XINC  ,B      ( 601)),
     7(XPRIM ,C      ( 711)),(XPRIMB,C      ( 911)),(XWHOLE,B      ( 110)),
     8(X     ,B      ( 401)),(ERLOG ,B      (  17)),(EREF  ,A      (  13)),
     9(Q     ,B      (  59)),(OUTPOT,B      ( 399))
C
C     PART 1. SET UP THE STARTING SEQUENCE FOR ERROR CONTROL AND DELAY CHECKING
C     THE ERROR UNTIL TWO STEPS ARE COMPLETED.  THE ASSIGNED GO TOS NSTART AND
C     IBEGIN CONTROL STARTING.
      NEQ = NEQ
    1 DO 2 J=1,NEQ
      XPRIM(J,2) = XPRIM(J,1)
      XPRIMB(J,2) = XPRIMB(J,1)
    2 X(J) = XPRIM(J,1)
      NSTART = 0
      TRSFER = 0.
      H2 = DELT
      DELT = DELT/2.
  220 CALL EQUATE
      IF (OUTPOT) 222,221,222
  221 CALL OUTPUT
  222 DO 3 J=1,3
      XWHOLE(J)=VX(J)
    3 XWHOLE(J+3) = RB(J)
C     CHANGE INTEGRATION VARIABLES IF IMODE IS -.
      IF (IMODE) 4,5,5
    4 CALL TESTTR
      GO TO 1
    5 CALL TESTTR
      IF (TRSFER) 1,205,1
  205 ASSIGN 21 TO NSTART
C     STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART 7A.
      IF (RB(2)) 7,6,8
    6 IF(VX(2)) 7,8,8
    7 ASSIGN 37 TO NREV1
      ASSIGN 35 TO NREV2
      GO TO 9
    8 ASSIGN 33 TO NREV1
      ASSIGN 37 TO NREV2
    9 DO 10 J=1,NEQ
      XDOTPM(J,1) = XDOT(J)
      XINC(J) = 0.
   10 CONTINUE
   11 KSUB = 1
      ASSIGN 16 TO N
C
C     PART 2.  RUNGE-KUTTA SUBINTERVAL SCHEME.  EQUATE PRODUCES THE NECCESSARY
C     DERIVATIVES XDOT(J).
   12 DO 13 J=1,NEQ
      XK(J) = XDOT(J) * DELT
      XINC(J) = XINC(J) + AW(KSUB)*XK(J)
   13 X(J) = XPRIM(J,2) + AK(KSUB)*XK(J)
   14 CALL EQUATE
   15 GO TO N,(16,17,18,20)
C
C     PART 3. SUBINTERVALS 2, 3, AND 4, TO STATEMENT 19 FINISH A
C     RUNGE-KUTTA STEP AND INCREMENT XPRIM(J,2) IN DOUBLE PRECISION.
   16 KSUB = 2
      ASSIGN 17 TO N
      GO TO 12
   17 KSUB = 3
      ASSIGN 18 TO N
      GO TO 12
   18 DO 19 J=1,NEQ
      XINC(J) = XINC(J) + AW(4) *XDOT(J) * DELT
  180 CALL EXADD(XPRIM(J,2), XPRIMB(J,2), XINC(J))
      X(J) = XPRIM(J,2)
   19 CONTINUE
C
C     PART 4. BEGIN A NEW RUNGA-KUTTA STEP.  THIS ALSO GIVES DERIVATIVES
C     FOR THE LOWER ORDER INTEGRATION CHECK.
      ASSIGN 20 TO N
      GO TO 14
   20 GO TO NSTART,(27,23,21)
C
C     PART 5.  STARTING PHASE PROGRAM.
C     PART 5A. THIS SECTION COMPLETES THE FIRST STEP OF STARTING PHASE.
   21 ASSIGN 23 TO NSTART
      DO 22 J=1,NEQ
      OLDINC(J)=XINC(J)
      XINC(J)=0.
      XDOTPM(J,2) = XDOT(J)
   22 CONTINUE
      GO TO 11
C
```

```
C     PART 5B. MAX ERROR TEST--STARTING ONLY--CHECK THE MAX ERROR AND
C     EITHER ENTER RUNNING MODE OR REPEAT START WITH SMALLER STEP.
   23 DO 24 J=2,NEQ
   24 XINC(J) =(XINC(J)+OLDINC(J))*3.-(XDOTPM(J,1)+XDOTPM(J,2)*4.
     1+XDOT(J))*DELT
  240 CALL ERRORZ
   25 IF(E2-ERLIMT) 26,26,56
   26 ASSIGN 27 TO NSTART
      ASSIGN 11 TO IBEGIN
      A1 = A2
      GO TO 31
C
C     PART 6.  RUNNING PHASE PROGRAM.
C     PART 6A. CHECK THE INTEGRATION BY INTEGRATING OVER THE LAST
C     RUNGE KUTTA STEP BUT USE DOTS FOR LAST TWO INTERVALS, OLDDEL
C     AND DELT RESPECTIVELY.  STATEMENT 28 IS THE LOWER INTEGRATION
C     MINUS RUNGE-KUTTA INCREMENTS.  ERRORZ COMPUTES THE MAXIMUM RELATIVE
C     ERROR AND STATEMENT 29 TESTS THIS ERROR AGAINST THE LIMIT VALUE.
   27 RATIO = DELT/OLDDEL
      HFACT=DELT/(1.+RATIO)
      ACOEF1=-RATIO*RATIO*HFACT
      ACOEF2=RATIO*(DELT+3.*OLDDEL)
      ACOEF3=DELT+DELT+HFACT
      DO 28 J=2,NEQ
   28 XINC(J) = ACOEF1*XDOTPM(J,1)+ACOEF2*XDOTPM(J,2)-6.*XINC(J)
     1+ACOEF3*XDOT(J)
  280 CALL ERRORZ
   29 IF (E2-ERLIMT) 30,30,57
C
C     PART 7A. LAST POINT OKAY. COUNT THE REVOLUTIONS PAST THE X-AXIS.
C     A STEP GREATER THAN 1/2 REV. MAY FAIL TO ADD IN.
   30 H2 = DELT
   31 QMAX = MAX1F(Q,QMAX)
      IF(RB(2)) 32,34,34
   32 GO TO NREV1, (37,33)
   33 ASSIGN 37 TO NREV1
      ASSIGN 35 TO NREV2
      GO TO 37
   34 GO TO NREV2, (37,35)
   35 ASSIGN 33 TO NREV1
      ASSIGN 37 TO NREV2
   36 REVS = REVS + 1.
   37 IF (XABSF(IMODE)-1) 42,38,42
C
C     PART 7B. IN ORBIT ELEMENTS. ADJUST ARGUMENT OF PERICENTER AND MEAN ANOMALY
C     TO + OR - PI TO MAINTAIN ACCURACY IN SIN-COS ROUTINES.
   38 IF (EMONE) 39,42,42
   39 DO 41 J=4,7,3
      ADJ2=INTF(XPRIM(J,2)/6.28318532+SIGNF(.5,XPRIM(J,2)))
      IF (ADJ2) 40,41,40
   40 ADJ3 = -ADJ2*6.28125
  400 CALL EXADD(XPRIM(J,2),XPRIMB(J,2),ADJ3)
      ADJ3=-ADJ2*.0019353072
  401 CALL EXADD(XPRIM(J,2),XPRIMB(J,2),ADJ3)
   41 CONTINUE
C
C     PART 7C. ADVANCE THE REMAINING PARAMETERS, FIND NEW STEP SIZE,
C     AND TEST FOR AN ORIGIN TRANSLATION.
   42 DO 43 K=1,3
      XWHOLE(K)=VX(K)
   43 XWHOLE(K+3) = RB(K)
      DO 44 J=1,NEQ
      XDOTPM(J,1) = XDOTPM(J,2)
      XDOTPM(J,2) = XDOT(J)
      XPRIM(J,1) = XPRIM(J,2)
      XPRIMB(J,1) = XPRIMB(J,2)
      XINC(J) = 0.
   44 CONTINUE
      OLDDEL = DELT
   45 CALL STEP
      IF (DONE) 67,450,67
  450 IF (NSTART) 451,1,451
  451 IF (MBODYS) 46,47,46
   46 CALL TESTTR
      IF (TRSFER) 1,47,1
   47 IF (XABSF(IMODE)-3) 11,48,11
C
C     PART 7D. IF IN TEMPORARY RECTANGULAR COORDINATES, TEST FOR RETURN
C     TO ORBIT ELEMENTS.  FIRST, E IS FOUND.  IF TIME HAS NOT ADVANCED
C     SUFFICIENTLY, INTEGRATION CONTINUES IN RECTANGULAR VARIABLES (STATE. 48).
C     STATEMENT 49 DETERMINES IF KEPLERS EQUATION CAUSED IMODE = 3.  IF NOT,
C     AN E CLOSE TO 1 CHECK IS MADE IN STATEMENT 50.  IF IT DID, RECTANGULAR
C     VARIABLES WILL BE USED IF THE LIMIT IS TOO SMALL (STATEMENT 52), OR
C     IF E IS 5 OR GREATER (STATEMENT 53) OR IF THE PATH LIES CLOSE TO AN
C     ASYMPTOTE (STATEMENT 55).
   48 CALL CONVT1 (VX,AMC)
      EXMODE=SQRTF(1.+AMSQRD/GK2M*(VSQRD/GK2M-2./R))
      EMONE=EXMODE-1.
      IF ((XPRIM(1)-TTEST)*DELT) 11,11,49
   49 IF (ASYMPT) 51,50,51
   50 IF (ETOL-ABSF(EMONE)) 55,11,11
   51 IF(EMONE) 55,55,52
   52 IF(CONSTU-1.E-7) 11,53,53
   53 IF (EXMODE-5.) 54,11,11
   54 CALL CONVT2
      IF (ABSF(TRU)-2.2/SQRTF(EXMODE)) 55,55,11
   55 ASYMPT = 0.0
      IMODE=-2
  555 CALL TESTTR
      GO TO 1
C
```

```
C     PART 8. COMES HERE WHEN ERROR TEST FAILED--BOTH STARTING AND RUN.
C     RETRIEVE OLD POINT AND RECOMPUTE WITH SMALLER INTERVAL.
C     IF TWO CONSECUTIVE TRYS FAIL (STATEMENT 59) THE STARTING SEQUENCE OCCURS.
   56 ASSIGN 1 TO IBEGIN
   57 DO 58 J=1,NEQ
      XPRIM(J,2) = XPRIM(J,1)
      XPRIMB(J,2) = XPRIMB(J,1)
      XDOT(J)=XDOTPM(J,2)
      XINC(J)= 0.
   58 CONTINUE
      STEPNO=STEPNO+1.
      H2 = DELT
      DELT=SIGNF(EXPF((ERLOG-A2)/5.),DELT)
      A2 =A1
   59 IF (FAIL-STEPGO) 60,61,60
   60 FAIL = STEPGO
      GO TO IBEGIN, (11,1)
   61 ASSIGN 1 TO IBEGIN
      IF (STEPNO + STEPGO - STEPMX) 62,62,45
   62 GO TO IBEGIN, (11,1)
C
C     PART 10. PRINT OUT THE ERROR INFO. IF EREF HAS A - SIGN.  THEN RETURN.
   67 IF (EREF) 68,72,72
   68 WRITE OUTPUT TAPE 6,70
      REWIND 4
      DO 69 I=1,INDERR
      READ TAPE 4, BEX
   69 WRITE OUTPUT TAPE 6,71,BEX
      REWIND 4
      INDERR = 0
   70 FORMAT(7H1  STEP,6X,4HTIME,6X,4HDELT,7X,2HA2,8X,2HE2,7X,4HMASS,6X,
     14HE,VX,4X,8HOMEGA,VY,2X,8HNODES,VZ,3X,6HINCL,X,5X,4HMA,Y,6X,3HP,Z,
     24X,1HK//)
   71 FORMAT(F5.,1H+F3.,1P11G10.2,I2)
   72 RETURN
      END
                                                  SUBROUTINE ORDER
C     THIS ROUTINE TAKES THE BODY LIST READ FROM CARDS AND SORTS THEM IN
C     ORDER SO THAT THE DISTANCE FROM THE REFERENCE TO EACH BODY IS
C     DEPENDENT UPON ALREADY COMPUTED DISTANCES ONLY.
C
C     ELLIPSE DATA ARE READ INTO A BLOCK OF 120 STORES RESERVED FOR
C     TEN ELLIPSES.  ONE ELLIPSE IS READ INTO A 12 STORE BLOCK.
C     THE SINES AND COSINES OF THE 3 ANGLES ARE COMPUTED AND STORED
C     IN THE TDATA ARRAY ALONG WITH THE REST OF THE ELLIPSE DATA.
C     A BLOCK IS ARRANGED AS FOLLOWS'
C
C     (1) = NAME OF BODY IN BCD,ONLY 6 CHARACTERS.
C     (2) = NAME OF REFERENCE BODY IN BCD,SAME RESTRICTION.
C     (3) = MASS OF THE BODY IN SUN MASS UNITS.
C     (4) = RADUIS INSIDE OF WHICH COORDINATES WILL BE TRANSLATED TO THIS BODY.
C     (5) = SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
C     (6) = ECCENTRICITY OF THE ORBIT.
C     (8) = LONGITUDE OF ASCENDING NODE.
C     (7) = ARGUMENT OF PERIHELION.
C     (9) = INCLINATION OF THE ORBIT.
C     (10)= PERIGEE PASSAGE JULIAN DAY.
C     (11)= PERIGEE PASSAGE FRACTION OF DAY.
C     (12)= PERIOD OF THE ELLIPSE IN MEAN SOLAR DAYS.
C
C     AMASS  = MASS OF EACH BODY, SUN MASSES.  ORDER OF PNAME.
C     BMASS  = SELECTED FROM AMASS. CORRESPONDS TO BNAME LIST.
C     BNAME  = THE ORDERED LIST OF BCD BODY NAMES. CAN BE USED IN OUTPUT.COMMON.
C     BODYCD = THE ORIGINAL BCD NAMES READ FROM CARDS.
C     BODY L = THE LIST OF BCD BODY NAMES WITH THE REFERENCE BODY AT TOP.
C              INITIALLY EQUAL TO BODY CARD LIST (BODYCD).
C     IBODY  = ARRAY OF SUBSCRIPTS. WHEN A DISTANCE IS FOUND FROM EPHEMERIS, IT
C              MAY BE ADDED (OR SUBTRACTED) FROM THE BODY POSITION GIVEN BY
C              XP(IBODY) TO OBTAIN THE POSITION OF THE PRESENT BODY.  COMMON.
C     KZERO  = COUNT OF ZERO REFERENCES. THERE MUST BE ONE AND ONLY ONE ZERO.
C              FROM LOCATION IN BNAME LIST. NOT IN COMMON.
C     MANE   = ARRAY OF SUBSCRIPTS.  INVERSE OF NAME.  GIVES NEW LOCATION OF
C              BNAME LIST IN TERMS OF BODYL.  NOT IN COMMON.
C     NBODYS = COUNTED INTERNALY. TOTAL NUMBER OF  BODYS.
C     MBODYS = COMPUTED INTERNALY. TOTAL NUMBER OF EPHEMERIDES (NBODYS-1).
C     NAME   = ARRAY OF SUBSCRIPTS.  GIVES OLD LOCATION OF NAMES IN BODYL
C     NEFMRS = ARRAY OF SUBSCRIPTS. GIVES LOCATION OF BODY IN PNAME LIST
C              IN TERMS OF THE EFMRS LIST.  STORED IN COMMON.
C     NREFER = ARRAY OF SUBSCRIPTS. LOCATES THE REFERENCE BODY IN BODYL.
C              ORDER OF THE ARRAY CORRESPONDS TO BODYL.  NOT IN COMMON.
C     NNREFR = ARRAY OF SUBSCRIPTS. LIKE NREFER BUT REFERS AND CORRESPONDS TO
C              BNAME LIST.  NOT IN COMMON.
C     PNAME  = A PERMANENT LIST OF BCD BODY NAMES.  1 WORD EACH (6 CHARACTERS
C              MAX).  USED TO IDENTIFY MASS, REFERENCE NAMES, ETC.  THE LIST IS
C              A MAXIMUM OF 30 NAMES.  PRECISION TAPE NAMES ARE FROM 1 TO 20,
C              ELLIPTIC NAMES ARE FROM 21 TO 30.
C     REFER  = A PERMANENT LIST OF BCD BODYS THAT ARE THE REFERENCES OF
C              DISTANCES GIVEN IN EPHERMERIDES (TAPES OR ELLIPSE). CORRESPONDS
C              TO PNAME LIST.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1    AMASS (30),          BMASS (8),           BNAME (8),
     2    BODYL (8),           EFMRS (7),           IBODY (8),
     3     MANE (8),            NAME (8),          NEFMRS (8),
     3   NEFMRT (8),          NNREFR (8),          BODYCD (8),
     4   NREFER (8),           PNAME (30),         RBCRIT (7),
     5    RCRIT (30),          REFER (30),          TDATA (18,7),
     6     TDEL (7),             TIM (7),           ELIPS (120),
     7     NDUD (9),           XPRIM (200)
C
```

```
      EQUIVALENCE
     1(A     ,C      (  11)),(AMASS ,A      ( 347)),(ATMN  ,A      (  21)),
     2(AU    ,A      (  29)),(B     ,C      (1111)),(BMASS ,B      ( 137)),
     3(BNAME ,B      ( 122)),(BODYCD,A      ( 143)),(BODYL ,B      ( 153)),
     4(EFMRS ,B      ( 130)),(ELIPS ,A      ( 167)),(FILE  ,B      (  22)),
     5(GK2M  ,B      (  36)),(GKM   ,B      (  37)),(IBODY ,B      ( 177)),
     6(MBODYS,B      (  42)),(NBODYS,B      (  41)),(NEFMRS,B      ( 185)),
     7(PNAME ,A      ( 287)),(RATMOS,B      (  23)),(RATM  ,A      (  22)),
     8(RBCRIT,B      ( 145)),(RCRIT ,A      ( 377)),(REFER ,A      ( 317)),
     9(RE    ,A      (  25)),(RESQRD,B      (   7)),(REVOLV,B      (  21))
      EQUIVALENCE
     1(ROTATE,A      (  39)),(SPD   ,A      (  44)),(SQRDK1,A      (  47)),
     2(SQRDK ,B      (  35)),(TDATA ,B      ( 265)),(TDEL  ,B      ( 170)),
     3(TFILE ,A      (   6)),(TIM   ,B      ( 163)),(TRSFER,B      (   8)),
     4(XPRIM ,C      ( 711)),(OUTPOT,B      ( 399))
      EQUIVALENCE (MANE(1),NOUD(2))
C
C     THIS SECTION SEES WHAT ELLIPSE DATA WAS READ FROM CARDS AND PUTS THE
C     NAMES IN PLACE SO THAT DATA  WILL BE USED IF NEEDED. ELLIPSE DATA HAS
C     PRIORITY OVER TAPE DATA BECAUSE LAST DATA IN LIST IS THAT ACTUALLY USED.
C     FUNCTION COMPARF(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C
B     COMPARF(A,B) = (A+B)*(-(A*B))
      DO 2 K=1,120,12
      IF(ELIPS(K)) 1,2,1
    1 KOUNT = (K-1)/12+21
      PNAME(KOUNT)  = ELIPS(K)
      REFER(KOUNT)  = ELIPS(K+1)
      AMASS(KOUNT)  = ELIPS(K+2)
      RCRIT(KOUNT)  = ELIPS(K+3)
    2 CONTINUE
C
C     PART 0. THROW AWAY BLANKS AND DUPLICATES IN BNAME LIST.
C             ALSO COUNT THE BODIES.
      IF (TRSFER) 4,3,4
    3 BNAME(1) = BODYCD(1)
    4 DO 5 K=1,8
    5 BNAME(K+1)= BODYCD(K)
      L = 1
      BODYL(0) = 0.
      DO 8 I=1,9
      BODYL(I) = 0.
      DO 6 K=1,L
      IF (COMPARF (BNAME(I), BODYL(K-1))) 6,7,6
    6 CONTINUE
      BODYL(L) = BNAME(I)
      L = L+1
    7 BNAME(I) = 0.
    8 CONTINUE
      NBODYS = L-1
      MBODYS = NBODYS-1
C
C     PART 1. FIND THE REFERENCE BODY FOR EACH BODY IN THE LIST OF BODYS
C             READ FROM CARDS.  CLEAR NREFER AND BNAME.
      DO 13 KL=1,NBODYS
      NREFER(KL) = 0
      NEFMRT(KL) =0
      BNAME (KL) = 0.
      DO 12   KP= 1,30
      IF (COMPARF(BODYL(KL),PNAME(KP))) 12,9,12
    9 NEFMRT(KL) = KP
      DO   11   KR = 1,8
      IF (COMPARF(REFER(KP),BODYL(KR))) 11,10,11
   10 NREFER(KL) = KR
   11 CONTINUE
   12 CONTINUE
   13 CONTINUE
C     PART 2 . COUNTS 0  REFERENCES AND SAVES TEMPORARY SET OF INDEXS.
   14 IF (NBODYS) 24,24,15
   15 KZEROS = 0
      MISPEL = 0
      DO  20  K = 1,NBODYS
      NNREFR(K) = NREFER(K)
   16 IF (NEFMRT(K)) 18,17,18
   17 MISPEL = MISPEL + 1
   18 IF(NREFER(K)) 20,19,20
   19 KZEROS = KZEROS + 1
   20 CONTINUE
   21 IF  (KZEROS- 1)  24,22,24
   22 IF (MISPEL) 24,23,24
   23 IF (NBODYS-8) 28,28,24
C
C     PART 3 . REPORTS ERRORS IN BODY LIST.
   24 WRITE OUTPUT TAPE 6,25 ,NBODYS,MISPEL,KZEROS,(BODYL(K),K=1,NBODYS)
      WRITE OUTPUT TAPE 6,26 ,(NREFER(K),K=1,NBODYS)
      WRITE OUTPUT TAPE 6,27 ,(K,PNAME(K),REFER(K),K=1,30)
   25 FORMAT (26HOGOOFY BODY LIST (NBODYS =I2,13H,  MISSPELL =I2,
     1 11H,  KZEROS =I2,1H)/11HOBODYLIST =8(3X,A6))
   26 FORMAT (11H   NREFER =I6,7I9)
   27 FORMAT (/5(3H  K3X,4HBODY4X,5HREFER5X,)/5(I3,2X,A6,2X,A6,5X))
      CALL EXIT
C
C     PART 4.  TRACES OUT ..REFERENCE TO BODY.. RELATIONSHIPS
   28 KK = 2
      KN = 1
      NAME(1) = 1
   29 IF (NREFER(KN))  24,31,30
   30 NAME(KK) = NNREFR(KN)
      NNREFR(KN) = 0
      KN = NAME(KK)
      KK = KK + 1
      GO TO 29
C
```

```
C     PART 5.  TRACES OUT  ..BODY TO REFERENCE..  RELATIONSHIP
   31 DO  34  KN = 1,NBODYS
      DO  34 K = 1,NBODYS
   32 IF (NNREFR(K) - NAME(KN)) 34,33,34
   33 NAME(KK) = K
      KK = KK + 1
   34 CONTINUE
C
C     PART 6. INVERTS NAME TO MANE,STORES BNAME, BMASS, RBCRIT,  AND A
C             TEMPORARY NEFMRS.
      DO 35 K = 1,NBODYS
      N = NAME(K)
      MANE(N) = K
      NEF = NEFMRT(N)
      BNAME(K) = PNAME(NEF)
      BMASS(K) = AMASS(NEF)
      RBCRIT(K) = RCRIT(NEF)
      NEFMRS(K) = NEF
   35 CONTINUE
C
C     PART 7. FINDS NNREFR REFERENCE FOR BNAME LIST , ALSO TEMP. IBODY
      DO 36  K = 1, NBODYS
      N = NAME(K)
      NRF = NREFER(N)
      NNREFR(K) = MANE(NRF)
   36 IBODY(K) = MANE(NRF)
C
C     PART 8 . FINDS IBODY FOR BACKWARD REFERENCE.
      DO 39 K=1,8
   37 IF(NNREFR(K)) 24,40,38
   38 N = NNREFR(K)
      IBODY(N) = -K
   39 CONTINUE
C     IBODY LIST IS COMPLETE.
C
C     PART 9 . WRITES OUT EPHEMERIS LIST TO BE USED IN STORING DATA AND
C              MAKES FINAL NEFMRS LIST.
   40 KK = 1
      DO 43 K=1,NBODYS
   41 IF(NNREFR(K)) 42,43,42
   42 EFMRS(KK) = BNAME(K)
      NEFMRS(KK) = NEFMRS(K)
      KK = KK + 1
   43 CONTINUE
      NEFMRS(NBODYS) = 0
C
C     PART 10. SAVES ELLIPSE DATA
      FILE = 0.
      IF (MBODYS) 430,480,430
  430 DO 48 K=1,MBODYS
   44 IF(NEFMRS(K)-20) 47,47,45
   45 DO 46 J=5,12
      L= (NEFMRS(K) - 21) * 12 +J
   46 TDATA(J-4,K) = ELIPS(L)
      DO 50 J=7,9
      L = (NEFMRS(K)-21)*12+J
      TDATA(J+2,K) = SINF(ELIPS(L))
   50 TDATA(J+5,K) = COSF(ELIPS(L))
      GO TO 48
C
C     PART 10A.  LOADS A FALSE (VERY EARLY) TAPE TIME TO FORCE TAPE
C                READING BY THE EPHMRS ROUTINE. FILE = 0 UNLESS TAPE IS USED.
   47 TDEL(K) = 0
      TIM(K) = 2400000.5
      FILE = 10.
   48 CONTINUE
C
C     PART 11. COMPUTE GRAVITATIONAL CONSTANTS. 1.9866 E+30 = KILOGRAMS/SUN MASS
C     IF ORIGIN BODY HAS AN ATMOSPHERE, SET ROTATION RATE AND ATMOSPHERE RADIUS.
C     POSITION THE EPHEMERIDES TAPE AT THE BEGINNING OF THE CORRECT EPHEMERIS
C     BY MATCHING THE EPHEMERIS NUMBER READ FROM TAPE (FILE) WITH THE DESIRED
C     EPHEMERIS NUMBER (TFILE).
C
  480 RESQRD = RE**2
      SQRDK = SQRDK1*AU**3/SPD**2
      GK2M = SQRDK*(BMASS(1) + XPRIM(2)/1.9866 E30)
      GKM = SQRTF(GK2M)
      REVOLV = 0.
      RATMOS = 0.
      IF (ATMN-BNAME(1)) 51,49,51
   49 REVOLV = ROTATE
      RATMOS  = RATM
   51 IF (FILE) 56,56,52
   52 CALL BSFILE(3)
   53 READ TAPE 3, FILE
      IF (FILE-TFILE) 54,56,55
   54 CALL SKFILE(3)
      GO TO 53
   55 BACKSPACE 3
      BACKSPACE 3
      GO TO 52
C
C     PART 12.    WRITES THE BNAME LIST ON TAPE 6 .
   56 IF (OUTPOT) 58,59,58
   59 WRITE OUTPUT TAPE 6,57,BNAME(1),(BNAME(K),K=2,NBODYS)
   57 FORMAT (19HOREFERENCE BODY IS A6,5X,23H  PERTURBING BODIES ARE
     1  7(2X,A6))
   58 RETURN
      END
```

```
                                                 SUBROUTINE OBLATE
C     THIS SUBROUTINE COMPUTES THE OBLATENESS ACCELERATIONS (OBLAT) DUE TO AN
C     AXIALLY SYMMETRIC EARTH.  THE 2ND, 3RD, AND 4TH SPHERICAL HARMONIC COEFF.
C     ARE OBLATJ, OBLATH, AND OBLATO RESPECTIVELY.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   RB(3), OBLAT(3)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(B     ,C      (1111)),(GK2M  ,B      (  36)),
     2(OBLATJ,A      (  26)),(OBLATO,A      (  27)),(OBLATH,A      (  28)),
     3(OBLAT ,B      (  75)),(R     ,B      ( 102)),(RB    ,B      ( 193)),
     4(RE    ,A      (  25)),(RSQRD ,B      (  45)),(RESQRD,B      (   7))
C
      AA = RB(3)/R
      AB = AA*AA
      IF (ABSF(AA)-1.E-6) 1,1,2
    1 AA=0.
      AB=0.
    2 AC = RESQRD/RSQRD
      AD = GK2M/RSQRD/R*AC
      AE = OBLATJ*AD
      AF = OBLATH*AD*RE/R
      AG = OBLATO*AD*AC
      AH = AE*(5.AB-1.)+AF*(7.AB-3.)*AA+AG*(6.AB-9.AB**2-0.4285714286)
      OBLAT(1) = AH*RB(1)
      OBLAT(2) = AH*RB(2)
      OBLAT(3) = (AH-2.AE+AG*(4.AB-1.714285714))*RB(3)-AF*(3.AB-0.6)*R
    3 RETURN
      END
```

```
                                                 SUBROUTINE OUTPUT
C      ENTS AND RECTANGULAR COORDINATES ARE OUTPUTTED.  IF THE OBJECT IS NOT WITH
C      THIS IS THE ROUTINE WHICH FORMS THE BASIC DATA OUTPUT.  BOTH ORBIT ELEM-
C      IN AN ATMOSPHERE (PRESS=0.), ONE LINE OF DATA IS DELETED.  LIKEWISE,
C      ONLY THOSE PERTURBING BODIES PRESENT HAVE THEIR DISTANCES OUTPUTTED.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1     R     (8),          ORBELS (6),           VATM (3),
     2    BNAME (8),              RB(3,8),           DIRCOS(3,8),
     3    XPRIM (200),          RAMC (5)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(ALPHA ,A      (  49)),(ALT   ,A      (   4)),
     2(AMC   ,B      (  87)),(AM    ,B      (  90)),(AREA  ,B      (   6)),
     3(BNAME ,B      ( 122)),(B     ,C      (1111)),(CD    ,A      ( 165)),
     4(CL    ,A      ( 164)),(COSALF,B      (  48)),(COSTRU,B      (  53)),
     5(DTOFFJ,A      (  23)),(H2    ,B      (  15)),(IMODE ,A      (   1)),
     6(MBODYS,B      (  42)),(NBODYS,B      (  41)),(ORBELS,B      ( 116)),
     7(PRESS ,B      (  33)),(P     ,B      (  84)),(PSI   ,B      (  30)),
     8(PSIR  ,B      ( 398)),(PUSH  ,A      ( 166)),(Q     ,B      (  59)),
     9(RAMC  ,B      ( 393)),(RB    ,B      ( 193)),(REVS  ,A      (  48))
      EQUIVALENCE
     1(R     ,B      ( 102)),(SINALF,B      (  46)),(SINTRU,B      (  52)),
     2(STEPGO,A      (  41)),(STEPNO,A      (  42)),(TABLT ,B      (  20)),
     3(TRU   ,B      (  40)),(VATM  ,B      (  97)),(VQ    ,B      ( 100)),
     4(V     ,B      (  95)),(VX    ,B      (  92)),(VY    ,B      (  93)),
     5(VZ    ,B      (  94)),(XPRIM ,C      ( 711)),(OUTPOT,B      ( 399))
C
      DAYJ=(DTOFFJ-2.4E6)+TABLT
      ALPHA1 = ALPHA*57.29577951
      REV = REVS+ARCTAN(-RB(2),-RB(1))/6.28318532 + .5
   16 CALL CONVT1(VX,AMC)
      IMODE=IMODE
      GO TO (2,1,1),IMODE
    1 CODE=6HRECTAN
   18 CALL CONVT 2
      GO TO 4
    2 DO 3 K=1,6
    3 ORBELS(K) = XPRIM(K+2)
      CODE=5HORBIT
      TRU=ARCTAN(SINTRU,COSTRU)
    4 PSI = ATANF((RB(1)*VX+RB(2)*VY+RB(3)*VZ)/AM)57.2957795
      IF (OUTPOT) 19,6,19
    6 WRITE OUTPUT TAPE 6, 11,STEPGO,STEPNO,ORBELS(1),ORBELS(2),V,R(1),B
     1NAME(1),CODE,IMODE,XPRIM(1),ORBELS(6),TRU,VX,RB(1),XPRIM(2),DAYJ,O
     2RBELS(5),ORBELS(3),VY,RB(2),REV,ALPHA1,PSI,ORBELS(4),VZ,RB(3),H2
C
C     IF WITHIN AN ATMOSPHERE COMPUTE DRAG, LIFT, G, ETC., AND PRINT EXTRA LINE.
   19 IF (PRESS) 5,7,5
    5 XIFT = Q*AREA*CL
      DRAG = Q*AREA*CD
      G = (PUSH-DRAG*COSALF+XIFT*SINALF)/XPRIM(2)/9.80665
   17 CALL CONVT1(VATM,RAMC)
      PSIR = ATANF((RB(1)*VATM(1)+RB(2)*VATM(2)+RB(3)*VATM(3))/RAMC(4))*
     1 57.2957795
      IF (OUTPOT) 7,14,7
   14 WRITE OUTPUT TAPE 6,12,ALT,PSIR,DRAG,VQ,G,PUSH
C
```

```
C     IF PERTURBATING BODIES ARE PRESENT, FIND THEIR DISTANCES AND PRINT THEM.
    7 IF(MBODYS) 8,10,8
    8 DO 9 J=2,NBODYS
      DO 9 K=1,3
    9 DIRCOS(K,J) = -RB(K,J)/R(J)
      IF (OUTPOT) 10,15,10
   15 WRITE OUTPUT TAPE 6,13,
     1(BNAME(J),R(J),DIRCOS(1,J),DIRCOS(2,J),DIRCOS(3,J),J=2,NBODYS)
   10 RETURN
   11 FORMAT(6H0STEP=F6.,2H +F4.,3X,13HECCENTRICITY=1PG15.8,7H OMEGA=G15
     1.8,4H  V=G15.8,3H R=G15.8,7H REFER=A6,1X,A6,I2/6H TIME=1PG14.7,14
     2H SEMILATUS R.=G15.8,7H TRU A=G15.8,4H VX=G15.8,3H X=G15.8,7H RMAS
     3S=G15.8/9H JDAY= 240PF10.4,15H  MEAN ANOMALY=1PG15.8,7H  NODE=G15.
     48,4H VY=G15.8,3H Y=G15.8,7H REVS.=G15.8/6H ALFA=G14.7,14H   PATH A
     5NGLE=G15.8,7H  INCL=G15.8,4H VZ=G15.8,3H Z=G15.8,7H  DELT=G15.8)
   12 FORMAT(6H ALT.=1PG14.7,14H R PATH ANGLE=G15.8,7H  DRAG=G15.8,4H VR
     1=G15.8,3H G=G15.8,7H  PUSH=G15.8)
   13 FORMAT(2(1X,A6,3H R=1PG14.7,0P3F10.6,11X))
      END

                                          FUNCTION QUAD (X,IC)
C     THIS ROUTINE COMPUTES ANY VARIABLE, QUAD, AS A QUADRATIC FUNCTION OF X.
C     QUAD = A + BX + CXX.  THERE MAY BE SEVERAL SETS OF COEFFIENTS, EACH SET
C     BELONGING TO A PARTICULAR REGION OF X.  THE COEFN ARRAY IS ARRANGED AS --
C        X1,A1,B1,C1,X2,A2,B2,C2,X3,A3,B3,C3,X4, .............
C     WHERE  A1,B1,C1 ARE THE COEFFIENTS TO BE USED FOR X BETWEEN X1 AND X2,ETC.
C     AND X1 IS LESS THAN X2, X2 IS LESS THAN X3, X3 IS LESS THAN X4, ETC.
C     IC IDENTIFIES WHICH DEPENDENT VARIABLE, QUAD, IS BEING SOUGHT.
C     ICC(IC) DEFINE THE STARTING LOCATIONS IN THE COEFN ARRAY FOR VARIABLES X.

      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   COEFN(190), ICC(5)
C
      EQUIVALENCE
     1(A     ,C     (  11)),(B     ,C     (1111)),(COEFN ,A     ( 407)),
     2(ICC   ,A     ( 153))
C
      I=ICC(IC)
    1 IF (X-COEFN(I)) 2,3,3
    2 I = I-4
      GO TO 1
    3 IF(X-COEFN(I+4)) 5,5,4
    4 I = I+4
      GO TO 3
    5 QUAD = COEFN(I+1)+X*(COEFN(I+2)+X*COEFN(I+3))
      ICC(IC)=I
      RETURN
      END



                                                SUBROUTINE STAGE
C     THIS ROUTINE IS CALLED TO PREPARE DATA FOR USE IN NBODY.  STAGE DATA IS
C     TAKEN FROM PERMINENT STORES AND LOADED INTO WORKING STORES.  STAGE DATA
C     MAY BE SET ASIDE FOR LATER USE (IF ON NSAVE-NSTAGE).  WHEN IMODE IS 4,
C     CONVERSION FROM EARTH-SPHERICAL TO RECTANGULAR OR ORBIT ELEMENTS TAKES
C     PLACE IN TUDES.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1XPRIM(200),XPRIMB(200),TB(10),FLOW1(10),AEXIT1(10),SIMP1(10),
     2AREA1(10),DELT1(10),IDENT(10),TABLE(200),RMASS1(10),D(600)
C
      EQUIVALENCE
     1(A     ,C     (  11)),(AEXIT ,B     (   3)),(AEXIT1,A     ( 103)),
     2(AREA1 ,A     ( 113)),(AREA  ,B     (   6)),(B     ,C     (1111)),
     3(DELT1 ,A     ( 133)),(DELT  ,B     (   1)),(D     ,C     (2111)),
     4(DEL   ,A     (  43)),(DELMAX,A     (  19)),(DONE  ,B     (  39)),
     5(EREF  ,A     (  13)),(ERLOG ,B     (  17)),(EXITA ,B     ( 392)),
     6(FLOW  ,B     (   5)),(FLOW1 ,A     (  83)),(IDENT ,A     ( 123)),
     7(IMODE ,A     (   1)),(LSTAGE,A     (  38)),(MODOUT,A     (  20)),
     8(NCASE ,C     (   1)),(NCASES,A     ( 600)),(NSAVE ,C     (   4)),
     9(NSTAGE,A     (   3)),(PUSHO ,B     ( 391)),(RMASS1,A     (  73))
      EQUIVALENCE
     1(SIMP1 ,A     (  93)),(SIMP  ,B     (   2)),(TB    ,A     (  63)),
     2(TABLE ,C     (1911)),(TKICK ,A     (  15)),(TMAX  ,B     (   4)),
     3(TTOL  ,A     (  45)),(XPRIMB,C     ( 911)),(XPRIM ,C     ( 711)),
     4(RETURN,B     ( 400)),(OUTPOT,B     ( 399))
C
```

```
C     PART 0. SAVE INITIAL DATA IF DESIRED.  LOAD STAGE DATA INTO WORKING
C     STORAGE.  ALLOW ADDITIONAL STAGE INPUT.
      IF (DEL) 100,99,100
   99 DEL = DELMAX-TKICK
  100 IF (NSAVE-NSTAGE) 103,101,103
  101 NCASES = NCASE
      DO 102 J=1,1100
  102 D(J) = A(J)
      IF (OUTPOT) 103,97,103
   97 WRITE OUTPUT TAPE 6,98,NSTAGE,NCASE
   98 FORMAT(29H SAVED INITIAL DATA FOR STAGEI2,8H OF CASEI4,1H.)
  103 NSTAGE = NSTAGE
      TMAX = XPRIM(1)+TB(NSTAGE)
      XPRIMB(2) = 0.
      IF (RMASS1(NSTAGE)) 117,117,118
  117 XPRIM(2) = XPRIM(2)+RMASS1(NSTAGE)
      GO TO 119
  118 XPRIM(2) = RMASS1(NSTAGE)
  119 FLOW = FLOW1(NSTAGE)
      SIMP = SIMP1(NSTAGE)
      AEXIT = AEXIT1(NSTAGE)
      AREA = AREA1(NSTAGE)
      DELT = DELT1(NSTAGE)
      ID = IDENT(NSTAGE)
    3 CALL INPUT (ID,C,TABLE)
      ERLOG = LOGF(ABSF(EREF))
      TTOL = 5E-8*ABSF(TMAX)+1E-8
      PUSHO = SIMP*FLOW*9.80665
      EXITA = AEXIT*100.
      MODOUT = MODOUT
      IF (DELT) 105,104,105
  104 DELT = TB(NSTAGE)/100.
      DELT1(NSTAGE) = DELT
  105 GO TO (109,106,106,109), MODOUT
  106 IF (DEL-DELMAX) 108,108,107
  107 DEL = MODF(DEL,DELMAX)
  108 IF (DEL) 114,109,114
  114 DELT = MIN1F(DELT,DEL)
  109 IF (XABSF(IMODE)-4) 1,110,1
  110 CALL TUDES
      IMODE = XSIGNF(2,IMODE)
    1 CALL NBODY
    2 CALL EXTRAS
C
C     PART 9. COMES HERE FOR END OF SUB TRAJECTORY.
      IF (DONE) 113,111,111
  111 DONE = 0.
      IF (NSTAGE-LSTAGE) 112,115,115
  112 NSTAGE = NSTAGE+1
      GO TO 100
  113 DONE = 0.
  115 CALL EXTRA
      IF (RETURN) 103,116,100
  116 RETURN
      END
                                                SUBROUTINE STEP
C   SUBROUTINE STEP TESTS FOR THE END OF THE PROBLEM, COMPUTES STEP SIZE, AND
C   CONTROLS QUANTITY OF OUTPUT DATA.  END OF PROBLEM OCCURS IF TIME = TMAX,
C     STEPGO+STEPNO = STEPMX, OR C(LOOKX) = XLOOK.  THE LAST OPTION ALLOWS STOP-
C     PING ON A DEPENDENT VARIABLE.  THE TEST FOR STOPPING AT XLOOK IS NOT MADE
C     UNTIL C(LOOKSW) IS GREATER THAN SWLOOK. CONTROL ON QUANTITY OF OUTPUT IS
C
C     MODOUT=1  OUTPUT EVERY NTH STEP(N=STEPS) UNTIL TIME = TMIN, THEN
C                   GO TO MODE 2 .
C             2  OUTPUT AT INTERVALS OF DELMAX UNTIL TIME = TMAX.
C             3  OUTPUT AT INTERVALS OF DELMAX UNTIL TIME = TMIN, THEN
C                   GO TO MODE 4 .
C             4  OUTPUT EVERY NTH STEP UNTIL TIME = TMAX.
C

      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   XPRIM(200),   DELT1 (10)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(A1    ,B      (  10)),(A2    ,B      (  11)),
     2(B     ,C      (1111)),(DELMAX,A      (  19)),(DEL   ,A      (  43)),
     3(DELT  ,B      (   1)),(DONE  ,B      (  39)),(E2    ,B      (  18)),
     4(END   ,A      (   5)),(ERLOG ,B      (  17)),(H2    ,B      (  15)),
     5(INLOOK,A      ( 599)),(LOOKSW,A      (   9)),(LOOKX ,A      (   8)),
     6(MODOUT,A      (  20)),(NSTAGE,A      (   3)),(DELT1 ,A      ( 133)),
     7(RATIO ,B      (  58)),(SIGNAL,B      (  31)),(SPACES,B      (  16)),
     8(STEPGO,A      (  41)),(STEPMX,A      (  16)),(STEPNO,A      (  42)),
     9(STEPS ,A      (  17)),(SWLOOK,A      (  10)),(TABLE ,C      (1911))
      EQUIVALENCE
     1(TMAX  ,B      (   4)),(TMIN  ,A      (  18)),(TTOL  ,A      (  45)),
     2(XLOOK ,A      (  12)),(XPRIM ,C      ( 711)),(XTOL  ,A      (  11)),
     3(NSTART,B      (  24)),(SWITCH,A      (601)),(OUTPOT,B      ( 399))
      CHECKF(A,B,C) = ABSF(A-B) - ABSF(A-C)
C
C     PART 1.  TEST FOR END OF THE PROBLEM (MAXIMUM PROBLEM TIME OR MAXIMUM
C              NUMBER OF STEPS).
      STEPGO = STEPGO + 1.
      OUT = OUTPOT
      IF (ABSF(TMAX-XPRIM(1))-TTOL) 1,1,3
    1 DONE = 1.0
  112 CALL OUTPUT
      IF (OUTPOT) 26,111,26
  111 WRITE OUTPUT TAPE 6,2,NSTAGE
    2 FORMAT(6HOSTAGEI2,11H COMPLETED.//)
      GO TO 26
    3 IF (STEPGO+STEPNO-STEPMX) 7,4,4
    4 CALL OUTPUT
      WRITE OUTPUT TAPE 6,5,STEPMX
    5 FORMAT (22HOSTEPGO+STEPNO=STEPMX=F6.)
      CALL EXIT
C
```

```
C     PART 2.  COMPUTE STEP SIZE (DELT) AND CONTROL OUTPUT.
    7 N=1
      A3 = (A2-A1)*RATIO+A2
      AA = (ERLOG-A3)/5.
      IF ((ABSF(AA)-88.028)*ABSF(SWITCH)) 8,8,60
    8 DELT = SIGNF(EXPF(AA),DELT)
      IF (DELT/H2-3.) 10,10,9
    9 DELT = 3.*H2
   10 MODOUT = MODOUT
      GO TO (11,15,13,21),MODOUT
   11 IF(DELT*(XPRIM(1) + 3.*DELT-TMIN)) 21,12,12
   12 MODOUT = 2
      DEL = TMIN - XPRIM(1)
      GO TO 16
   13 IF(DELT * (XPRIM(1) - TMIN)) 15,15,14
   14 MODOUT = 4
      GO TO 21
   15 DEL = DEL-H2
   16 SPACES = INTF((DEL/DELT)+SIGNF(.9,(DEL/DELT)))
   17 IF(SPACES) 20, 18,20
   18 CALL OUTPUT
      N=2
      DEL = DELMAX
      IF (ABSF(DEL) - ABSF(DELT)) 19,16,16
   19 DELT = SIGNF(DEL,DELT)
      GO TO 16
   20 DELT = DEL/SPACES
      GO TO 23
   21 IF (MODF(STEPGO,STEPS)) 23,22,23
   22 CALL OUTPUT
      N=2
C
C     PART. 3.  SEARCH FOR C(LOOKX) = XLOOK UNLESS LOOKX=0.
   23 IF(LOOK X) 27,42,27
   27 LOOK X = LOOK X
      LOOK SW = LOOK SW
      OUTPOT = 1.
      GO TO (44,45),N
   44 CALL OUTPUT
   45 IF(SWITCH) 32,28,33
   28 IF(SW LOOK - C(LOOK SW)) 29,29,42
   29 XTOL1 = XTOL*ABSF(XLOOK)
      IF (XTOL1) 31,30,31
   30 XTOL1 = XTOL
   31 SWITCH = -1.
      GO TO 41
   32 SWITCH = 1.
      ASSIGN 43 TO MODE
      OVER = 0.
      F = 0.
      T=0.
   33 SLOPE = (C(LOOKX)-OLDX)/H2
      GO TO MODE, (43,35)
   43 IF(SLOPE *(C(LOOK X) - X LOOK)) 350,41,41
  350 ASSIGN 35 TO MODE
   35 IF(ABSF(C(LOOK X)- X LOOK) - XTOL1) 36,36,37
   60 T=1.
   36 IF (OUT) 63,46,63
   46 OUTPOT = 0.
      CALL OUTPUT
   63 IF (T) 61,47,61
   61 IF (OUT) 62,51,62
   51 WRITE OUTPUT TAPE 6,64, LOOKX,C(LOOKX),H2,LOOKX,SLOPE
   64 FORMAT(3HOC(I4,4H) = 1PG15.8,31H   CONVERGENCE TROUBLE.   DELT=
     1G15.8,14H   SLOPE OF C(I4,13H) VS. TIME = G15.8//)
      GO TO 62
   47 IF (OUT) 62,50,62
   50 WRITE OUTPUT TAPE 6,48,LOOK X, C(LOOK X)
   48 FORMAT(3HOC(I4,2H)=1PG15.8//)
   62 LOOKX = 0
      XTOL1 = 0.
      SIGNAL = 1.
      SWITCH = 0.
      DONE = END
      NSTART = 0
      NSTAGE=NSTAGE
      DELT = DELT1(NSTAGE)
   49 CALL INPUT(INLOOK,C,TABLE)
      IF (DONE) 110,42,110
  110 IF (OUT) 26,111,26
   37 SIGN = CHECKF(OLDX,XLOOK,C(LOOK X))
      IF(SIGN) 40,40,38
   40 OVER = 1.
      GO TO 400
   38 IF (OVER) 400,401,400
  401 XGUESS = C(LOOKX)+SLOPE*DELT
      IF (CHECKF(C(LOOKX),XLOOK,XGUESS)) 402,41,41
  402 F = F+1.
      IF (F-7.) 400,400,403
  403 SLOPE = SLOPE/F
  400 IF (SLOPE) 404,60,404
  404 DELT = SIGNF(ABSF(XLOOK-C(LOOKX))/SLOPE,SIGN*H2)
   41 OLDX = C(LOOK X)
   42 IF (ABSF(TMAX-XPRIM(1))-ABSF(DELT)) 25,26,26
   25 DELT = TMAX-XPRIM(1)
      GO TO (26,24,24,26),MODOUT
   24 DEL = DEL-DELT
   26 OUTPOT = OUT
      RETURN
      END
```

```
                                                  SUBROUTINE STDATA
C     THIS ROUTINE CLEARS THE A, XPRIM, XPRIMB ARRARYS AND LOADS A SET OF
C     STANDARD DATA INTO THE MACHINE.  ANY VALUES SET HERE MAY BE OVERWRITTEN BY
C     INPUT 1 IN THE MAIN PROGRAM.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1    PNAME (12),         AMASS (30),         XPRIM (200),
     2                        COEFN (190),          ICC (4),
     3       AK (3),           XDOT (100),          IND (3),
     4     REFER (12),        RCRIT (30),            AW (4),
     5   RMASS1 (10)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(AK    ,A      (  51)),(AMASS ,A      ( 347)),
     2(AU    ,A      (  29)),(AW    ,A      (  55)),(B     ,C      (1111)),
     3(BODYCD,A      ( 143)),(COEFN ,A      ( 407)),(CONSTU,A      (  32)),
     4(CONSU ,A      (  31)),(DTOFFJ,A      (  23)),(EREF  ,A      (  13)),
     5(ERLIMT,A      (  14)),(ETOL  ,A      (  30)),(GASFAC,A      (  46)),
     6(ICC   ,A      ( 153)),(IMODE ,A      (   1)),(IND   ,A      (  60)),
     7(LOOKSW,A      (   9)),(MODOUT,A      (  20)),(NEQ   ,A      (   2)),
     8(NSTAGE,A      (   3)),(OBLATD,A      (  27)),(OBLATH,A      (  28)),
     9(OBLATJ,A      (  26)),(PNAME ,A      ( 287)),(RCRIT ,A      ( 377))
      EQUIVALENCE
     1(REFER ,A      ( 317)),(RE    ,A      (  25)),(SPD   ,A      (  44)),
     2(SQRDK1,A      (  47)),(STEPMX,A      (  16)),(STEPS ,A      (  17)),
     3(TFILE ,A      (   6)),(XDOT  ,B      ( 501)),(XPRIM ,C      ( 711)),
     4(XTOL  ,A      (  11)),(RMASS1,A      (  73))
C
C     CLEAR INITIAL CONDITIONS AND CONTROL PARAMETERS.
      DO 1 J=1,1100
    1 A(J) = 0.
C
C     THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
      PNAME(1) = 3HSUN
      PNAME(2) = 6HMERCUR
      PNAME(3) = 5HVENUS
      PNAME(4) = 5HEARTH
      PNAME(5) = 4HMARS
      PNAME(6) = 6HJUPITE
      PNAME(7) = 6HSATURN
      PNAME(8) = 6HURANUS
      PNAME(9) = 6HNEPTUN
      PNAME(10)= 5HPLUTO
      PNAME(11)= 4HMOON
      PNAME(12)= 6HEARTHM
C
C     FILL OUT SUN REFERENCE LIST.  INITIALIZE MASS ARRAY.
      DO 2 K=1,10
      RMASS1(K) = 1.
    2 REFER(K+1) = PNAME(1)
      REFER(12) = PNAME(1)
C
C     FILL OUT EARTH REFERENCE LIST.
      REFER(1) = PNAME(4)
      REFER(4) = 5HZERO+
      REFER(11) = PNAME(4)
C
```

```
C     LOAD THE REMAINING STANDARD DATA.
      AK(1) = 0.5
      AK(2) = 0.5
      AK(3) = 1.0
      AMASS(1) = 1.0
      AMASS(2) = 1.0/6120000.0
      AMASS(3) = 1.0/408645.0
      AMASS(4) = 1.0/332951.3
      AMASS(5) = 1.0/3088000.0
      AMASS(6) = 1.0/1047.39
      AMASS(7) = 1.0/3500.0
      AMASS(8) = 1.0/22869.0
      AMASS(9) = 1.0/18889.0
      AMASS(10) = 1.0/400000.0
      AMASS(11) =AMASS(4)/81.335
      AMASS(12) =AMASS(4)+ AMASS(11)
      AU = 1.49599 E11
      AW(1)=1./6.
      AW(2)=AW(1)+AW(1)
      AW(4)=AW(1)
      AW(3)=1.-(AW(2)+(AW(1)+AW(4)))
      BODYCD = PNAME(4)
      COEFN(1) = -1E20
      COEFN(189) = 1E20
      CONSTU = 1.0 E-6
      CONSU = 1E-6
      ETOL = 0.01
      DTOFFJ = 244.E4
      EREF = 1E-6
      ERLIMT = 3E-6
      GASFAC = 20.064881
      ICC(1) =185
      ICC(2) =185
      ICC(3) =185
      ICC(4) =185
      IMODE = 1
      IND(1)=2
      IND(2)=3
      IND(3)=1
      LOOKSW = 711
      MODOUT = 4
      NEQ=8
      NSTAGE = 1
      OBLATJ = 1.62345 E-3
      OBLATH = -5.75 E-6
      OBLATD = 7.875 E-6
      RCRIT(1) = 1.0 E+20
      RCRIT(2) = 1.0 E+8
      RCRIT(3) = 6.14 E+8
      RCRIT(4) = 9.25 E+8
      RCRIT(5) = 5.78 E+8
      RCRIT(6) = 4.81 E+10
      RCRIT(7) = 5.46 E+10
      RCRIT(8) = 5.17 E+10
      RCRIT(9) = 8.61 E+10
      RCRIT(10) =3.81 E+10
      RCRIT(11) =1.60 E+8
      RE = 6378165.
      SPD = 86400.0
      SQRDK1 = 2.959122083 E-4
      STEPMX= 100.0
      STEPS = 1.
      TFILE = 1.
      XDOT(1) = 1.0
      XPRIM(2) = RMASS1(1)
      XTOL = 5E-8
      WRITE OUTPUT TAPE 6,3
    3 FORMAT (15HOSTANDARD DATA.)
      RETURN
      END
```

```
                                                  SUBROUTINE TESTTR
C   SUBROUTINE TESTTR MAY BE CALLED FOR ONE OF TWO REASONS, (1) TO TEST FOR AND
C   POSSIBLY TRANSLATE THE ORIGIN (WHEN IMODE IS +) OR (2) TO CHANGE THE
C   VARIABLES OF INTEGRATION (WHEN IMODE IS -).  A TRANSLATION OF THE ORIGIN
C   OCCURS WHEN THE OBJECT MOVES INTO A SPHERE OF INFLUENCE WHICH IS SMALLER
C   THAN ANY OTHERS IT MAY ALSO BE IN.  WHEN THIS HAPPENS, THE NAME OF THE NEW
C   ORIGIN IS MOVED TO THE BEGINNING OF THE BNAME LIST AND ORDER IS
C   CALLED TO REORDER THE BNAME LIST.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1     XPRIM(100,2),XPRIMB(100,2),XWHOLE(6),VEFM(3,8),VX(3),
     2ORBELS(6),BMASS(8),BNAME(8),RB(3,8),RBCRIT(8),R(8)
C
      EQUIVALENCE
     1(A     ,C     (  11)),(AMC   ,B      (  87)),(ASYMPT,A     (   7)),
     2(B     ,C     (1111)),(BMASS ,B      ( 137)),(BNAME ,B     ( 122)),
     3(CHAMP ,B     (  25)),(DELT  ,B      (   1)),(GK2M  ,B     (  36)),
     4(IMODE ,A     (   1)),(NBODYS,B      (  41)),(ORBELS,B     ( 116)),
     5(RBCRIT,B     ( 145)),(RB    ,B      ( 193)),(REVS  ,A     (  48)),
     6(R     ,B     ( 102)),(SQRDK ,B      (  35)),(TABLE ,C     (1911)),
     7(TMAX  ,B     (   4)),(TRSFER,B      (   8)),(TRU   ,B     (  40)),
     8(TTEST ,A     (  54)),(VEFM  ,B      ( 241)),(VX    ,B     (  92)),
     9(XPRIM ,C     ( 711)),(XPRIMB,C      ( 911)),(XWHOLE,B     ( 110))
      EQUIVALENCE
     1(OUTPOT,B     ( 399))
C
      IMODE = IMODE
      IF (IMODE) 12,12,1
C
C     IF IMODE IS +, TEST FOR TRANSLATION OF THE ORIGIN.
    1 CHAMP= 1.E+30
      DO 4 JB=1,NBODYS
      IF (R(JB)-RBCRIT(JB)) 2,4,4
    2 IF (CHAMP-RBCRIT(JB)) 4,4,3
    3 CHAMP = RBCRIT(JB)
      NCHAMP = JB
    4 CONTINUE
      IF (NCHAMP-1) 26,26,5
    5 TRSFER = 1.0
    8 BTEMP = BNAME(1)
      BNAME(1) = BNAME(NCHAMP)
      BNAME(NCHAMP) = BTEMP
      TTEST = 0.
      REVS = 0.
      IF (OUTPOT) 6,9,6
    9 WRITE OUTPUT TAPE 6,10,BNAME(NCHAMP),BNAME(1)
   10 FORMAT (28HOORIGIN IS TRANSLATING FROM A6,4H TO A6)
    6 CALL EPHMRS
      DO 11 K=1,3
      VX(K) = VX(K)-VEFM(K,NCHAMP)
      RB(K) = RB(K,NCHAMP)
      XPRIM(K+2,1)=VX(K)
      XPRIM(K+5,1)=RB(K)
      XPRIMB(K+2,1) = 0.
      XPRIMB(K+5,1) = 0.
      XWHOLE(K)= VX(K)
   11 XWHOLE(K+3) = RB(K)
      GO TO 20
C
C     IF IMODE IS -, CHANGE THE VARIABLES OF INTEGRATION.
   12 DO 13 K=1,3
      XPRIM(K+2,1)=XWHOLE(K)
      XPRIM(K+5,1)=XWHOLE(K+3)
      XPRIMB(K+2,1) = 0.
      XPRIMB(K+5,1) = 0.
      VX(K) = XWHOLE(K)
   13 RB(K) = XWHOLE(K+3)
      GO TO (16,14,15),IMODE
   14 CODE = 5HORBIT
      IMODE = 1
      GO TO 18
   15 IMODE = 3
      GO TO 17
   16 IMODE = 2
   17 CODE = 6HRECTAN
   18 NCHAMP = 1
      IF (OUTPOT) 20,7,20
    7 WRITE OUTPUT TAPE 6,19,CODE
   19 FORMAT (33HOINTEGRATION MODE IS CHANGING TO A6)
   20 GO TO (21,26,26),IMODE
   21 CALL CONVT1(VX,AMC)
      GK2M= SQRDK*(BMASS(NCHAMP)+XPRIM(2,1)/1.9866 E+30)
   30 CALL CONVT2
C     IF ORIGIN TRANSLATION CAUSES PATH TO LIE NEAR AN ASYMPTOTE, CHANGE
C     INTEGRATION VARIABLES TO RECTANGULAR IF THEY ARE ORBIT ELEMENTS.
      IF (ORBELS(1)-1.) 24,24,22
   22 IF (ABSF(TRU)-2.3/SQRTF(ORBELS(1))) 24,24,23
   23 ASYMPT = 1.0
      GO TO 15
   24 DO 25 J=1,6
   25 XPRIM(J+2,1) = ORBELS(J)
   26 IF (TRSFER) 27,28,27
   27 CALL INPUT (101,C,TABLE)
   29 CALL ORDER
   28 RETURN
      END
```

```
                                                SUBROUTINE THRUST
C     THIS ROUTINE COMPUTES X,Y,Z THRUST ACCELERATIONS.  THE THRUST VECTOR IS
C     ASSUMED COINCIDENT WITH THE LONGITUNDINAL AXIS OF THE VEHICLE, WHICH IS
C     ORIENTED TO THE RELATIVE WIND VELOCITY BY THE ANGLE OF ATTACK (ALPHA) AND
C     THE ROLL ANGLE (BETA). ALPHA IS ASSUMED TO BE A QUADRATIC FUNCTION OF TIME
C     WHEREAS BETA IS ASSUMED TO BE CONSTANT.
C     REVOLV IS THE EARTHS ROTATION RATE IN RADIANS/SEC (7.29211585E-5) AND THE
C     FACTOR 8589934592.= 2**33 IS REMOVED TO PREVENT OVERFLOW.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1 FORCE(3), PAR(3), VATM(3), P(3), IND(3),RAMC(5),RB(3),X(100)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(AEXIT ,B      (   3)),(ALPHA ,A      (  49)),
     2(B     ,C      (1111)),(BETA  ,A      (  50)),(COSALF,B      (  48)),
     3(COSBET,B      (  49)),(EXITA ,B      ( 392)),(FLOW  ,B      (   5)),
     4(FORCE ,B      (  66)),(IND   ,A      (  60)),(PAR   ,B      (  60)),
     5(PMAGN ,B      (  50)),(PRESS ,B      (  33)),(P     ,B      (  84)),
     6(PUSHO ,B      ( 391)),(PUSH  ,A      ( 166)),(RAMC  ,B      ( 393)),
     7(RATMOS,B      (  23)),(RB    ,B      ( 193)),(REVOLV,B      (  21)),
     8(R     ,B      ( 102)),(RSQRD ,B      (  45)),(SIMP  ,B      (   2)),
     9(SINALF,B      (  46)),(SINBET,B      (  47)),(VATM  ,B      (  97))
      EQUIVALENCE
     1(VQ    ,B      ( 100)),(VQSQRD,B      ( 101)),(VX    ,B      (  92)),
     2(VY    ,B      (  93)),(VZ    ,B      (  94)),(X     ,B      ( 401))
C


      SINBET = SINF(BETA/57.2957795)
      COSBET = COSF(BETA/57.2957795)
      VATM(1)=VX+REVOLV*RB(2)
      VATM(2)=VY-REVOLV*RB(1)
      VATM(3)=VZ
    3 CALL CONVT1(VATM,RAMC)
    4 ALPHA = QUAD(X(1),1)/57.2957795
      SINALF=SINF(ALPHA)
      COSALF=COSF(ALPHA)
      DO 1 J1=1,3
      J2=IND(J1)
      J3=IND(J2)
    1 P(J1) = (VATM(J2)*RAMC(J3)-VATM(J3)*RAMC(J2))/8589934592.
      PMAGN= SQRTF(P(1)*P(1)+P(2)*P(2)+P(3)*P(3))
      PUSH = PUSHO-EXITA*PRESS
      TDPMAG = PUSH/PMAGN/X(2)
      R4    =  SINBET/VQ
      R5    =  COSALF/RAMC(4)
      DO 2 J1=1,3
      J2=IND(J1)
      J3=IND(J2)
      PAR(J1)=P(J2)*VATM(J3)-P(J3)*VATM(J2)
    2 FORCE(J1) = TDPMAG*(SINALF*(COSBET*P(J1)+R4*PAR(J1))-R5*(P(J2)*
     1              RAMC(J3)-P(J3)*RAMC(J2)))
      RETURN
      END


                                                SUBROUTINE TUDES
C     THIS ROUTINE COMPUTES THE RECTANGULAR POSITION AND VELOCITY COMPONENTS
C     WITH RESPECT TO THE EARTH MEAN EQUINOX AND EQUATOR OF 1950.0 FROM THE
C     LATITUDE, LONGITUDE, AZIMUTH, ELEVATION, ALTITUDE, TOTAL VELOCITY, AND
C     TIME.  ALSO, WHEN TKICK DOES NOT EQUAL ZERO, A NON-DRAG VERTICAL STEP OF
C     SIZE TKICK IS MADE IN CLOSED FORM (STATEMENTS 2 TO 4).  THE INTEGRATION
C     WILL THEN BEGIN AT TIME EQUAL TO TIME+TKICK WITH THE ORIENTATION SPECIFIED
C     BY THE ABOVE FOUR ANGLES AND THE COMPUTED VALUES OF ALTITUDE AND VELOCITY.
C     FOR THE CLOSED FORM APPROXIMATION, A CONSTANT FLOW RATE (FLOW), VACUUM
C     SPECIFIC IMPULSE (SIMP) AND ENGINE EXIT AREA (AEXIT) ARE ASSUMED KNOWN.
C     THE ATMOSPHERIC PRESSURE IS TAKEN TO BE THE SEA LEVEL VALUE.
C
      COMMON C
C
      DIMENSION A(600), B(700), C(4000),
     1   SINA(4), COSA(4), ANGLEB(4), XPRIM(200)
C
      EQUIVALENCE
     1(A     ,C      (  11)),(AEXIT ,B      (   3)),(ALT   ,A      (   4)),
     2(AZI   ,A      (  35)),(B     ,C      (1111)),(DTOFFJ,A      (  23)),
     3(ELEV  ,A      (  36)),(FLOW  ,B      (   5)),(GK2M  ,B      (  36)),
     4(LAT   ,A      (  33)),(LONG  ,A      (  34)),(OBLATJ,A      (  26)),
     5(OBLATN,A      (  40)),(RE    ,A      (  25)),(RESQRD,B      (   7)),
     6(ROTATE,A      (  39)),(SIMP  ,B      (   2)),(SPD   ,A      (  44)),
     7(STEPGO,A      (  41)),(STEPNO,A      (  42)),(TKICK ,A      (  15)),
     8(TOFFT ,A      (  24)),(VEL   ,A      (  37)),(XPRIM ,C      ( 711)),
     9(OUTPOT,B      ( 399))
      EQUIVALENCE (QLAT,LAT),( QLONG,LONG)
C
```

```
      ALT1 = 0.
      VEL1 = VEL
      DEL1 = 0.
      DEL = 0.
      ASSIGN 1 TO NGO
      DAYS = DTOFFJ - 2433282.5
      GREEN = MODF(100.0755426+.985647346DAYS+2.9015E-13DAYS**2
     1+7.29211585E-5*(TOFFT*SPD+XPRIM(1))*57.2957795,360.)
      SINA(1) = SINF(QLAT/57.2957795)
      IF (OBLATN) 102,101,102
  101 RADIUS = RE + ALT
      GO TO 8
  102 RADIUS=6356783.28/SQRTF(.9933065783+.006693421685*SINA(1)**2)+ALT
      GO TO 8
    1 XPRIM(6) = COSA(2)*COSA(1)*RADIUS
      XPRIM(7) = SINA(2)*COSA(1)*RADIUS
      XPRIM(8) = SINA(1)*RADIUS
      RMASSO = XPRIM(2)
      XPRIM(2) = XPRIM(2)-FLOW*TKICK
      IF (OUTPOT) 12,11,12
   11 WRITE OUTPUT TAPE 6,3,STEPGO,STEPNO,LAT,LONG,AZI,ELEV,ALT,XPRIM(
     11),VEL,RMASSO,(XPRIM(J),J=6,8)
    3 FORMAT(6HOSTEP=F5.,2H +F4.,4X,6H LAT.=1PG15.8,7H LONG.=G15.8,6H AZ
     1I.=G15.8,7H ELEV.=G15.8,6H ALT.=G15.8/6H TIME=G15.8,6H VEL.=G15.8,
     67H RMASS=G15.8,4X,2HX=G15.8,5X,2HY=G15.8,4X,2HZ=G15.8)
   12 IF (TKICK) 2,50,2
    2 XPRIM(1) = XPRIM(1)+TKICK
      B1 = LOGF(RMASSO/XPRIM(2))
      SIMPSL = SIMP-AEXIT/FLOW*10332.275
      VEL1 = VEL+SIMPSL*9.80665*B1-G*TKICK
      ALT1 = TKICK*(VEL-G*TKICK/2.+9.80665*SIMPSL*(1.-B1*XPRIM(2)/
     1     (RMASSO-XPRIM(2))))
    4 RADIUS = RADIUS + ALT1
      GREEN = GREEN + 7.29211585E-5*TKICK*57.2957795
      ASSIGN 5 TO NGO
      GO TO 8
    5 XPRIM(6) = COSA(2)*COSA(1)*RADIUS
      XPRIM(7) = SINA(2)*COSA(1)*RADIUS
      XPRIM(8) = SINA(1)*RADIUS
   50 IF (OBLATN) 6,7,6
    6 DEL1 = ATANF((C2-1.)/(C3-1.)*SINA(1)/COSA(1))*57.2957795-QLAT
    7 DEL2 = RADIUS/G*SINA(1)*COSA(1)*ROTATE*ROTATE*57.29577951
      DEL = DEL1 + DEL2
      ASSIGN 10 TO NGO
    8 ANGLEB(1) = QLAT + DEL
      ANGLEB(2) = QLONG + GREEN
      ANGLEB(3) = AZI
      ANGLEB(4) = ELEV
      DO 9 I=1,4
      SINA(I) = SINF(ANGLEB(I)/57.2957795)
    9 COSA(I) = COSF(ANGLEB(I)/57.2957795)
      C1 = 5.*RESQRD/RADIUS/RADIUS*OBLATJ
      C2 = C1*(SINA(1)*SINA(1)-.6)
      C3 = C1*(SINA(1)*SINA(1)-.2)
      G = GK2M/RADIUS/RADIUS
      GO TO NGO, (1,5,10)
   10 COS1 = COSA(1)*SINA(4)-COSA(4)*COSA(3)*SINA(1)
      COS2 = COSA(4)*SINA(3)
      XPRIM(3) = VEL1*(COS1*COSA(2)-COS2*SINA(2))-XPRIM(7)*ROTATE
      XPRIM(4) = VEL1*(COS1*SINA(2)+COS2*COSA(2))+XPRIM(6)*ROTATE
      XPRIM(5) = VEL1*(SINA(1)*SINA(4)+COSA(1)*COSA(3)*COSA(4))
      RETURN
      END

                                                SUBROUTINE TAPE
C     SUBROUTINE TAPE USES THE MASTER MERGED EPHEMERIDES TAPE (TAPE 9 AT LEWIS)
C     TO COMPILE A WORKING EPHEMERIS TAPE (TAPE 3 AT LEWIS) WHICH CONTAINS ONLY
C     THAT DATA NEEDED AT EXECUTION TIME.  THIS MINIMIZES TAPE HANDLING DURING
C     EXECUTION.  2 EPHEMERIS FILES ARE ON TAPE 9, FIRST FILE HAS DATA AND IS
C     IDENTIFIED BY THE SECOND WORD OF EACH 254 WORD RECORD (FIRST WORD IS THE
C     DUMMY FORTRAN COMPATIBLE WORD, SECOND WORD=2).  THE SECOND FILE IS ONLY 2
C     WORDS LONG, FIRST WORD IS FORTRAN COMPATIBLE, SECOND WORD=3).
C     MASTER FILE 1 -- PLANETS (EXCEPT MERCURY AND EARTH), SUN, MOON, AND
C                      EARTH-MOON BARYCENTER FROM SEPT.25, 1960 TO ABOUT 2000.
C     EACH EPHEMERIS COMPILED REQUIRES A SET OF INPUT 300 DATA.  THE FIRST PIECE
C     OF DATA WRITTEN ON A FILE IS THE FILE IDENTIFICATION NUMBER, FILE.  EACH
C     FILE IS NUMBERED CONSECUTIVELY STARTING WITH FILE=1. SINCE MOON DATA IS IN
C     TERMS OF EARTH RADII, THE CONVERSION OF MOON DATA TO A.U. IS MADE BEFORE
C     WRITING ON TAPE 3.  THE COMMON USED IN SUBROUTINE TAPE IS LOCAL AND ALL
C     BUT TAPE3 IS CLEARED BY A FINAL CLEARING LOOP.
C     FUNCTION COMPARF(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C     NORMAL INPUT - ELIST, TBEGIN, TEND, TAPE3
C
C     ELIST-  THE BCD LIST OF EPHEMERIS DATA NAMES TO BE PLACED ON
C             TAPE 3 .  THE NAMES ARE READ FROM CARDS,  AND IS USED TO
C             MAKE THE TMAKE LIST. ELIST IS NOT CHANGED IN STORAGE UNTIL
C             THE FINAL CLEAR FOR THIS SUBROUTINE.
C     TMAKE-  THE LIST OF EPHEMERIS NAMES WITH DUPLICATES DROPPED AND
C             ZERO SPACES CLOSED IN. AS THE EPHEMERIDES ARE FINISHED THE
C             NAMES ARE ERRASED FROM THIS LIST.
C     TMADE-  LIKE TMAKE BUT IS HELD FOR OUTPUT.
C     TBEGIN- THE BEGINNING DATE  EXPRESSED AS A JULIAN DAY.
C     TEND-   ENDING DATE EXPRESSED AS A JULIAN DAY.
C     INTVAL- THE APPROX. NUMBER OF DAYS COVERED BY ONE SET OF COEFF. IT
C             IS USED TO DECIDE WHICH DATA ARE TO BE ENTERED DOUBLE. THE
C             DOUBLE ENTRIES PERMIT FASTER OPERATION IF REVERSAL OF
C             INTEGRATION IS REQUIRED FOR ANY REASON.
C     EDATE-  JULIAN ENDING DATE FOR THE MASTER EPHEMERIS.
C     ERTOAU- EARTH RADII PER A.U.
C
      COMMON   C
C
```

```
      DIMENSION
     1        C (700),         TMAKE (12),          LIST (30),
     2    EDATE (12),         INTVAL (30),          KTAG (12),
     3    ELIST (11),          TMADE (12),         INTVA (2),
     4    PNAME (30),         TDATUM (252),       DATUMT (21,12)
C
      EQUIVALENCE
     1( TAPE3,C(  2)),(ERTOAU,C(  3)),(  KTAG,C(  4)),(  FILE,C( 16)),
     2( ELIST,C( 17)),(TBEGIN,C( 29)),(  TEND,C( 30)),( PNAME,C( 31)),
     3( KHAMP,C( 61)),( TMADE,C( 73)),( TMAKE,C( 85)),(TDATUM,C(441)),
     4( EDATE,C(127)),(INTVAL,C(157)),( INTVA,C(156)),(DATUMT,C(189))
C
B     COMPARF(A,B) = (A+B)*(-(A*B))
      REWIND 3
      DO 1  K=1,4000
    1 C(K) = 0.0
C
C     THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
C     NOTE.  THE EARTH IS NOT IN THIS LIST (NO EPHEMERIS FOR EARTH.)
      PNAME(1) = 3HSUN
      PNAME(2) = 6HMERCUR
      PNAME(3) = 5HVENUS
      PNAME(4) = 4HMARS
      PNAME(5) = 6HJUPITE
      PNAME(6) = 6HSATURN
      PNAME(7) = 6HURANUS
      PNAME(8) = 6HNEPTUN
      PNAME(9) = 5HPLUTO
      PNAME(10)= 4HMOON
      PNAME(11)= 6HEARTHM
C
C     PART 2.  SET UP JULIAN DATES ENDING EACH EPHEMERIS.
      EDATE(1) = 2451872.5                                             11/24/00
      EDATE(3) = 2451848.5                                             10/31/00
      EDATE(4) = 2451020.5                                              7/26/98
      EDATE(5) = 2473520.5                                                 2060
      EDATE(6) = 2473520.5                                                 2060
      EDATE(7) = 2473520.5                                                 2060
      EDATE(8) = 2473520.5                                                 2060
      EDATE(9) = 2473520.5                                                 2060
      EDATE(10)= 2440916.5                                             11/26/70
      EDATE(11)= 2451848.5                                             10/31/00
      INTVA  = 30000
      INTVAL(1) = 8
      INTVAL(2) = 5
      INTVAL(3) = 15
      INTVAL(4) = 44
      INTVAL(5) = 330
      INTVAL(6) = 825
      INTVAL(7) = 1211
      INTVAL(8) = 1172
      INTVAL(9) = 1101
      INTVAL(10) = 2
      INTVAL(11) = 15
      FILE = 1.
      ERTOAU = 4.26546512 E-5
    2 MOON = 0
      LI = 1
C
C     PART 2B. CALL INPUT AND SEE IF TAPE IS TO BE MADE.  INPUT  MUST ALWAYS
C                 MAKE TAPE3=0.0 IF TAPE IS TO BE MADE.
      TAPE3 = 3.
    8 CALL INPUT(300,C,LIST)
      IF (TAPE3) 63,3,63
    3 IF (FILE-1.) 20,10,20
   10 CALL SKFILE(9,2)
C
C     PART 3. TAPE IS TO BE MADE SO MOVE EPHEMERIS LIST TO TMAKE AND
C          TO TMADE (FOR OUTPUT), CANCEL ANY ZERO OR DUPLICATE NAMES.
   20 KOUNT = 1
      DO 6 K=1,11
      TMAKE(K) = 0.
      TMADE(K) = 0.
    4 DO 5 J=1,KOUNT
      IF (COMPARF(ELIST(K),TMAKE(J-1))) 5,6,5
    5 CONTINUE
      TMAKE(KOUNT) = ELIST(K)
      TMADE(KOUNT) = ELIST(K)
      KOUNT = KOUNT+1
    6 CONTINUE
      KOUNT = KOUNT - 1
C
C     PART 4. FIND INPUT ERRORS.
    7 IF(TBEGIN-2437202.5) 66,9,9
    9 KM = 2
   11 ERROR = 0.
      WRITE TAPE 3,FILE
      DO 21 J=1,KOUNT
      KTAG(J) = 0
   12 DO 13 K=1,20
      IF (COMPARF(PNAME(K),TMAKE(J))) 13,16,13
   13 CONTINUE
C

C     PART 5. PRINTS OUT THE  MISSPELLED NAMES AND OTHER ERRORS.
   14 PRINT   15,  TMAKE(J), TBEGIN, TEND
      WRITE OUTPUT TAPE 6 , 15,  TMAKE(J), TBEGIN, TEND,(PNAME(K),
     1EDATE(K),K=1,20)
   15 FORMAT( 23H TROUBLE ON TAPE 3 MAKE  / 2X,A6,10H  T BEGIN= F10.1,8H
     1  T END= F10.1//2(2X,A6,F20.1))
      ERROR = 1.
      GO TO 21
C
```

```
C     PART 4B. CHECKS DATES AND STORES INDEX FOR MOON SO THAT EARTH
C     RADII CAN BE CONVERTED TO A.U.
   16 IF (10-K) 18,17,18
   17 MOON = J
   18 KTAG(J) = K
   19 IF (EDATE(K)- TEND)  14,21,21
   21 CONTINUE
      ASSIGN 36  TO NS1
      IF (ERROR) 22,22,68
C
C     PART 6. FIX UP A TAG (KTAG) TO INDICATE WHETHER TO ENTER DATA DOUBLE OR
C             NOT.  KHAMP WILL BE SHORTEST INTERVAL.  KTAG WILL BE NON-ZERO IF
C             ANY DATA ENTERS MORE THAN ONCE FOR 10 ENTRIES OF THE MOST
C             FREQUENT DATA.
   22 KHAMP = INTVAL(0)
      DO 23  J=1,KOUNT
      K = KTAG(J)
      KHAMP = XMINOF(KHAMP,INTVAL(K))
   23 CONTINUE
      KHAMP = KHAMP *10
      DO 24  J=1,KOUNT
      K = KTAG(J)
   24 KTAG(J) = INTVAL(K) / KHAMP
C
C     PART 7.  LOCATE FILE 2 ON TAPE 9.
   25 READ TAPE 9, KFILE
   26 IF (KM-KFILE) 27,31,29
   27 IF (KFILE - 3)  28,28,29
   28 BACKSPACE 9
      BACKSPACE 9
      CALL BSFILE(9)
      GO TO 25
C     BY PASS A FILE.
   29 CALL SKFILE(9)
      GO TO 25
C
C     PART 8. THIS IS CORRECT FILE ON TAPE 9, READ DATA. THERE CAN BE UP
C        TO 12 SETS OF DATA PER RECORD. A SET OF DATA IS 21 WORDS.
   31 BACKSPACE 9
   32 READ TAPE 9, KTAPE,(TDATUM(I), I=1,252)
      GO TO NS1, (36,46)
C
C     PART 9. IS THIS A SATISFACTORY STARTING POINT, QUESTION MARK.
C     THE 1ST SET OF DATA FOR EACH PLANET MUST PRE DATE TBEGIN.
C     PART 9 IS EXECUTED ONLY ONCE.
   36 DO 42 J=LI,KOUNT
      DO  37 K=1,232,21
      IF (COMPARF(TDATUM(K),TMAKE(J))) 37,39,37
   37 CONTINUE
   38 LI = J
      BACKSPACE 9
      BACKSPACE 9
      GO TO 32
   39 IF (TDATUM(K+1)-TDATUM(K+2)-TBEGIN) 40,40,38
   40 DO 41 KJ=1,21
      K1 = K + KJ - 1
   41 DATUMT(KJ,J) = TDATUM(K1)
   42 CONTINUE
      IF (MOON) 43,45,43
   43 DO 44 KJ=4,21
   44 DATUMT(KJ,MOON) = DATUMT(KJ,MOON)*ERTOAU
   45 ASSIGN 46 TO NS1
C
C     PART 10. PUT AWAY NEEDED DATA.  TEST NAME, TIME OF BEGIN AND END.  DO NOT
C              WRITE TAPE 3 UNTIL TBEGIN PREDATES THE END OF THE FITTED
C              INTERVAL.  50 REPEATS OLD DATA, 57 WRITES NEW DATA.  THE NAMES
C              ARE ERASED FROM TMAKE AS SOON AS THE DATA POST DATES TEND.  WHEN
C              ALL NAMES ARE GONE, RETURN TO INPUT 300 TO SEE IF ANOTHER
C              EPHEMERIS IS TO BE CONSTRUCTED.
   46 DO 65 K=1,232,21
      DO 47 J=1,KOUNT
      IF (COMPARF(TDATUM(K),TMAKE(J))) 47,48;47
   47 CONTINUE
      GO TO 65
   48 SWT = TBEGIN-TDATUM(K+1)-TDATUM(K+2)
      IF (SWT) 49,49,52
   49 IF(KTAG(J)) 50,52,50
   50 WRITE TAPE 3,(DATUMT(KJ,J) , KJ=1,21)
   52 DO  53 KJ=1,21
      K1 = K + KJ
   53 DATUMT(KJ,J) = TDATUM(K1-1)
      IF (J-MOON) 56,54,56
   54 DO  55 KJ = 4,21
   55 DATUMT(KJ,J) = DATUMT(KJ,J)*ERTOAU
   56 IF (SWT) 57,57,58
   57 WRITE TAPE 3,(DATUMT(KJ,J),KJ=1,21)
   58 IF(TEND-DATUMT(2,J)-DATUMT(3,J)) 59,59,65
   59 TMAKE(J) = 0.
      DO 60 KK=1,KOUNT
      IF (TMAKE(KK)) 65,60,65
   60 CONTINUE
      WRITE OUTPUT TAPE 6, 61, FILE,TBEGIN,TEND, KOUNT,(TMADE(KK),
     1KK=1,KOUNT)
   61 FORMAT(28HOEPHEMERIS COMPLETED,  FILE=F3.,6H, FROM F10.1,3H TO
     1 F10.1, 4H FOR I2, 18H BODIES AS FOLLOWS / 12(2X,A6))
      FILE = FILE + 1.
      END FILE 3
      GO TO 2
   63 WRITE TAPE 3, FILE
      REWIND 3
      REWIND 9
      TAPE3 = 3.
      DO 64 J=3,4000
   64 C(J) = 0.
      RETURN
C
```

```
   65 CONTINUE
      GO TO 32
   66 PRINT 67, TBEGIN
      WRITE OUTPUT TAPE 6,67,TBEGIN
   67 FORMAT(33H TBEGIN PREDATES 2437202.5,IT IS F10.1)
   68 CONTINUE
      REWIND 9
      END

       REM BSFILE(I,J) BACKSPACES TAPE I UNTIL IT IS POSITIONED JUST
       REM BEHIND THE J TH EOF MARK.
       REM
       ENTRY   BSFILE
       PZE
       PZE
       PZE
       BCD 1BSFILE
BSFILE SXD     *-4,1
       SXD     *-4,2
       SXD     *-4,4
       XEC*    $(TES)
       TSX     $(RER),4
       LXD     BSFILE-2,4
       CLA*    1,4
       TSX     $(IOS),4
       CLA*    $(RDS)
       STA     BSF
       ANA     A07000
       STA     BTT1
       STA     BTT2
       LXD     BSFILE-2,4
       CAL     2,4
       ANA =0777777700000
       ERA =0007400000000
       TNZ ONEARG
       CLA* 2,4
       TZE BACK
       PDX ,1
       AXC     *+1,4
       XEC*    $(TCO)
BTT1   BTTA    **
       TRA     *+1
BSF    BSFA    **
       XEC*    $(RDS)
       XEC*    $(BSR)
       AXC     *+1,4
       XEC*    $(TCO)
BTT2   BTTA    **
       TRA CHECK
       TIX BSF,1,1
       XEC* $(RDS)
BACK   AXC     *+1,4
       XEC*    $(TCO)
       AXC     *+1,4
       XEC*    $(TRC)
       NOP
       AXC     *+1,4
       XEC*    $(TEF)
       NOP
       LXD     BSFILE-4,1
       LXD     BSFILE-3,2
       LXD     BSFILE-2,4
       TRA 3,4
CHECK  TXL BACK,1,1
       LXD BSFILE-2,4
       CLA ERR+1
       STO 0
       CLA* 1,4
       LDQ* 2,4
ERR    TSX 8,4
       TXI BACK,0,14
       PZE BSFILE-2,0,ERR
ONEARG CLA BSFILE-2
       ADD =01000000
       STO BSFILE-2
       LXD CHECK,1
       TRA BTT1-2
A07000 OCT 7000
       END
```

```
       REM SKFILE(I,J) SKIPS TAPE I OVER J EOF MARKS.
       REM
       ENTRY   SKFILE
       PZE
       PZE
       PZE
SKFILE SXD     *-3,1
       SXD     *-3,2
       SXD     *-3,4
       TSX     $(RER),4       CHECK LAST READ
       TEFA    *+1
       TEFB    *+1
       LXD     SKFILE-1,4
       CLA*    1,4            PICK UP THE TPE NUMBER
       TSX     $(IOS),4       SET UP THE TAPE ADDRESSES
       LXD     SKFILE-1,4     LOAD IT AGAIN--MAN
       CAL     2,4            IS THERE A SECOND ARGUMENT
       ANA     =0777777700000
       ERA     =0007400000000
       TNZ     ONEARG         NO SECOND ARGUMENT
GOGO   CLA*    2,4            PICK UP THE SECOND ARGUMENT
       TZE     BUMP+1         DID SOME DUMMY WANT   NO FILES
LOOP   SUB     =01000000
RDS    XEC*    $(RDS)         READ THE TAPE
       TCOA    *
       TCOB    *
       TEFA    BUMP           DID WE HIT
       TEFB    BUMP           AN END OF FILE
       TRA     RDS            GO READ SOME MORE
BUMP   TNZ     LOOP
       LXD     SKFILE-3,1
       LXD     SKFILE-2,2
       LXD     SKFILE-1,4
       NOP
       TRCA    *+1            TURN OFF TAPE CHECK
       TRCB    *+1
       TRA     3,4
ONEARG CLA     SKFILE-1
       ADD     =01000000      SET UP XR4 FOR PROPER RETURN
       STO     SKFILE-1
       PXD     0,,0           SET UP FOR ONE FILE
       TRA     RDS
       END
       COUNT   1200                                                       00020
       REM INPUT ROUTINE USING ARITHMETIC STATEMENTS. CF NASA TN D-1092   00030
       LBL     INPUT,6                                                    00040
       ENTRY   INPUT                                                      00050
       REM THIS IS SUBROUTINE INPUT.  ITS CALLING SEQUENCE                00060
       REM CONTAINS THREE ARGUMENTS---AN IDENTIFICATION                   00070
       REM CODE NUMBER, THE FIRST LOCATION RELATIVE TO WHICH              00080
       REM ALL DATA IS TO BE LOADED, AND THE FIRST LOCATION               00090
       REM OF A TABLE TO BE USED BY THE ROUTINE.                          00100
       REM                                                                00110
       REM                                                                00120
       REM INCLUDED IN THIS ASSEMBLY ARE SUBROUTINES                      00130
       REM    1  INPUT                                                    00140
       REM    2  CHRCTR                                                   00150
       REM    3  CLEAR                                                    00160
       REM    4  COMPAR                                                   00170
       REM    5  ERROR                                                    00180
       REM    6  LOOK                                                     00190
       REM    7  NAME                                                     00200
       REM    8  NUMBR                                                    00210
       REM    9  STORE                                                    00220
       REM   10  TABLE                                                    00230
       REM   11  TEST                                                     00240
       REM   12  ACCUM, FIX, FLT, BINARY                                  00250
       REM   13  PRINX                                                    00260
       REM   14  READ.                                                    00270
       REM                                                                00280
INTAPE PZE     0,,7          LEWIS INPUT TAPE  NOT STD.
OUTAPE PZE     0,,6          FORTRAN STANDARD OUTPUT TAPE
 INDX  PZE                         STORAGE FOR IRA                        00290
       PZE                         IRB                                    00300
       PZE                         IRC                                    00310
       BCI     1,INPUT                                                    00320
INPUT  SXD     INDX,1        SAVE INDEX REGISTER A.                       00330
       SXD     INDX+1,2      SAVE INDEX REGISTER B.                       00340
       SXD     INDX+2,4      SAVE INDEX REGISTER C.                       00350
       NZT*    1,4           IF THE IDENTIFICATION  NUMBER IS Z           00360
       TRA     4,4           RETURN TO THE CALLING PROGRAM.               00370
       CLA     =1B35                                                      00380
       ADD     2,4           2,4 IS THE BASE LOCATION.                    00390
       STA     SET                                                        00400
       STA     LOC1                                                       00410
       STA     LOC4                                                       00420
       CLA     TSXBS         OPEN BACKSPACE GATE                          00430
       STO*    $(LINK)       CALL CHAIN WILL BACKSPACE                    00440
LOCA   CLA     1,4           1,4 IS THE IDENTIFICATION NUMBER.            00450
       STA     NREG1                                                      00460
       AXT     36,1          INITIALIZE 36                                00470
       STZ     I+1,1         LOCATIONS                                    00480
       TIX     *-1,1,1         TO ZERO.                                   00490
       STO     ILOC1         MAKE NON'ZERO.                               00500
       CLA     3,4           3,4 IS THE LOCATION OF THE TABLE.            00510
       STA     LOCFC         PREPARE                                      00520
       STA     NREG1-1                                                    00530
       ADD     =1B35               THE                                    00540
       STA     LOCFA           ARGUMENT STORAGES                          00550
       STA     LOCKL                                                      00560
       TSX     CLEAR,4       CLEAR THE VAR REGION.                        00570
```

```
LOCA1  CLA     =0076100000000     INHIBIT READING UNTIL
       STO     READ.               ARRAY RECORD REFRESHED
       AXT     43,2          43 FORCES RECORD TO BE FILLED
       SXD     I,2           IN CHRCTR
       REM LOOK AT  THE FIRST CHARACTER ON THE FIRST CARD
       REM IN SEARCH OF A $ SIGN.
LOCAA  TSX     CHRCTR,4
       SUB     =H0000$0            CHECK FOR A $ SIGN
 LOCA. STO     WORD
       TSX     COMPAR,4
       OCT     242517630000        D, E, FILE FLAG, T
       LXA     NREG1,4       ZERO IF $D HAS BEEN READ.
       TXL     *+2,4,0
       TXI     *+1,2,4       BEFORE $D ADD 4 TO INDEX 2.
       TXH     ERRU,2,7      JUNK
       TXH     SGNOUT,2,6    $17 BEFORE $D. FILE FLAG. OFF
       TXL     *+3,2,5       $E BEFORE $D
       TSX     READ.+1,4     CRASH READ GATE
       TRA     LOCA1         SHOULD NOW HAVE $D CARD
       TXH     LOCAD,2,4     FIRST $D.
       TXH     LOCAJ,2,3     $T AFTER $D.
       TXH     LOCCK,2,2     $17 AFTER $D. FILE FLAG
       TXH     LOCBG,2,1     $E AFTER $D.
       REM
LOCAC  LXA     READ.,4       $D AFTER $D. TEST IF BUFFER
       TXL     ERRU,4,0            OVERWRITTEN
       REM THIS IS THE PROGRAM RETURN.
RTN    LXD     INDX,1        RESET INDEX A.
       LXD     INDX+1,2      RESET INDEX B.
       LXD     INDX+2,4      RESET INDEX C.
       TRA     4,4           RETURN TO CALLING PROGRAM.
       REM HUNT FOR THE = SIGN OF THE $ DATA CARD.
 LOCAD CLA     =0076100000000      INHIBIT READING UNTIL
       STO     READ.               $DATA FIELD SCANNED
       TSX     CHRCTR,4
       TSX     COMPAR,4
       BCI     1,=00000
       TRA     *+5,2,2
       TRA     ERRD          JUNK
       TRA     LOCAD         ALPHABETIC
       TRA     ERRD          NUMERIC
       SXD     ALF,4         = SIGN
       REM USE ALF MODE TO TEST ALL CHARACTERS.
       REM
       REM COMES HERE WHEN = SIGN HAS BEEN FOUND.  GET THE
       REM IDENTIFICATION NUMBER FROM THE CARD.
 LOCAF LXD     I,4
       TXH     *+2,4,43
       TXH     LOCAG,4,42    CARD SCANNED OUT.
       TSX     CHRCTR,4
       TSX     COMPAR,4
       BCI     1,,$+- 0
       TRA     *+9,2,2
       TRA     ERRM          JUNK
       TRA     ERRM          ALPHABETIC
 LOCAE TSX     BINARY,4      FORM BIN WD IN VAR
       TRA     LOCAF         BLANK
       SXD     ERSW,2        MINUS  SET TO BY PASS.
       TRA     LOCAF         PLUS   NO EFFECT.
       STO     SIGN          DOLLARS
       REM COMES HERE TO CHECK THE REGION CODE AND THE
       REM VALUE APPEARING ON THE $DATA CARD.
LOCAG  CLA     VAR           COMMA
       TZE     ERRU          DATA SET NO. MISSING
       ALS     18
       STD     **            SAVE IDENT AT TABLE(1),
NREG1  SUB     **            PLACE FIRST ARG IN THIS ADDRESS.
       TNZ     RTN                 0 IF CALL CODE = $DATA CODE
       STZ     ALF           ALF = 0 MEANS NO ALF INFO.
       SXA     NREG1,0             INDICATE $DATA IS READ.
       REM INST. BELOW ALSO EXECUTED AT READ., PLACED THERE BY CHRCTR
 TSXRD TSX     READ.+1,4     HERE SNEAK PAST READ. GATE
       SXD     TESTJK,0
       TRA     LOCAN
       REM
       REM COMES HERE IF IT WAS A $ TABLE CARD.
LOCAJ  TSX     TABLE,4
       TRA     LOCAN3
       REM
       REM COMES HERE IF AN ALPHABETIC CHARACTER WAS FOUND.
LOCAK  TSX     NAME,4
       TNZ     SET-1         ZERO MEANS ON LEFT OF = SIGN.
       LXD     JK1,1         IF JK1 DIDNOT INCREASE THEN
TESTJK TXL     ERRL,1,**     AN = SIGN WAS NOT USED.
       SXD     TESTJK,2      SAVE JK1 FOR NEXT TEST.
       CLA     ILOC          SAVE SIGN OF TABLE ENTRY.
       STO     ILOC1
       TRA     LOCAN2
       REM
       LXD     JK,2          PREPARE TO ACCUMULATE THE NUMBERS
SET    CLA     **,2          IN THE PSEUDO ACCUMULATOR.
       STO     TEMP
       CLA     ILOC
       TPL     LOCAM         MINUS MEANS FLOAT THE NUMBER.
       TSX     FLT,4
       TRA     LOCAM
       REM
       REM COMES HERE IF NUMERIC FIELD.
LOCAL  TSX     NUMBER,4
       STO     TEMP
```

```
LOCAM   TSX     ACCUM,4         ACCUMULATE RESULTS IN ACC.
        TSX     CLEAR,4
        LXA     WORD,4
        PXA     0,4                   + WORD IN ACC FOR LOCAR
        TXL     LOCAR,4,58      NOT COMMA
        TXH     LOCAR,4,59      NOT COMMA
        LXD     JK1,2           COMMA
        CLA     ACC
        STZ     ACC             INITIALIZE
        LDQ     ILOC1           IS THIS VARIABLE FIXED POINT.
        TQP     LOC1            NEGATIVE IS FIXED POINT.
        TSX     FIX,4
LOC1    STO     **,2            STORE THE NUMBER RELATIVE TO BASE.
LOCAN   LXD     JK1,2
LOCAN1  TXI     *+1,2,1         RAISE STORING INDEX BY ONE.
LOCAN2  SXD     JK1,2           SAVE IT.
LOCAN3  LXD     OPER,1          ANY OPERATORS LEFT OVER.
        TXL     *+3,1,0
ERRL    TSX     ERROR,4
        BCI     1,0(L)
        CLA     ACC             ANY DATA LEFT OVER.
        TNZ     ERRL
        REM
        REM CALL THIS THE SWITCH HOUSE.
LOCAO   TSX     CLEAR,4
LOCAP   TSX     CHRCTR,4
LOCAQ   TSX     COMPAR,4
        BCI     1,.(0000
        TRA     *+6,2,2
        TRA     LOCAR           $D, $T, OR OPERATORS.
        TRA     LOCAK           ALPHABETIC
        TRA     LOCAL           NUMERIC
        TRA     LOCAT           ( SIGN
        TRA     LOCAL           DECIMAL
LOCAR   LXD     OPER,1          ANY OPERATORS LEFT OVER.
        TXH     ERRL,1,0        HIGH MEANS ALREADY HAS OPERATOR.
        SUB     =H0000$0        SPLIT OFF $ FROM OTHERS
        TPL     LOCA.           IF + PROCESS $ TYPE CH
        REM WHAT KIND OF OPERATOR IS THIS.
        TSX     COMPAR,4
        BCI     1,+-/*,0
        TXH     ERRL,2,5        REMOVE THE JUNK.
        TXH     LOCAN,2,4       COMMA
        SXD     OPER,2          SAVE REST, WILL BRANCH IN SUB ACCU
        TRA     LOCAP           AFTER BOTH OPERANDS HAVE BEEN FOUN
        REM
        REM COMES HERE IF THE OCT OR ALF MODE.
LOCAT   TSX     CHRCTR,4
        TSX     COMPAR,4
        BCI     1,)0A000
        TRA     *+5,2
        TRA     ERRL            JUNK
        TRA     LOCAZ           A CHARACTER
        TRA     LOCAU           O CHARACTER
        REM
        REM COMES HERE IF EMPTY PARENTHESIS WERE FOUND.
        TSX     CHRCTR,4        )SIGN,  GET NEXT CHARACTER.
        TQP     *+2             MINUS FOR NEW CARD
        TSX     TEST,4          INSERT COMMA IF NEEDED.
        CLA     ILOC1
        STO     ILOC            PREPARE TO GET VALUE OF
        LXD     JK1,2           CURRENT LEFT SIDE.
        TRA     SET
        REM COMES HERE IF OCTAL MODE.
LOCAU   TSX     CHRCTR,4
        SUB     =H00000)
        TNZ     LOCAU
        TRA     LOCAW           ) SIGN
 LOCAV  LDQ     VAR
        RQL     3               REPLACE TOP 3 BITS
        LGR     3               BY NEXT OCTAL CHARACTER
        RQL     3               PUT IN BOTTOM OF MQ
        STQ     VAR
        REM COMES HERE WHEN ) IS FOUND.
LOCAW   TSX     CHRCTR,4
        TQP     *+2             MINUS FOR NEW CARD
        TSX     TEST,4
        LXA     WORD,4          CHARACTER TO IRC
        TXL     LOCAV,4,7       OCTAL DIGITS
        TXL     ERRJ,4,58       ALPHABETIC, JUNK, 8, 9.
        TXH     ERRJ,4,59       SPLITS (
LOCAX   LXD     JK1,2           COMMA
        CLA     VAR
        TRA     LOC1
        REM                CONVERT THE NUMBER TO BINARY.
LOCAY   TSX     BINARY,4
        REM
        REM COMES HERE IF ALF MODE.
LOCAZ   TSX     CHRCTR,4
        TSX     COMPAR,4
        BCI     1,)00000
        TRA     *+5,2,2
        TRA     ERRK            JUNK
        TRA     LOCAZ           ALPHABETIC
        TRA     LOCAY           NUMERIC
        REM COMES HERE WHEN ) IS FOUND
```

```
LOCBA  LXA     VAR,1          ) SIGN                                    02540
       TNX     ERRK,1,0       ALF COUNT WAS ZERO.                       02550
       SXD     ALF,1                                                    02560
       TSX     CLEAR,4                                                  02570

       TSX     CHRCTR,4       PULL THROUGH CHARACTERS AND STORE         02580
       SUB     =017           FILE FLAG, NEVER NEG.                     02590
       TZE     ERRB                 COUNT WENT PAST E O JOB.            02600
       TSX     STORE,4        THEM ONE AT A TIME.                       02610
       TIX     *-4,1,1        GO BACK TILL NCHAR = 1                    02620
       LXD     J,1                                                      02630
       LXD     MSHIFT,4                                                 02640
       CAL     BLANK                                                    02650
       LGR     42,4                                                     02660
       ORS     VAR+1,1        FILL IN PARTIAL WORD WITH BLANKS.         02670
 LOCBB AXT     1,4            IRC TO 1                                  02680
       LXD     JK1,2                                                    02690
       CLA     J              PREPARE TO STORE ALF WDS                  02700
       STD     LOCBC1                                                   02710
 LOCBC SXD     JK1,2                                                    02720
       CLA     VAR+1,4                                                  02730
LOC4   STO     **,2                                                     02740
       TXI     *+1,4,1        J = J + 1                                 02750
LOCBC1 TXH     LOCBD,4,**                                               02760
       TXI     LOCBC,2,1      JK=JK+1                                   02770
LOCBD  STZ     ALF                                                      02780
       TSX     CLEAR,4                                                  02790
       TSX     CHRCTR,4       LOOK AT NEXT CHARACTER.                   02800
       TQP     *+2            MINUS FOR NEW CARD                        02810
       TSX     TEST,4         PUT IN COMMA IF NEEDED.                   02820
       SUB     =H00000,                                                 02830
       TZE     LOCAN          GO RAISE AND STORE JK1.                   02840
       REM                                                              02850
       REM THESE ARE ERROR CALLS                                        02860
ERRB   TSX     ERROR,4                                                  02870
       BCI     1,0(B)                                                   02880
ERRD   TSX     ERROR,4                                                  02890
       BCI     1,0(D)                                                   02900
ERRJ   TSX     ERROR,4                                                  02910
       BCI     1,0(J)                                                   02920
ERRK   TSX     ERROR,4                                                  02930
       BCI     1,0(K)                                                   02940
ERRM   TSX     ERROR,4                                                  02950
       BCI     1,0(M)                                                   02960
ERRU   TSX     ERROR,4                                                  02970
       BCI     1,0(U)                                                   02980
       REM                                                              02990
       REM        $E COMES HERE AFTER $D                                03000
LOCBG  CLA     =0076100000000    NOP                                    03010
       STO*    $(LINK)        CLOSE BACKSPACE GATE                      03020
       TRA     LOCAC          RETURN                                    03030
       REM PURPOSE OF $END CARD IS TO PROTECT FORIEGN DATA FROM         03040
       REM BACKSPACE WHEN CHAIN IS CALLED.                              03050
       REM                                                              03060
       REM     END OF THE MAIN SEGMENT                                  03070
       REM THIS A ROUTINE TO BACKSPACE THE INPUT TAPE WHEN A            03080
       REM CALL CHAIN IS GOING TO SPILL THE BUFFER.                     03090
       REM THIS ROUTINE IS EXECUTED FROM CHAIN VIA THE ONE              03100
       REM WORD SUBROUTINE (LINK) WHICH CONTAINS EITHER TSX OR NOP      03110
TSXBS  TSX     LOCBS,4        TO BE STORED AT (LINK                     03120
LOCBS  SXA     *+4,4          SAVE INDEX 4                              03130
       CLA     INTAPE         INPUT TAPE NUMBER
       CALL    $(IOS)         SELECT INPUT TAPE                         03150
       XEC*    $(BSR)         BACKSPACE IT                              03160
       AXT     **,4           RESTORE INDEX                             03170
       TRA     1,4            RETURN TO THE CHAIN ROUTINE               03180
       EJECT                                                            03190
       REM THIS IS SUBROUTINE CHRCTR.  IT STORES SUCCESSIVE             03200
       REM CHARACTERS FROM THE CARD AT LOCATION WORD, READS             03210
       REM SUCCESSIVE CARDS INTO THE ARRAY RECORD, AND PRINTS           03220
       REM $$ TYPE CARDS. THE FIRST CHARACTER FROM A NEW CARD           03230
       REM IS STORED IN WORD WITH A MINUS SIGN.                         03240
       REM                                                              03250
       REM                                                              03260
CHRCTR SXD     TEMP-10,2                                                03270
       SXD     TEMP-17,4                                                03280
       LXD     I,2            CARD COL COUNT, SAW COUNT                 03290
       TXH     *+2,2,83       TOO EARLY TO READ.                        03300
       XEC     READ.          GATE MAY BE CLOSED                        03310
       LDQ     Q              HAS UNUSED CHARACTERS FROM BEFORE         03320
       CLA     SIGN           ZERO OR $ GOES TO TAG                     03330
LOCCA  ALS     6              SHIFT LEFT 1 CHARACTER                    03340
       SLW     TAG            CLEARS OR PRELOADS TAG                    03350
LOCCB  LXD     ALF,4          NONZERO MEANS ALF MODE.                   03360
LOCCC  TXH     LOCCD,2,43     SAW COUNT GIVES COL 81 = 43.              03370
       TXH     LOCCG,2,42     WAS COL 80 PROCESSED.                     03380
LOCCD  PXD     0,0            CLEAR ACCUMULATOR.                        03390
       LGL     6              SHIFT NEXT CHARACTER INTO ACC.            03400
       TIX     LOCCE,2,14     COUNT DOWN BY 14                          03410
       LDQ     RECORD+3,2     LOAD NEXT WORD                            03420
       TXI     *+1,2,69       JUMP BACK COUNTER.                        03430
LOCCE  TXH     LOCCF,4,0      RETURN IF ALF MODE.                       03440
       PAX     0,1            MOVE CHR. INTO  INDEX  1                  03450
       TXH     LOCCF,1,48     TRA MEANS GOOD CHARACTER.                 03460
       TXH     LOCCC,1,47     TRA IF BLANK                              03470
       TXH     LOCCF,1,43     TRA IF GOOD CHARACTER.                    03480
       TXL     LOCCF,1,42     TRA IF GOOD CHARACTER.                    03490
       ZET     TAG            HERE ON $                                 03500
       TRA     PRINT          HERE ON $$  GO PRINT                      03510
       TRA     LOCCA          $ GOES TO TAG.                            03520
       REM                                                              03530
```

```
LOCCF  SXD     I,2             SAVE SAW COUNT                                 03540
       STQ     Q               SAVE UNUSED CHARACTERS.                        03550
       ADD     TAG             ATTACH $ SIGN IF PRESENT.                      03560
       STO     WORD            SAVE THE CHARACTER AT WORD.                    03570
       LDQ     SIGN            SIGN OF MQ NEGATIVE IF NEW CARD.               03580
       STZ     SIGN            CLEAR SIGN.                                    03590
       STZ     TAG             CLEAR TAG OF ANY $                             03600
       LXD     TEMP-17,4                                                      03610
       LXD     TEMP-10,2                                                      03620
       TRA     1,4             RETURN                                         03630
       REM PRINT OUT THE  $$ CARDS.                                           03640
PRINT  STQ     Q                                                              03650
       XEC*    $(TES)          CHECK FOR QUIET BUFFERS.                       03660
       XEC     READ.           FETCH NEXT CARD.                               03670
       LDQ     Q                                                              03680
       LGL     6               SPACE CONTROL SAFE IN ACC                      03690
       LDQ     BLANK                                                          03700
       AXT     4,4             FILL END OF OUTPUT                             03710
       STQ     OUTBUF+19,4      BUFFER WITH BLANKS.                           03720
       TIX     *-1,4,1                                                        03730
       LGR     6               SPACE CONTROL BACK TO MQ.                      03740
       STQ     OUTBUF          STORE SPACE CONTROL.                           03750
       AXT     14,4                                                           03760
       LDQ     RECORD+2,4                                                     03770
       STQ     OUTBUF+15,4                                                    03780
       TIX     *-2,4,1                                                        03790
       TSX     PRINX,4                                                        03800
       TRA     *+3                                                            03810
LOCCG  XEC     READ.           ALMOST ALWAYS A NOP.                           03820
       XEC*    $(TES)          WAIT FOR QUIET READ BUFFER.                    03830
       STZ     TAG             CLEAR THE $$ CHARACTERS.                       03840
       AXT     14,2            FETCH CARD.                                    03850
       LDQ     INBUF+14,2      14 WORDS                                       03860
       STQ     RECORD+2,2                                                     03870
       TIX     *-2,2,1                                                        03880
       CLA     TSXRD           OPEN READ. GATE                                03890
       STO     READ.                                                          03900
LOCCJ  AXT     84,2            CARD COL 1 IS 84                               03910
       CLS     =0              SET MINUS ZERO IN SIGN                         03920
       STO     SIGN                                                           03930
       LGL     12              SAVE COLUMN 79 AND 80
       LDQ     BLANK           BLANK OUT COLUMN 81 TO 84
       LGR     12              MAY HAVE LOOK AHEAD
       STQ     RECORD+1
       LDQ     RECORD-12                                                      03940
       TRA     LOCCA                                                          03950
       REM                                                                    03960
       REM          COMES HERE ON END OF FILE FLAG                            03970
 LOCCK LXD     TESTJK,4                                                       03980
       TXH     RTN,4,0         WAS DATA LOADED.  YES RTN                      03990
SGNOUT XEC*    $(TES)          WAIT FOR QUIET OUTPUT BUFFER                   04000
       AXT     6,4                                                            04010
       LDQ     OUT+6,4                                                        04020
       STQ     OUTBUF+6,4                                                     04030
       TIX     *-2,4,1                                                        04040
       AXT     13,4                                                           04050
       LDQ     BLANK                                                          04060
       STQ     OUTBUF+19,4                                                    04070
       TIX     *-1,4,1                                                        04080
       TSX     PRINX,4                                                        04090
       XEC*    $(TES)          WAIT FOR QUIET BUFFER.                         04100
LOCOUT CALL    $EXIT           THIS WAY OUT FOR KEEPS                         04110
       REM                                                                    04120
OUT    BCI     6,1END OF FILE INPUT TAPE JOB COMPLETE                         04130
       REM                                                                    04150
       REM END OF THE SAP SUBROUTINE CHRCTR.                                  04160
       EJECT                                                                  04170
       REM THIS IS SUBROUTINE CLEAR.  IT INITIALIZES                          04180
       REM NECESSARY PARAMETERS FOR SUBROUTINE STORE.                         04190
       REM                                                                    04200
 CLEAR SXD     J,0                     SET J TO 0.                            04210
       STZ     VAR             CLEAR VAR(1).                                  04220
       SXD     MSHIFT,0                RESET MSHIFT.                          04230
       TRA     1,4             RETURN TO CALLING PROGRAM                      04240
       REM                                                                    04250
       REM END OF THE SAP SUBROUTINE CLEAR.                                   04260
       REM                                                                    04270
       REM                                                                    04280
       REM THIS IS FUNCTION COMPAR.  IT EXAMINES THE CURRENT                  04290
       REM CHARACTER AND TESTS IT AGAINST THE CHARACTERS                      04300
       REM FOUND IN THE ARGUMENT. ALPHABETIC AND NUMERIC                      04310
       REM SPLITS ARE MADE IF THE CHARACTER IS NOT FOUND                      04320
       REM IN THE ARGUMENT. THESE TESTS ARE COUNTED AND                       04330
       REM THE NUMBER LEFT IN INDEX 2 CORRESPONDS TO THE                      04340
       REM SUCCESSFUL TEST. IF NO TEST IS SUCCESSFUL                          04350
       REM THEN INDEX 2 CORRESPONDS TO THE TOTAL TESTS +1.                    04360
       REM                                                                    04370
COMPAR LDQ     1,4             USE FIRST ARGUMENT IN CALLING                  04380
       AXT     1,2                                                            04390
LOCDA  PXD     0,0                                                            04400
       LGL     6               PULL IN 1ST TEST CHARACTER.                    04410
       TZE     LOCDD           DONE IF ZERO.                                  04420
       CAS     WORD            CHECK TEST WORD AGAINST CARD                   04430
       TXI     LOCDA,2,1       CHARACTER.                                     04440
       TRA     LOCDC           EQUAL.                                         04450
LOCDB  TXI     LOCDA,2,1       NOT EQUAL.  GET NEXT TEST                      04460
LOCDC  CLA     WORD            CHARACTER.                                     04470
       TRA     2,4             PROGRAM RETURN.                                04480
```

```
LOCDO  CLA     2,4           USE SECOND ARGUMENT IN THE CALLING         04490
       PDX     0,1           SEQUENCE (DECREMENT) AS THE TEST           04500
       TNX     LOCDC,1,1024  FOR ALPHABETIC-NUMERIC SPLIT.              04510
       SXD     LOCDF,1             BECOMES INCREMENT                    04520
       LXA     WORD,1              CHARACTER TO IRA                     04530
       TXL     LOCDC,1,9           NUMERIC                              04540
       TXH     LOCDF,1,57          SPECIAL 0 ZONE, $X                   04550
       TXH     LOCDE,1,49          ALPHABETIC 0 ZONE, NO /              04560
       TIX     *+2,1,32            KNOCK OFF 11 ZONE EXCEPT -           04570
       TIX     *+1,1,16            KNOCK OFF 12 ZONE EXCEPT +           04580
       REM     + AND - SIGNS WILL BE (16)10, / WILL BE (17)10           04590
       TXH     LOCDF,1,9           SPECIAL                              04600
 LOCDE TXI     LOCDF,2,-1          ADJUST IRB FOR ALPHABETIC            04610
 LOCDF TXI     LOCDC,2,**          ADJUST IRB FOR SPLIT                 04620
       REM                                                              04630
       REM END OF THE SAP SUBROUTINE COMPAR.                            04640
       EJECT                                                            04650
       REM THIS IS SUBROUTINE ERROR.  IT IS CALLED IF AN                04660
       REM ERROR WAS DETECTED ON ANY OF THE INPUT CARDS.                04670
       REM                                                              04680
 ERROR SXA     *+2,4         SAVE SOURCE                                04690
       XEC*    $(TES)        WAIT FOR QUIET BUFFERS                     04700
       AXT     **,4                                                     04710
       CLA     1,4           GET PRINT ARGUMENT                         04720
       STO     OUTBUF                                                   04730
       AXT     1,1                                                      04740
       CAS     R                                                        04750
       TRA     *+3           S THROUGH V                                04760
       TXI     *+1,1,-1      R                                          04770
 MES&A PXD     BLANK+4,1     A THROUGH N                                04780
       ANA     =7B17                                                    04790
       ARS     16                                                       04800
       ACL     MESSA                                                    04810
       STA     IO2.                                                     04820
       AXT     4,4                                                      04830
 IO2.  LDQ     **,4                                                     04840
       STQ     OUTBUF+5,4                                               04850
       TIX     *-2,4,1                                                  04860
       AXT     14,4                                                     04870
       LDQ     RECORD+2,4                                               04880
       STQ     OUTBUF+19,4                                              04890
       TIX     *-2,4,1                                                  04900
       TSX     PRINX,4                                                  04910
       XEC*    $(TES)        WAIT FOR QUIET BUFFER                      04920
       AXT     19,2                                                     04930
       CLA     BLANK                                                    04940
       STO     OUTBUF+19,2                                              04950
       TIX     *-1,2,1                                                  04960
       LDQ     =H     *      PICK UP *                                  04970
       LXD     I,2           SAW COUNT                                  04980
       TXL     *+2,2,71      BACK UP IF OVER 71                         04990
       TXI     *+3,2,-69                                                05000
       RQL     6             ROTATE ACCORDING TO CHR PART.              05010
       TIX     *-1,2,14      COUNT CHARACTER PART.                      05020
       STQ     OUTBUF+19,2   STORE ACCORDING TO RESIDUAL                05030
       TSX     PRINX,4       PRINT THE *                                05040
       XEC*    $(TES)        WAIT FOR THE *  TO BE PRINTED              05050
       LXD     ERSW,4        PICK UP ERROR SWITCH.                      05060
       TXL     LOCOUT,4,0    NON ZERO MEANS TRY NEXT SET                05070
       AXT     1208,4        BYPASS MARK                                05080
       SXD     BLANK,4       MARK BYPASSED CARDS                        05090
       LXA     NREG1,4       NONZERO IF THIS $DATA CARD.                05100
       TXL     *+2,4,0                                                  05110
       TSX     READ.+1,4     CRASH READ GATE IF $DATA CARD.             05120
LOCEB  TSX     CHRCTR,4      SKIP TO NEXT $DATA AND TRY THAT SET.       05130
       TQP     LOCEB                                                    05140
       SUB     =H0000$D                                                 05150
       TNZ     LOCEC         TRA NOT A $DATA CARD                       05160
       STQ*    NREG1-1       PUTS - SIGN IN TABLE(1)                    05170
       LXD     BLANK+7,4                                                05180
       SXD     BLANK,4                                                  05190
       LXD     INDX+2,4                                                 05200
       TRA     LOCA                                                     05210
LOCEC  ADD     =5            TEST FOR END FILE FLAG                     05220
       TZE     SGNOUT        END FILE.. GET OFF                         05230
       TRA     LOCEB         OTHER                                      05240
       REM                                                              05250
       REM ERROR MESSAGES. FIRST WORD ALSO USED AS A BLANK.             05260
 BLANK BCI     4,       REDUNDANCY CHECK                                05270
       BCI     4,ILLEGAL CHARACTER                                      05280
       BCI     4,NO MANTISSA BEFORE E.                                  05290
       BCI     4,NO ENTRY IN TABLE                                      05300
       BCI     4,$TYPE MISSING OR WRONG                                 05310
       BCI     4,EXPON. OUT OF RANGE                                    05320
       REM                                                              05330
       REM END OF THE SAP SUBROUTINE ERROR.                             05340
       EJECT                                                            05350
       REM THIS IS SUBROUTINE LOOK.  IT SEARCHES THE TABLE              05360
       REM FOR THE NAME STORED AT LOCATION VAR.  IF FOUND,              05370
       REM THE ACC IS NON-ZERO AT THE RETURN.                           05380
       REM                                                              05390
LOOK   SXD     TEMP-12,4     SAVE INDEX REGISTER C.                     05400
       CLA     J             SUBROUTINE.                                05410
       STD     LOCFE                                                    05420
       AXT     2,2     JK = 2 IN INDEX B                                05430
       AXT     1,1                 J1 = 1 IN INDEX A                    05440
LOCFA  CAL     **,2          CAL TABV(JK).                              05450
       TZE     LOCFG               NO ENTRY THIS VARIABLE               05460
       STD     LOCFD               DECREMENT HAS NEXT                   05470
       ACL     =0377777000000                                           05480
       ANA     =0377777000000      ENTRY LOC. SAVE DECR                 05490
       SUB     J             ONLY.   CHECK ENTRY LENGTH.                05500
       TNZ     LOCFD         IF NOT THE SAME, LOOK AT NEXT ENTR         05510
       PXD     0,2                                                      05520
       PDX     0,4           JM = JK IN INDEX C.                        05530
```

```
LOCFB  CLA     VAR+1,1       SEE IF VAR AND THIS
LOCFC  CAS     **,4               ENTRY AGREE
       TRA     *+2                IF SO, CHECK REST OF NAME
       TXI     *+2,4,1       RAISE JM BY ONE.
LOCFD  TXI     LOCFA-1,2,**    IF NOT SO, GO TO NEXT ENTRY.
       TXI     *+1,1,1       RAISE J1 BY ONE.
LOCFE  TXL     LOCFB,1,**    FINISHED IF J1 IS GREATER THAN J.
       TSX     CLEAR,4       CLEAR IF THE ENTRY AGREES.
 LOCFF CLA*    LOCFA
       STO     ILOC          SAVE COMMON INDEX AT ILOC.
LOCFG  LXD     TEMP-12,4     PREPARE TO RETURN.
       TRA     1,4           RETURN TO THE CALLING PROGRAM.
       REM
       REM
       REM END OF THE SAP SUBROUTINE LOOK.
       EJECT
       REM THIS IS SUBROUTINE NAME.  IT IS USED TO
       REM CORRELATE NAMES FROM INPUT CARDS WITH INTERNAL
       REM MEMORY LOCATIONS BY REFERRING TO THE TABLE.
       REM
       REM
NAME   SXD     TEMP-20,4     SAVE INDEX C.
       REM GET THE REST OF THE VARIABLE NAME. STOP AT ANY
       REM NON ALPHANUMERIC CHARACTER.
LOCGB  TSX     STORE,4
LOCGC  TSX     CHRCTR,4
       TQP     *+2           MINUS FOR NEW CARD
       TSX     TEST,4        COMMA MAY BE NEEDED.
       TNZ     *+3           LOOK FOR ZERO. IF ZERO, MAKE IT
       ACL     =H000000             A LETTER O
       STO     WORD
LOCGE  TSX     COMPAR,4
       BCI     1,=(0000
       TRA     *+5,2,1
       TRA     LOCGF         JUNK OR OPERATORS
       TRA     LOCGB         NUMERIC OR ALPHABETIC
       TRA     LOCGG         ( SIGN
       STZ     ILOC1         = SIGN
       REM GO TO THE TABLE LOOKUP ROUTINE IF AN = SIGN
       REM   OR AN OPERATOR WAS FOUND.
LOCGF  TSX     LOOK,4        FIND THE NAME IN TABLE.
       TZE     ERRT          NAME WAS FOUND IN TABLE IF NON-ZER
       LXA     ILOC,2
       TRA     LOCGL
       REM
       REM GO TO THE TABLE VARIABLE LOOKUP ROUTINE IF A
       REM ( SIGN WAS FOUND.
LOCGG  TSX     LOOK,4
       TNZ     LOCGJ
ERRT   TSX     ERROR,4
       BCI     1,0(T)
       REM CONVERT THE INDEX TO BINARY.
LOCGH  TSX     BINARY,4
       REM GET THE NUMERICS FOR THE INDEX TO THE VARIABLE.
LOCGJ  TSX     CHRCTR,4
       TXL     LOCGH,1,9     NUMERIC
       TXL     ERRC,1,27     JUNK
       TXH     ERRC,1,29     JUNK
       TSX     CHRCTR,4      ) SIGN. GET NEXT CHARACTER.
       TQP     *+2           MINUS FOR NEW CARD
       TSX     TEST,4        COMMA MAYBE NEEDED.
       TSX     COMPAR,4
       BCI     1,=00000
       TRA     *+4,2,1
       TRA     LOCGK         OPERATORS
       TRA     ERRL          ALPHABETIC AND NUMERIC
       STZ     ILOC1         = SIGN
       REM
LOCGK  CLA     VAR           COMPUTE STORING INDEX.
       ACL     ILOC
       PAX     0,2           STORE ADDRESS AT DECREMENT WITHOUT
       TXI     *+1,2,-1
LOCGL  SXD     JK,2          ACCUMULATOR OVERFLOW.
       CLA     ILOC1
       LXD     TEMP-20,4     RESTORE INDEX C.
       TRA     1,4           RETURN TO CALLING PROGRAM.
       REM     CONSTANTS AND ERROR CALL.
ERRC   TSX     ERROR,4
       BCI     1,0(C)
       REM
       REM END OF THE SAP SUBROUTINE NAME.
       EJECT
       REM THIS IS SUBROUTINE NUMBER.  IT IS USED TO
       REM ASSEMBLE NUMERIC DATA FROM CARDS.  ALL VALUES ARE
       REM TREATED AS FLOATING POINT NUMBERS IN THIS ROUTINE.
       REM
NUMBER SXD     TEMP-23,4     SAVE INDEX C.
       SXD     KNT2,4        INITIALIZE
       STZ     KNT3            THE SUBROUTINE
       STZ     KNT1              BRANCH PARAMETERS.
       STZ     KNT4
       STZ     TEMP
       TRA     LOCHB
       REM
LOCHA  TSX     CHRCTR,4
       TQP     *+2,          MINUS MEANS FROM NEW CARD
       TSX     TEST,4
```

```
LOCHB  TSX     COMPAR,4                                                06510
       BCI     1,.E0000                                                06520
       TRA     *+6,2,2                                                 06530
       TRA     LOCHK             JUNK OR AN OPERATOR                   06540
       TRA     ERRE              ALPHABETIC                            06550
       TRA     LOCHC             NUMERIC                               06560
       TRA     LOCHE             E                                     06570
       CLA     KNT2              DECIMAL POINT.                        06580
       TNZ     *+3               ZERO MEANS THIS IS THE SECOND POIN    06590
       TSX     ERROR,4                                                 06600
       BCI     1,0(N)                                                  06610
       STZ     KNT2                                                    06620
       STZ     NEXP                                                    06630
       TRA     LOCHA                                                   06640
LOCHC  CLA     NEXP              COUNT THE NUMBER OF DIGITS BEHIND     06650
       ADD     =1B35             THE.  IF THERE IS ONE                 06660
       STO     NEXP                                                    06670
 LOCHD LXA     KNT1,1                                                  06680
       TXH     LOCHD2,1,10               DO NOT ACCUMULATE PAST 10     06690
LOCHD1 TSX     BINARY,4          CONVERT THE DIGIT TO BINARY.          06700
       TZE     LOCHA             DO NOT COUNT LEADING ZEROS.           06710
LOCHD2 TXI     *+1,1,1           COUNT TOTAL NO. OF DIGITS             06720
       SXA     KNT1,1                                                  06730
       TRA     LOCHA                                                   06740
       REM COMES HERE WHEN THE EXPONENT FIELD IS                       06750
LOCHE  CLA     KNT1              ENCOUNTERED.                          06760
       TNZ     LOCHH             THERE MUST BE AT LEAST ONE DIGIT      06770
       TSX     ERROR,4           BEFORE THE E OF AN E FORMAT NUMBER    06780
       BCI     1,0(S)                                                  06790
LOCHF  CLA     KNT3              SEE IF EXPONENT DIGITS HAVE ARRIVE    06800
       TRA     *+2                                                     06810
LOCHG  CLS     KNT3              SEE IF EXPONENT DIGITS HAVE ARRIVE    06820
       TNZ     LOCHK-2           NON ZERO MEANS SIGN IS OPERATOR.      06830
       STO     TEMP              STORE SIGN OF EXPONENT.               06840
       CLA     KNT4                                                    06850
       TNZ     ERRF              NONZERO MEANS MORE THAN 1 EXP SIGN    06860
       SXD     KNT4,2            MAKE NOZERO.                          06870
LOCHH  TSX     CHRCTR,4                                                06880
       TQP     *+2,              MINUS MEANS FROM NEW CARD             06890
       TSX     TEST,4                                                  06900
       TSX     COMPAR,4                                                06910
       BCI     1,+-.000                                                06920
       TRA     *+7,2,2                                                 06930
       TRA     LOCHK-2           OTHERS                                06940
       TRA     ERRF              ALPHABETIC                            06950
       TRA     LOCHJ             NUMERIC                               06960
       TRA     ERRF              DECIMAL                               06970
       TRA     LOCHG             MINUS                                 06980
       TRA     LOCHF             PLUS                                  06990
       REM CONVERT THE EXPONENT TO BINARY.                             07000
LOCHJ  CLA     TEMP                                                    07010
       ALS     2                                                       07020
       ADD     TEMP                                                    07030
       ALS     1                                                       07040
       ACL     WORD                                                    07050
       STO     TEMP                                                    07060
       SXD     KNT3,2            RECORD FACT FOR SECOND SIGN.          07070
       TRA     LOCHH                                                   07080
       REM COMES HERE WHEN AN OPERATOR WAS FOUND.                      07090
       CLA     KNT3              TEST FOR THE PRESENCE OF EXPONENT.    07100
       TZE     ERRF              ZERO MEANS NO EXPONENT CAME.          07110
LOCHK  CLA     KNT2                                                    07120
       TZE     *+2                                                     07130
       STZ     NEXP                                                    07140
       CLA     KNT1              SEE IF MORE THAN TEN NUMBERS HAVE     07150
       SUB     =10B35                    BEEN CONVERTED                07160
       TPL     *+2               IF SO, USE THE DIFFERENCE IN THE      07170
       PXD     0,0               COMPUTATION OF THE EXPONENT.          07180
       SUB     NEXP                                                    07190
       ADD     TEMP                                                    07200
       STO     NEXP                                                    07210
       REM MANTISSA IN VAR AND THE EXPONENT IS IN NEXP.                07220
       CLA     VAR                                                     07230
       TZE     LOCHQ             SHORT CUT IF ZERO.                    07240
       LDQ     =0233000000000            CHARACTERISTIC FOR LOW BITS   07250
       LGR     8                         LOW 8 BITS TO MQ              07260
       RQL     8                                                       07270
       LRS     0                         BRING SIGN                    07280
       STQ     VAR                                                     07290
       ORA     =0243000000000            CHARACTERISTIC FOR HIGH BITS  07300
       FAD     VAR                                                     07310
       FRN                                                             07320
       STO     VAR                                                     07330
       CLA     NEXP              THE EXPONENT                          07340
       AXT     1,2                                                     07350
       LDQ     =1.               PUT 1 IN  MQ                          07360
 LOCHL LBT                       EXPONENT IN ACCU                      07370
       TRA     LOCHM          FOUND NO BIT.                            07380
       TXH     ERRV,2,6          EXPONENT EXCEEDS 63                   07390
       STO     VAR-2                                                   07400
       FMP     TAB+1,2        THIS FORMS 10 **NEXP                     07410
       XCA                    SAVE IN MQ                               07420
       CLA     VAR-2                                                   07430
 LOCHM ARS     1                                                       07440
       TZE     LOCHN           10**NEXP FINISHED.                      07450
       TXI     LOCHL,2,1                                               07460
 LOCHN TMI     LOCHO          MULTIPLY IF PLUS.                        07470
       FMP     VAR                                                     07480
       FRN                                                             07490
       TRA     LOCHQ                                                   07500
 LOCHO STQ     VAR-2                                                   07510
       CLA     VAR               DIVIDE IF NEXP IS MINUS.              07520
       FDP     VAR-2                                                   07530
       XCA                       ANSWER BACK TO THE ACCUM              07540
LOCHQ  LXD     TEMP-23,4         RESTORE INDEX C.                      07550
       TRA     1,4               RETURN TO CALLING PROGRAM.            07560
       REM THESE ARE THE ERROR CALLS FOR SUB NUMBR.                    07570
```

```
ERRE   TSX     ERROR,4                                                 07580
       BCI     1,0(E)                                                  07590
ERRF   TSX     ERROR,4                                                 07600
       BCI     1,0(F)                                                  07610
ERRV   TSX     ERROR,4                                                 07620
       BCI     1.0(V)                                                  07630
       REM                                                             07640
       REM THIS IS THE FLOATING PT. TABLE USED IN DBC                  07650
       DEC     1E+32,1E+16,1E+8,1E+4,1E+2      CONVERSION TABLE        07660
 TAB   DEC     1E+1                                                    07670
       REM                                                             07680
       REM END OF THE SAP SUBROUTINE NUMBER.                           07690
       EJECT                                                           07700


       REM THIS IS SUBROUTINE STORE.  IT STORES CHARACTERS             07710
       REM AT THE ARRAY VAR.                                           07720
       REM                                                             07730
STORE  SXD     TEMP-13,1       SAVE INDEX A.                           07740
       SXD     TEMP-14,2       SAVE INDEX B.                           07750
       LXD     J,1             PUT J INTO INDEX REGISTER A.            07760
LOCJA  LXD     MSHIFT,2        LOAD INDEX B WITH MSHIFT.               07770
       TIX     LOCJB,2,6       ADVANCE MSHIFT.                         07780
       AXT     36,2            REFRESH MSHIFT.                         07790
       TXI     *+1,1,1         RAISE J BY ONE IF MSHIFT IS OVER        07800
       STZ     VAR,1           CLEAR NEXT CELL                         07810
 LOCJB CLA     WORD            LEAVE SIGN BEHIND                       07820
       LDQ     =0                                                      07830
       LGR     42,2            MOVE CHARACTER                          07840
       STQ     TEMP-7          PLACES TO THE LEFT.                     07850
       CAL     TEMP-7                                                  07860
       ORS     VAR+1,1         STORE THE CHARACTER AT VAR.             07870
       SXD     MSHIFT,2        SAVE MSHIFT.                            07880
       SXD     J,1             SAVE J.                                 07890
       LXD     TEMP-13,1       RESTORE INDEX A.                        07900
       LXD     TEMP-14,2       RESTORE INDEX B.                        07910
       TRA     1,4             RETURN TO CALLING PROGRAM.              07920
       REM                                                             07930
       REM END OF THE SAP SUBROUTINE STORE.                            07940
       EJECT                                                           07950
       REM THIS IS SUBROUTINE TABLE.  IT IS USED TO                    07960
       REM CONSTRUCT A TABLE OF NAMES TO BE USED ON CARDS              07970
       REM AND THEIR MEMORY LOCATIONS RELATIVE TO ARG 2 OF             07980
       REM THE CALLING SEQUENCE.                                       07990
       REM                                                             08000
       REM                                                             08010
TABLE  SXD     TEMP-15,4       SAVE INDEX C.                           08020
       STZ     TEMP                                                    08030
LOCKA  TSX     CHRCTR,4                                                08040
       TSX     COMPAR,4                                                08050
       BCI     1,,00000                                                08060
       TRA     *+5,2,2                                                 08070
       TRA     LOCKD+1         JUNK                                    08080
       TRA     LOCKA           ALPHABETIC                              08090
       TRA     LOCKD+1         NUMERIC                                 08100
LOCKB  STZ     TEMP            COMMA                                   08110
       TRA     LOCKD                                                   08120
       REM COMES HERE TO CONVERT THE ADDRESS TO OCTAL FOR              08130
LOCKC  CLA     TEMP            THE TABLE.                              08140
       ALS     2                                                       08150
       ADD     TEMP                                                    08160
       ALS     1                                                       08170
       ACL     WORD                                                    08180
       STO     TEMP            ADDS TO MAGNITUDE                       08190
       REM                                                             08200
       REM COMES HERE TO GET NUMERICS.                                 08210
LOCKD  TSX     CHRCTR,4                                                08220
       TSX     COMPAR,4                                                08230
       BCI     1,.=/000                                                08240
       TRA     *+7,2,2                                                 08250
       TRA     ERRA            JUNK                                    08260
       TRA     ERRA            ALPHABETIC                              08270
       TRA     LOCKC           NUMERIC                                 08280
       TRA     LOCKT                 / CHARACTER                       08290
       TRA     LOCKF           = SIGN                                  08300
       REM                                                             08310
       REM COMES HERE IF A DECIMAL PT WAS FOUND.                       08320
 LOCKE CAL     TEMP            DECIMAL POINT                           08330
       CHS                     MAKE SIGN MINUS                         08340
       STO     TEMP                                                    08350
       TRA     LOCKD                                                   08360
       REM                                                             08370
       REM COMES HERE IF AN = SIGN WAS FOUND.                          08380
LOCKF  TSX     CLEAR,4                                                 08390
LOCKG  TSX     CHRCTR,4                                                08400
       TQP     *+2,            MINUS MEANS FROM NEW CARD               08410
       TSX     TESTT,4                                                 08420
       TZE     LOCKH                                                   08430
       TSX     COMPAR,4                                                08440
       BCI     1,/,0000                                                08450
       TRA     *+5,2,1                                                 08460
       TRA     ERRG            JUNK                                    08470
       TRA     LOCKJ           ALPHABETIC OR NUMERIC                   08480
       SXD     B,2             COMMA                                   08490
       SXD     B,2             SLASH                                   08500
       TRA     LOCKK                                                   08510
       REM                                                             08520
       REM COMES HERE TO STORE CHARACTER.                              08530
 LOCKH ACL     =H000000             REPLACE ZERO WITH LETTER O         08540
       STO     WORD                                                    08550
LOCKJ  TSX     STORE,4                                                 08560
       TRA     LOCKG                                                   08570
       REM COMES HERE AT END OF NAME.                                  08580
```

```
LOCKK  TSX     LOOK,4
       TNZ     LOCKR          GOES TO LOCKR IF THERE IS AN ENTRY
       LXD     J,1
       TXI     *+1,1,1        ASSEMBLE KEY
       PXD     0,1            IRB HAS FIRST FREE LOC.
       ACL     TEMP
 LOCKL SLW     **,2           STORE KEY INTO TABLE
       SXD     *+1,1          ADVANCE TO END
       TXI     *+1,2,**
 LOCKM CAL     VAR+1,1        MOVE NAME, 0 TO TABLE
 LOCKN XEC     LOCKL           SLW IN TABLE
       TNX     *+2,1,1        TRANSFER WHEN DONE
       TXI     LOCKM,2,-1     GO BACK TO FINISH
       ARS     34             KEEP ZONE OF 1ST VAR CH.
       TZE     ERRG           WAS NUMERIC, OR J=0
       REM
       REM     REEXAMINE THE  CUT OFF CHARACTER.
LOCKP  LXD     B,2
       TXH     LOCKB,2,1            COMMA
LOCKQ  LXD     TEMP-15,4      / CHARACTER
       TRA     1,4            RETURN.
       REM
       REM     COMES HERE TO REPLACE KEY
 LOCKR ANA     =0377777000000       J+1 IN DECREMENT
       ACL     TEMP                 LOCATION AND SIGN
 LOCKS XEC     LOCKL          SLW IN TABLE
       TRA     LOCKP
 LOCKT CAL     TEMP                 IS / LEGAL
       TZE     LOCKQ                YES
       REM TRA     ERRA                 NO, NUMERICS WAITING
       REM
       REM THESE ARE THE ERROR CALLS.
ERRA   TSX     ERROR,4
       BCI     1,0(A)
ERRG   TSX     ERROR,4
       BCI     1,0(G)
       REM
       REM END OF THE SAP SUBROUTINE TABLE
       EJECT
       REM THIS IS SUBROUTINE TEST. IT LOOKS AHEAD TO CLASSIFY
       REM A NEW CARD. ACOMMA WILL BE PUT INTO THE CURRENT
       REM CHARACTER POSITION ONLY IF EITHER (1) THE NEXT
       REM CARD BEGINS WITH A $ SIGN FOLLOWED BY SOME OTHER
       REM CHARACTER OR (2) THE NEXT CARD BEGINS WITH AN
       REM ALPHABETIC AND AN = SIGN IS FOUND AND IT PRECEEDS
       REM     ALL , $ AND . CHARACTERS ON THAT CARD.
       REM
       REM
TEST   SKD     TEMP-12,4      SAVE INDEX FOR RETURN.
       SUB     =H0000$0       TEST FOR A $  SIGN.
       TPL     LOCLA          POSITIVE MEANS  $ SIGN.
       XEC     READ.          SAFE TO REFILL BUFFER
       TXL     LOCLB,1,16     NUMBERS AND SPECIAL
       TIX     *+1,1,33       FIX SO SLASH IS SPECIAL
       TIX     *,1,16         MOD OUT ZONE
       TXH     LOCLB,1,9      SPECIALS
       REM                    ALPHABETIC COME THRU.
LOCLC  AXT     15,1           SCAN THE CARD.
       LDQ     RECORD+3,1
       TXI     *+1,1,69       FOR CHARACTER COUNT
       TXL     LOCLB,1,70     DONE IF WHOLE CARD SCANNED
LOCLD  PXD     0,0            OK TO SEARCH 84  COLUNMS
       LGL     6
       PAX     0,2            ZONE TO IRB
       ANA     =017           KEEP DIGIT
       SUB     =013           DIGIT PART OF ,$.= CHR.
       TZE     LOCLJ                CHECK ZONE
       TIX     LOCLD,1,14     TRY NEXT CHARACTER
       TRA     LOCLC+1
 LOCLJ TXH     LOCLB,2,15           , . $ NEED NO COMMA
LOCLA  AXT     84,1           84 IS CARD COL 1
       SXD     I,1            RESET CHRCTR TO BEGIN CARD
       CLA     RECORD-12
       STO     Q
       CLA     =H00000,       SUBSTITUTE A COMMA.
       STO     WORD
LOCLB  LXD     TEMP-12,4
       CLA     WORD           IN AC FOR SR NAME, TABLE
       TRA     1,4            RETURN TO THE CALLING PROGRAM.
       REM
       REM        FOR TABLE SUB STATEMENTS
 TESTT SXD     TEMP-12,4
       AXT     84,4           IF NEXT CARD HAS VALID
       LDQ     RECORD-12      LEFT PART OF SUBSTATEMENT
 LOCNB PXD     0,0
       LGL     6
       PAX     0,2
       TXH     LOCLB,2,48     0 ZONES EXCEPT BLANK
       TXH     LOCNC,2,47     BLANK
       TXH     LOCLB,2,27     11 ZONES AND )
       TXH     LOCLA,2,26     .
       TXH     LOCLB,2,11     12 ZONES AND 8-4
       TXH     LOCLA,2,10           =
LOCNC  TIX     LOCNB,4,14     NUMERICS AND BLANK
       LDQ     RECORD+3,4
       TNX     LOCLB,4,1
       TXI     LOCNB,4,70
       REM
       REM END OF THE SAP SUBROUTINE TEST.
       EJECT
       REM THE FOLLOWING FOUR SUBROUTINES ARE USED TO
       REM CONVERT DECIMAL DIGITS TO BINARY  IN VAR,
       REM FIX FLOATING POINT NUMBERS, FLOAT FIXED POINT
       REM     NUMBERS, AND  FORM ARITHMETIC RESULTS IN THE
       REM     PSEUDO ACCUMU LATOR (ACC) FOR EACH OPERATION
       REM     ON A CARD.
       REM
```

```
BINARY CLA     VAR           ACCUMULATE A SERIES OF BASE 10
       ALS     2             DIGITS IN BINARY IN VAR.
       ADD     VAR
       ALS     1
       ACL     WORD
       STO     VAR
       TRA     1,4
       REM
FLT    CLA     TEMP          CONVERT TO FLOATING POINT THE
       LRS     18            CONTENTS OF THE STORAGE CALLED
       ORA     =0233000000000      TEMP.
       FAD     =0233000000000
       STO     TEMP          LEAVE THE ANSWER IN TEMP.
       TRA     1,4
       REM
 FIX   UFA     =0233000000000      CONV TO FIXED PT THE CONT
       LRS     0             OF THE ACCUMULATOR.
       ANA     =0377777
       LLS     0
       ALS     18            LEAVE THE FIXED POINT NUMBER IN
       TRA     1,4           THE ACCUMULATOR.
       REM
ACCUN  LXD     OPER,2        BRANCH FOR OPERATOR
       STZ     OPER          PREPARE FOR NEXT OPERATOR.
       CLA     TEMP
       TRA     *+5,2
       TRA     LOCMB         *
       TRA     LOCMA         /
       CHS                   MINUS
       FAD     ACC           PLUS
 ACCUN STO     ACC                 NONE
       TRA     1,4
       REM
LOCMA  CLA     ACC           DIVIDE.
       FDP     TEMP
       STQ     ACC
       TRA     1,4
       REM
LOCMB  LDQ     ACC           MULTIPLY.
       FMP     TEMP
       TRA     ACCUN
       REM
       REM
       REM END OF THE SAP SUBROUTINES ACCUM, FIX, FLOAT.
       EJECT
       REM SUBROUTINE PRINX DRAINS PRINT BUFFER TO LOGICAL TAPE
       REM                   GIVEN IN DECREMENT OF  OUTAPE
 PRINX SXA     PRINY,4             BUFFERED WRITE ROUTINE
       CLA     OUTAPE        LOGICAL OUTPUT TAPE NUMBER
       CALL    (IOS)
       AXC     IOCD,4
       XEC*    $(WRS)              =WTDL 6
       XEC*    $(RCH)              =RCHL 0,4
       PXA     0,4                 SAVE LOC OF
       STA*    $(WTC)              IO COMMAND FOR (WER)
       CLA     TSXWR               PRESET END ACTION
       STO*    $(TES)
 PRINY AXT     **,4
       TRA     1,4
 TSXWR CALL    (WER)               EXECUTED FROM (TES)
 IOCD  IORT    OUTBUF,,19
       EJECT
       REM SUBROUTINE READ. FILLS READ BUFFER FROM LOGICAL TAPE 7.
 READ. TSX     READ.+1,4           READ. GATE INITIALLY OPEN
       CLA     =0076100000000      CLOSE READ GATE
       STO     *-2
       SXA     AXT,4
       CLA     INTAPE        LOGICAL INPUT TAPE NO.
       CALL    (IOS)
       AXC     IOCD.,4
       XEC*    $(RDS)              =RTDL5
       XEC*    $(RCH)              =RCHL 0,4
       CLA     TSXTS               SET UP BUFFER TEST
       STO*    $(TES)
 AXT   AXT     **,4
       TRA     1,4
 TSXTS TSX     *+1,4               EXECUTED FROM (TES)
       SXA     AXT,4               CLOSE OUT BUFFER
       CLA     =0076100000000      SAY BUFFER IS QUIET
       STO*    $(TES)
       AXT     5,4                 PRESET REDUNDANCY
       SXA     RTT,4               COUNT
 TSXTT AXC     *+1,4
       XEC*    $(TCO)              =TCOL 0,4
       AXC     RTT,4
       XEC*    $(TRC)              =TRCL 0,4
       AXC     XIT.,4
       XEC*    $(TEF)              =TEFL 0,4 JOB COMPETE
       TRA     AXT                 RETURN
 RTT   AXT     **,4                INTERROGATE COUNT
       TIX     SXA.,4,1            GIVE ANOTHER TRY
       AXT     14,4                CARD SURE BAD
       CLA     INBUF+14,4          SAVE IMAGE
       STO     RECORD+2,4
       TIX     *-2,4,1
       AXT     84,4          MAKE ERROR ROUTINE LOSE*
       SXD     I,4                 IN INPUT BUFFER
       TSX     ERROR,4
 R     BCI     1,0(R)
 SXA.  SXA     RTT,4               SAVE COUNT
       AXC     *+2,4
       XEC*    $(TEF)              TURN OFF EOF IND
       XEC*    $(BSR)              BACKSPACE,
       AXC     IOCD.,4
       XEC*    $(RDS)              REREAD
       XEC*    $(RCH)
       TRA     TSXTT
```

```
IOCD.  IORT    INBUF,,14                                               10710
XIT.   CLA     =0531700000000      FILE FLAG                            10720
       STO     INBUF                                                    10730
       TRA     AXT                                                      10740
       EJECT                                                            10750
       REM BUFFERS AND COMMON STORAGE ASSIGNMENT                        10760
INBUF  BCI     7,$E                                                     10770
       BCI     7,                                                       10780
OUTBUF BCI     7,                                                       10790
       BCI     7,                                                       10800
       BCI     5,                                                       10810
       REM ************* COMMON STORAGE MAP                             10820
       COMMON  -203                 MOVE TO TOP-3                       10830
RECORD COMMON  13                  CD IMG *-12 TO*+1                    10840
I      COMMON  1                   CHARACTER POINTER FOR CARD           10850
ERSW   COMMON  1             ERROR SWITCH ZERO = GET OFF                10860
Q      COMMON  1                   UNTESTED CHARACTERS                  10870
WORD   COMMON  1                   CURRENT CHARACTER                    10880
OPER   COMMON  1                   DECR. REPRESENTS OPER.               10890
B      COMMON  1                   TEMP INDEX IN SUB TABLE              10900
NEXP   SYN     8                   NUMERIC VALUE OF EXPONENT            10910
J      COMMON  1                   COUNTER IN SUB STORE                 10920
MSHIFT COMMON  1                   COUNTER IN SUB STORE                 10930
ILOC   COMMON  1                   DATA BROUGHT FROM TABLE              10940
TEMP   COMMON  1                   TEMPORARY STORAGES                   10950
KNT1   COMMON  1                   COUNTER TOTAL DIGITS                 10960
KNT2   COMMON  1                   NONZERO UNTIL . FOUND                10970
KNT3   COMMON  1                   ZERO UNTIL DIGIT IN EXP              10980
SIGN   COMMON  1                   MINUS IF SUB CHAR READS C            10990
TAG    COMMON  1                   SAVES $ IN SUB CHAR                  11000
ALF    COMMON  1                   NONZERO MEANS ALF MODE               11010
       COMMON  1                                                        11020
JK     COMMON  1                   SUBSCR CORR TO NAME                  11030
       COMMON  13                                                       11040
JK1    COMMON  1                   CURR SUBSCR OF LEFT SIDE             11050
       COMMON  1                                                        11060
ACC    COMMON  1                   PSEUDO ACCUMULATOR                   11070
ILOC1  COMMON  1                   ILOC FOR LEFT SIDE                   11080
KNT4   COMMON  1                   NONZERO AFTER EXP SIGN               11090
VAR    COMMON  153                 SPACE FOR NAMES, ETC.                11100
       END                                                              11110
```

```
       COUNT   1
       REM ONE WORD SUBROUTINE TO CONTROL BACKSPACE OF INPUT TAPE
       LBL     LINK,6
       ENTRY   (LINK)
(LINK) NOP
       END
```

APPENDIX I

EXAMPLE II: LUNAR ORBITING PROBE

This example of a lunar orbiting probe illustrates the use of the ephemeris tape and the control parameters needed to consider the effects of perturbing bodies, atmospheric forces, oblateness, rotating Earth, and thrust. No effort was made to optimize this trajectory but rather to use plausible values for illustrative purposes. It is suggested that the input instructions contained in appendix G be read prior to the following detailed discussion.

Suppose the probe was launched at Cape Canaveral on December 7, 1961, by a three-stage vehicle with stage parameters as shown in the following table:

Parameters	Stage		
	1	2	3
Initial mass, m_0, kg	150,000	52,500	23,625
Engine exit area, A_e, m^2	3.0	1.0	.5
Vacuum specific impulse, I, sec	420	420	420
Propellant flow rate, $-\dot{m}$, kg/sec	750	125	56.25
Burning time, t_b, sec	117	207.9	370
Aerodynamic reference area, S, m^2	7.5	4.0	2.0

Figure 4 shows the assumed variation of $C_{D,0}$, $C_{D,i}$, and C_L with Mach number as well as the angle-of-attack schedule.

The vehicle will be flown as follows: First, there will be a short nondrag vertical flight, after which the desired velocity orientation will be set, and then a turn will be executed determined by gravity and the angle-of-attack schedule until first-stage burnout. The second and third stages follow a continuation of the same turning pattern. The third stage will be powered until the eccentricity of the trajectory equals 1.10. It will then coast until it is at lunar pericenter, at which time the engine will again be turned on (with $\alpha = \pi$) until the orbit about the Moon becomes nearly circular.

The chosen integration mode will be rectangular for the powered flight, but the mode of orbit elements will be used for the coast portions. Other bodies considered besides the Earth and the vehicle are the Sun, the Moon, and Jupiter. Jupiter is included to illustrate the use of ellipse ephemerides. The Sun and Moon will illustrate the use of the tape ephemeris.

The correct firing direction and launch time remain to be determined. This determination can be made by finding approximate values and then adjusting these values after one or more shots are fired. The adjustments could be made by an iteration scheme programed internally to make a closed system. For this example, however, they were made by hand by firing several shots at various azimuth angles close to an estimate obtained by using reference 12 and an ephemeris. From a plot of the z-direction cosine of the vehicle-Moon distance against vehicle-Earth distance, the azimuth angle that will intersect the Moon orbit can be determined. The correct launch time is found by using the previously determined azimuth angle and various times of day to determine the time of day at which the vehicle intersects the correct position in the Moon orbit (location of the Moon). This type of analysis gives an azimuth angle of about 64.5° and a time of day of about 7.0^h E.T. (E.T. is ephemeris time, which is approximately equal to Greenwich mean time.) For the present purpose, these values will be used.

The program begins by constructing the merged ephemeris tape for the Sun and Moon. This is done by SUBROUTINE TAPE in conjunction with the input shown as follows:

```
$DATA=300,$TABLE,2=TAPE3,17=ELIST,29=TBEGIN,30=TEND/ $$ ID. AND TABLF DEFINITION
TAPE3=0                        $$  NECESSARY TO MAKE TAPE
ELIST=(A3)SUN,(A4)MOON         $$  LIST OF DESIRED EPHEMERIS BODIES
TBEGIN=2437640.5               $$  JULIAN BEGINNING DATE
TEND=TBEGIN+5                  $$  JULIAN ENDING DATE
```

After the merged ephemeris tape is constructed, the set of standard data in SUBROUTINE STDATA is loaded, and the input is loaded as shown:

```
$DATA=1,$TABLE, 33=DTOFFJ, 34=TOFFT, 711=TIME, 716=X, 717=Y, 718=Z,
713=VX, 714=VY, 715=VZ, 11.=IMODE, 713=E, 714=OMEGA, 715=NODES,
716=INCL, 717=MA, 718=P, 43=LAT, 44=LONG, 45=AZI, 46=ELEV, 14=ALT,
47=VEL, 16=TFILE, 28=TMIN, 153=BODYCD, 177=ELIPS, 30.=MODOUT,
27=STEPS, 29=DELMAX, 26=STEPMX, 23=EREF, 24=ERLIMT, 4.=NSAVE,
5=RECALL, 3=CLEAR, 18.=LOOKX, 22=XLOOK, 19.=LOOKSW, 20=SWLOOK,
609.=INLOOK, 15=END, 31=ATMN, 32=RATM, 49=ROTATE, 417=COEFN,
163.=ICC, 60=BETA, 50=OBLATN, 73=TB, 93=FLOW, 103=SIMP, 123=AREA,
143=DELT, 83=RMASS, 113=AEXIT, 133.=IDENT, 48.=LSTAGE, 25=TKICK /
```

```
BODYCD=(A5)EARTH,(A4)MOON,(A6)JUPITE,(A3)SUN   $$ BODY NAMES, 1ST IS ORIGIN
ELIPS=(ALF6)JUPITE,(ALF3)SUN,.9547861E-3,4.81E+10,5.1913995,   $$ ELLIPTIC DATA
.0486288,.1765935,.056971884,.40587194,2433964.,.6664,4333.7153$$ FOR JUPITER
COEFN=0,.4,0,.6,1,1.15306,-.16326,.010204,8,.5,,,100,,10,,,  $$ AERO. COEFF.
100,,.025,,,100,,,,,15,-.6,.04,,40,.7,,,117,,,,1E3,180,,,1E8,ICC=24,14,19,1 $$
RMASS=150000,52500,23625         $$ STAGE MASSES
FLOW=750,125,56.25,0,56.25       $$ STAGE FLOW RATES
SIMP=420,420,420,,420            $$ STAGE SPECIFIC IMPULSES
AEXIT=3,1,.5                     $$ STAGE ENGINE EXIT AREAS
AREA=7.5,4,2                     $$ STAGE AERODY. REFERENCE AREAS
TB=117,207.9,400,1E7,50,3600     $$ STAGE BURNING TIMES
DELT=2,2,2,86400,2,600           $$ STAGE INITIAL INTEGRATION STEP SIZES
IDENT=,2,3,4,5,6                 $$ STAGE INPUT IDENTIFICATION NUMBERS
DTOFFJ=2437640.5,TOFFT=7/24      $$ TAKE-OFF DATE AND FRACTION OF DAY
LAT=28.280,LONG=-80.571,ELEV=81.7$$ LATITUDE,LONGITUDE,ELEVATION
AZI=64.5,ALT=10,IMODE=4          $$ AZIMUTH,ALTITUDE,INTEGRATION MODE
MODOUT=2,DELMAX=50               $$ MODE OF OUTPUT, TIME INTERVALS OF OUTPUT
STEPMX=300, OBLATN=(A5)EARTH     $$ MAXIMUM ALLOWED STEP NUMBER, OBLATE BODY
TKICK=10                         $$ TIME OF THE VERTICAL NON-DRAG STEP
ROTATE=7.29211585E-5             $$ ROTATION RATE OF THE ORIGIN BODY (EARTH)
ATMN=(A5)EARTH,RATM=1E11         $$ ATMOSPHERE NAME, RADIUS OF ATMOSPHERE
EREF=1E-5,ERLIMT=5E-5            $$ REFERENCE ERROR, LIMIT ERROR
```

SUBROUTINE ORDER reorders the list of bodies putting the Sun before Jupiter (i.e., the Sun's position relative to the vehicle must be found before Jupiter's relative position can be computed). The elliptic data for finding Jupiter's position are relocated according to the computed body list. The gravitational constants, μ and $\sqrt{\mu}$, are then calculated. The atmosphere belongs to the body at the origin (Earth) so that the rotation rate and atmospheric radius are set. The final duty of ORDER is to position the merged ephemerides tape at the beginning of the correct ephemeris. In this case, only one merged ephemeris was constructed; nevertheless, it still must be identified and spaced to the beginning of the data.

The main program now calls SUBROUTINE STAGE, which is responsible for controlling the sequencing of the stages for the flight. The data for the first stage are set into their proper locations as in example I. Before calling SUBROUTINE NBODY, however, SUBROUTINE TUDES is called (since IMODE = 4) to transform the Earth-centered spherical coordinates into rectangular coordinates. In addition, TUDES computes the closed-form solution for the initial vertical non-drag step. The path is integrated from this point on, where the initial orientation is specified by the spherical coordinates. The small error introduced by this procedure is offset by avoiding the complications associated with integrating the takeoff. One such difficulty is the thrust-direction specification when the velocity is zero, especially if the origin body is rotating.

The SUBROUTINE STAGE then calls upon SUBROUTINE NBODY to perform the integration of the first-stage path. The derivatives are supplied by SUBROUTINE EQUATE which, in turn, calls upon SUBROUTINES EPHMRS, ICAO, AERO, THRUST, and OBLATE. SUBROUTINE EPHMRS computes the perturbations that result from bodies other than the origin body. The positions of the Sun and Moon are determined through use of the merged ephemeris tape, while the position of Jupiter is determined by SUBROUTINE ELIPSE, which uses the ellipse data loaded on input cards.

SUBROUTINE AERO determines the aerodynamic accelerations through use of quadratic equations for the lift and drag coefficients and SUBROUTINE ICAO, which determines density, pressure, and temperature as functions of altitude. SUBROUTINE THRUST computes the thrust magnitude as a function of ambient pressure (eq. (4)) and then determines the thrust orientation relative to the x,y,z axes. Oblateness accelerations are determined in SUBROUTINE OBLATE.

The integration of the first-stage path is terminated by SUBROUTINE STEP when $t = 117$ seconds. Control is then returned to STAGE whereupon the second-stage data are set in place for integration of the second-stage path, which is terminated by STEP when $t = 324.9$ seconds.

It is required that the third-stage engine cease operating when the eccentricity of the path equals 1.10. Let it also be required that the output occur only every 100 seconds instead of every 50 seconds, as during stages one and two. These results may be obtained by placing the following cards after the previous set:

```
$D=3,DELMAX=100                               $$ OUTPUT EVERY 100 SEC. FOR STAGE 3
LOOKX=1226,XLOOK=1.1,END=1,INLOOK=30   $$  STOP WHEN ECCENTRICITY = 1.1
```

These cards are read into the computer after the third-stage data are set in place (since IDENT (3) = 3), but before integration of the third-stage path begins. The third-stage burning time, 400 seconds, was purposely set high enough to allow sufficient time for the eccentricity to reach the value 1.10.

To illustrate the input facilities, the coasting third stage will be called the fourth stage, the reverse-thrust portion at the Moon a fifth stage, and the final coast portion about the Moon a sixth stage. Under these conditions, it becomes necessary to determine the initial masses of these stages. The initial masses of the fourth and fifth stages may be computed on the following input card, which is placed after the previous cards:

```
$D=30,$T,712=MASS/ RMASS(4)=MASS,MASS    $$  COMPUTE MASSES FOR STAGES 4 AND 5.
```

This card will be read into the computer immediately following the condition of $e = 1.10$, since INLOOK = 30.

The fourth-stage path is to be integrated in orbit elements with less output than previously required. The following cards are sufficient for these purposes:

```
$D=4,IMODE=-2, MODOUT=3, DELMAX=21600 $$ INTEGRATE IN ORBIT ELEMENTS FOR STAGE 4
STEPS=10, TMIN=86400 $$ OUTPUT EVERY 6 HOURS TILL 1 DAY, THEN EVERY 10 STEPS
```

About a half day later, the vehicle is close enough to the Moon so that the coordinate-system origin is translated to the Moon. This translation is accompanied by a shift to rectangular integration variables, since the vehicle is approaching the Moon far out on a hyperbolic asymptote. After the origin is translated to the Moon, an input card may be read to cause termination of the fourth stage when the true anomaly about the Moon is zero:

```
$D=101,LOOKX=1150, XLOOK=0  $$ INTEGRATE UNTIL TRUE ANOMALY IS ZERO
```

The fifth-stage path is integrated with reverse thrust until $e = 0.05$. The angle α is computed to be π by using the COEFN array. The termination of stage five is caused by the cards:

```
$D=5,LOOKX=1226,XLOOK=.05,INLOOK=50       $$ STAGE 5, REVERSE THRUST UNTIL E=.05
IMODE=-1  $$  SWITCH TO RECTANGULAR COORDINATES
```

The sixth-stage mass is computed by the card:

```
        $D=50,RMASS(6)=MASS      $$ COMPUTE MASS FOR STAGE 6.
```

In addition, output is desired at 10-minute intervals, which may be accomplished with the card:

```
$D=6,MODOUT=2,DELMAX=600,IMODE=-2       $$ OUTPUT EVERY 10 MINUTES FOR STAGE 6.
```

The last step output is reproduced as follows:

```
STEP=  181. + 18.    ECCENTRICITY= 5.00567108E-02 OMEGA=-2.49327761
TIME= 114425.91      SEMILATUS R.= 2127902.28      TRU A=-3.08236283
JDAY= 2437642.1157  MEAN ANOMALY=-3.07620674       NODE=-1.65888368
ALFA= 180.00000        PATH ANGLE=-0.17870317      INCL= 0.77642872
EARTH  R= 3.7524622E 08  0.099065  0.940877  0.323939
JUPITE R= 8.4530520E 11  0.579731 -0.742897 -0.334688

         V= 1441.76900      R= 2239823.88      REFER=MOON   ORBIT  1
        VX= 859.314598      X= 884837.336      RMASS= 3004.18549
        VY= 868.288918      Y=-1786946.89      REVS.= 0.82317527
        VZ= 765.735352      Z= 1020144.20       DELT= 462.857422
SUN      R= 1.4697769E 11 -0.240123 -0.890528 -0.386394
```

REFERENCES

1. Vienop, Edna, and Brady, Joseph L.: The Themis Code: An Astronomical Numerical Integration Program for the IBM 704. UCRL 5242, Radiation Lab., Univ. Calif., May 20, 1958.

2. Holdridge, D. B.: Space Trajectories Program for the IBM 7090 Computer. TR 32-223, Jet Prop. Lab., C.I.T., Mar. 2, 1962.

3. Technical Staff of Radiation, Inc., ed.: Space Trajectories. Academic Press, Inc., 1960.

4. Strack, William C., Dobson, Wilbur F., and Huff, Vearl N.: The N-Body Code - A General FORTRAN Code for Solution of Problems in Space Mechanics by Numerical Methods. NASA TN D-1455, 1963.

5. Wilson, Raymond H., Jr.: A Gravitational Force Function for the Earth Representing All Deviations From a Spherical Geoid. Franklin Inst. Jour., vol. 268, no. 5, Nov. 1959, pp. 378-387.

6. Baker, R. M. L., Jr., et al.: Efficient Precision Orbit Computation Techniques. ARS Jour., vol. 30, no. 8, Aug. 1960, pp. 740-747.

7. Dobson, Wilbur F., Huff, Vearl N., and Zimmerman, Arthur V.: Elements and Parameters of the Osculating Orbit and Their Derivatives. NASA TN D-1106, 1962.

8. Turner, Don N., and Huff, Vearl N.: An Input Routine Using Arithmetic Statements for the IBM 704 Digital Computer. NASA TN D-1092, 1961.

9. Clarke, V. C., Jr.: Constants and Related Data Used in Trajectory Calculations at the Jet Propulsion Laboratory. TR 32-273, Jet Prop. Lab., C.I.T., May 1962.

10. Dobson, Wilbur F., Mackay, John S., and Huff, Vearl N.: Starting Conditions for Nonoscillatory Low-Thrust Planet-Escape Trajectories. NASA TN D-1410, 1962.

11. Kopal, Zdeněk: Numerical Analysis. John Wiley & Sons, Inc., 1955.

12. Weber, Richard J., Pauson, Werner M., and Burley, Richard R.: Lunar Trajectories. NASA TN D-866, 1961.

TABLE I. - ORBIT ELEMENTS AND OTHER DATA FOR ELLIPSE EPHEMERIDES

[Epoch: Sept. 23, 1960; Julian day, 2437 200.5; mean equinox and equator, 1950.0.]

Name	Reference	Mass, sun mass units	Radius of influence sphere, m	Semilatus rectum, AU	Eccentricity	Argument of pericenter, radians	Longitude of ascending node, radians	Inclination, radians	Julian day of perihelion passage	Fractional day of perihelion passage	Period, mean solar days
Mercury	Sun	1/6,120,000	10^8	0.3707315	0.205627	1.1679154	0.1896133	0.49924366	2437 163	0.283386	87.969252
Venus	Sun	1/466,645	6.14×10^8	.72329863	.006792	2.1567353	.13931743	.42703751	2437 132	.682782	224.70087
Mars	Sun	1/3,088,000	5.78×10^8	1.5104078	.093369	5.7966845	.058500499	.4310002	2437 081	.09531	686.97964
Jupiter	Sun	1/1047.39	4.81×10^{10}	5.1913995	.0486288	.1765935	.056971884	.40587194	2433 964	.6664	4333.7153
Saturn	Sun	1/3500	5.46×10^{10}	9.5554288	.0509895	1.4938359	.10416467	.39404007	2431 246	.5163	10,829.478
Uranus	Sun	1/22,869	5.17×10^{10}	19.100903	.0457866	2.9848628	.032257032	.41321621	2409 019	.272	30,587.016
Neptune	Sun	1/18,889	8.61×10^{10}	30.197622	.0045616	.3302296	.061416599	.38947933	2404 118	.842	60,612.183
Pluto	Sun	1/400,000	3.81×10^{10}	36.969138	.2502358	3.1999771	.76630286	.41231716	1639 376	.44	904,658.99
Sun	Earth	1.0	10^{20}	.99972025	.016716	4.923277	0	.4092062	2436 937.1	0	365.256

TABLE II. - PROGRAM CONTROL PARAMETERS

Control variables	COMMON location	Possible values	Setting	Description of use
ASYMPT	A(7)	0.0 or 1.0	Internal	Normally equal to 0.0; set equal to 1.0 in SUBROUTINE EQUATE when Kepler's equation fails to converge for $e > 1$, and then used to control branching in NBODY for IMODE = 3.
ATMN	A(21)	Any ALF coded body name	Input	Contains name of body that is to have an atmosphere. Causes SUBROUTINE AERO to be called in SUBROUTINE EQUATE if object is within that atmosphere.
CLEAR	C(3)	Any value	Input	If CLEAR = 0, SUBROUTINE STDATA is called from MAIN; if CLEAR $\neq$ 0, SUBROUTINE STDATA is bypassed. STDATA clears the A, XPRIM, and XPRIMB arrays.
CONSTU	A(32)	>0, $\sim 10^{-8}$ to $\sim 10^{-2}$ radian	STDATA: 10^{-6} input	Controls branching in SUBROUTINE EQUATE, which determines how accurate eccentric anomaly will be computed by Kepler's equation.
CONSU	A(31)	>0, $\sim 10^{-8}$ to $\sim 10^{-2}$ radian	STDATA: 10^{-6} input	Similar to CONSTU except that it is used in SUBROUTINE ELIPSE for perturbing bodies instead of object.
DELMAX	A(19)	Any number of seconds	Input	If MODOUT = 2 or 3, output is given only at intervals of DELMAX.
END	A(5)	Any number	Input	Used when LOOKX $\neq$ 0. After the condition C(LOOKX) = XLOOK is met, control is sent to main program if END < 0, to complete the stage in process if END = 0, or begin integrating the next stage if END > 0.
EREF	A(13)	Any number	STDATA: 10^{-6} input	Desired error value. Error control predicts step size such that E2 ~ EREF. If EREF < 0, it will be treated as +EREF; however, error data will be recorded and printed.
ERLIMT	A(14)	Any plus number	STDATA: 3×10^{-6} input	Maximum error value that allows step in question to be passed as good step. If E2 > ERLIMT, step is recomputed with smaller step size.
ETOL	A(30)	Positive number of order 0.01	STDATA: 0.01 input	If eccentricity falls in region 1±ETOL and integration is in orbit elements, integration mode is switched to temporary rectangular until eccentricity falls outside this region.
FILE	B(22)	Any plus integer	Internal	Set equal to 10.0 in SUBROUTINE ORDER if tape data is used to determine positions, velocities, and attractions of perturbing bodies. Then read as file number of tape 3. See TFILE.
ICC(10)	A(153)	Any fixed-point integer	Input internal	Index of independent variable in COEFN array used in FUNCTION QUAD. For each set of coefficients there is an ICC. They are set at input time and are reset each time QUAD is called.
IMODE	A(1)	1,2,3,4,-1,-2,-3,-4 (fixed point)	STDATA: 1 input internal	Indicates integration mode. Must agree with input data (if input data is rectangular, IMODE should equal 2 or -2). Values indicate: 1 = orbit elements 2 = rectangular variables 3 = temporary rectangular 4 = Earth spherical change to rectangular -1 = orbit elements, change to rectangular -2 = rectangular, change to orbit elements -3 = orbit, change to temporary rectangular -4 = Earth spherical, change to orbit elements
LOOKX	A(8)	Fixed-point integer	Input	The program searches for the value C(LOOKX) = XLOOK if LOOKX $\neq$ 0. The search begins when C(LOOKSW) $\geq$ SWLOOK. If and when C(LOOK) = XLOOK output occurs and program control is directed by the parameter END.
LOOKSW	A(9)	Fixed-point integer	STDATA: 711 input	The search for C(LOOKX) = XLOOK does not begin until C(LOOKSW) $\geq$ SWLOOK. Typically, time is the deciding parameter; therefore, STDATA sets LOOKSW = 711.
MODOUT	A(20)	1,2,3,4 (fixed point)	STDATA: 4 input internal	MODOUT = 1 Output every nth step (n = STEPS) until TIME = TMIN, then shift to mode 2. = 2 Output at time intervals of DELMAX until TIME = TMAX. = 3 Output at time intervals of DELMAX until TIME = TMIN, then shift to mode 4. = 4 Output every nth step until TIME = TMAX.
NSAVE	C(4)	0,1,2,. . .,10 (fixed point)	Input	If the initial data (arrays A, XPRIM, and XPRIMB) for the nth stage is to be saved, it will occur just prior to the nth stage integration if NSAVE = NSTAGE, the stage index. NSAVE = 0 is ignored (no stage data will be saved).
OBLATN	A(40)	Any ALF coded body name	Input	If oblateness effects are to be considered, loading a body name will cause SUBROUTINE OBLATE to be called from SUBROUTINE EQUATE when OBLATN matches reference body.
RECALL	C(5)	Any value	Input	If RECALL $\neq$ 0.0, "starting" data will be restored from D array in MAIN. See NSAVE.
SIGNAL	B(31)	0.0 or 1.0	Internal	If and when C(LOOKX) = XLOOK, SIGNAL is set to 1.0 for use in any of the subroutines EXTRAS, EXTRA, STAGE, and so forth. EXTRA resets SIGNAL = 0.
STEPMX	A(16)	Any plus number	STDATA: 100.0 input	If (STEPGO + STEPNO) $\geq$ STEPMX, problem terminates.
STEPS	A(17)	Any plus number	STDATA: 1.0 input	Used when MODOUT = 1 or 4. Output will occur at every nth step where n = STEPS.
SWLOOK	A(10)	Any number	Input	Used when LOOKX $\neq$ 0. Value of the parameter C(LOOKSW) to be equaled or exceeded before the search for C(LOOK) = XLOOK begins.
TAPE 3	C(2)	0.0 or 3.0	Input internal	If "working" ephemeris tape is to be made, TAPE 3 must be set equal to zero through input contained in SUBROUTINE TAPE. If no tape is made, or after tape is made, TAPE 3 is set to 3.0.
TFILE	A(6)	Any plus integer	STDATA: 1.0 input	Selects which file of "working" ephemeris tape is to be used. ORDER positions tape in correct position by matching desired file number (TFILE) with code word (FILE) written at beginning of each file on tape.
TMAX	B(4)	Any number in seconds	Internal	When time = TMAX a stage is terminated.
TMIN	A(18)	Any number in seconds	Input	When time = TMIN output mode is changed. See MODOUT.
TRSFER	B(8)	0.0 or 1.0	Internal	Normally TRSFER = 0.0, but when origin is being translated, TRSFER = 1.0, which causes SUBROUTINES EPHMRS and ELIPSE to compute velocities as well as positions.
TTEST	A(54)	Any number in seconds	Internal	When integration mode is changed to temporary rectangular, TTEST is set as time at which program will begin checking for return to orbit elements. See NBODY part 7D.
XLOOK	A(12)	Any number	Input	The value of C(LOOKX) that is searched for as the trajectory integration proceeds providing LOOKX $\neq$ 0.

TABLE III. - BASIC OUTPUT FORMAT

(a) Sample output

```
STEP=   O. + O.     ECCENTRICITY= 1.00000000     OMEGA=-2.64801353
TIME= O.            SEMILATUS R.= 1.93844640E-09 TRU A= 3.14159262
JDAY= 2437640.8350  MEAN ANOMALY= O.              NODE= 2.02516600
ALFA= O.              PATH ANGLE= 89.9209976      INCL= 1.57079409
ALT.= 0.1875000     R PATH ANGLE= 89.9209976      DRAG= 4.99665982E-03
SUN    R= 1.4728028E 11 -0.261730 -0.885466 -0.383989

       V= 9.99999976E-02 R= 6373346.50     REFER=EARTH  RECTAN 2
      VX= 3.86224359E-02 X=-2463371.37     RMASS= 150000.000
      VY= 7.90702742E-02 Y= 5043168.50     REVS.= 0.32231534
      VZ= 4.74994606E-02 Z= 3019569.50      DELT= 6.00000000
      VR= 9.99999976E-02 G= 1.49946962      PUSH= O.
MOON   R= 3.9293912E 08 -0.387660 -0.874846 -0.290456
```

(b) Parameter identification

Output format mnemonic	Identification
STEP	Count of total number of successful integration steps to left of plus sign and count of failures on right
TIME	Time since beginning of integration process, t, sec
JDAY	Current Julian date
ECCENTRICITY	Osculating orbit eccentricity, e
SEMILATUS R.	Semilatus rectum of osculating orbit, p, m
MEAN ANOMALY	Mean anomaly of osculating orbit, M
OMEGA	Argument of pericenter, ω, radians
TRU A	True anomaly of osculating orbit, v, radians
NODE	Equatorial longitude of ascending node of osculating orbit, Ω, radians
INCL	Orbit inclination referred to mean equator and equinox of 1950.0, i, radians
ALFA	Angle between thrust and velocity, α, deg
PATH ANGLE	Angle between path and local horizontal, deg
V, VX, VY, VZ	Velocity and its x,y,z components, V, m/sec
R, X, Y, Z	Radius and its x,y,z components, r, m
REFER	Name of reference body, followed by integration mode, IMODE
RMASS	Vehicle mass, m, kg
REVS.	Revolutions past x-axis
DELT	Step size for current step, h, sec
ALT.	Altitude above Earth, m
R PATH ANGLE	Relative path angle, relative to Earth, deg
DRAG	Total drag force, D, newtons
VR	Velocity relative to rotating reference body
G	Total Earth g's acting on longitudinal axis of missile
PUSH	Thrust force, newtons
BNAME(i)R	Vehicle to perturbing body distance, r_i, plus direction cosines

TABLE IV. - COMMON ALLOCATION

THE CCMMON ARRAY C IS ARRANGED IN SUBARRAYS AS FOLLOWS.

C(11) - C(710) = A	(Those parameters whose initial values must be identical on different flights if the trajectories are to be identical)
C(711) - C(910) = XPRIM	(Most significant half of the double-precision integration variables)
C(911) - C(1110) = XPRIMB	(Least significant half of the double-precision integration variables)
C(1111) - C(1910) = B	(Those parameters whose initial values need not be identical on different flights if the trajectories are to be identical)
C(1911) - C(2110) = TABLE	(Table required by the input routine to locate input data)
C(2111) - C(3210) = D	(Array of initial values for A,XPRIM, and XPRIMB if restart facility is being used)

ALLOCATION FOR THE ARRAY A

```
  1  IMODE    NEQ      NSTAGE   ALT      END      TFILE    ASYMPT   LOOKX    LOOKSW   SWLOOK
 11  XTOL     XLOCK    EREF     ERLIMT   TKICK    STEPMX   STEPS    TMIN     DELMAX   MODOUT
 21  ATMN     RATM     DTOFFJ   TOFFT    RE       OBLATJ   OBLATD   OBLATH   AU       EIOL
 31  CONSU    CCNSTU   LAT      LONG     AZI      ELEV     VEL      LSTAGE   ROTATE   OBLATN
 41  STEPGO   STEPNO   DEL      SPD      TTOL     GASFAC   SQRDK1   REVS     ALPHA    BETA
 51  AK       -        -        TTEST    AW       -        -        -        U        IND
 61  -        -        TB       -        -        -        -        -        -        -
 71  -        -        RMASS1   -        -        -        -        -        -        -
 81  -        -        FLOW1    -        -        -        -        -        -        -
 91  -        -        SIMP1    -        -        -        -        -        -        -
101  -        -        AEXIT1   -        -        -        -        -        -        -
111  -        -        AREA1    -        -        -        -        -        -        -
121  -        -        IDENT    -        -        -        -        -        -        -
131  -        -        DELT1    -        -        -        -        -        -        -
141  -        -        BODYCD   -        -        -        -        -        -        -
151  -        -        ICC      -        -        -        -        -        -        -
161  -        -        CDI      CL       CD       PUSH     ELIPS    -        -        -
171  -        -        -        -        -        -        -        -        -        -
181  -        -        -        -        -        -        -        -        -        -
191  -        -        -        -        -        -        -        -        -        -
201  -        -        -        -        -        -        -        -        -        -
211  -        -        -        -        -        -        -        -        -        -
221  -        -        -        -        -        -        -        -        -        -
231  -        -        -        -        -        -        -        -        -        -
241  -        -        -        -        -        -        -        -        -        -
251  -        -        -        -        -        -        -        -        -        -
261  -        -        -        -        -        -        -        -        -        -
271  -        -        -        -        -        -        -        -        -        -
281  -        -        -        -        -        -        PNAME    -        -        -
291  -        -        -        -        -        -        -        -        -        -
301  -        -        -        -        -        -        -        -        -        -
311  -        -        -        -        -        -        REFER    -        -        -
321  -        -        -        -        -        -        -        -        -        -
331  -        -        -        -        -        -        -        -        -        -
341  -        -        -        -        -        -        AMASS    -        -        -
351  -        -        -        -        -        -        -        -        -        -
361  -        -        -        -        -        -        -        -        -        -
371  -        -        -        -        -        -        RCRIT    -        -        -
381  -        -        -        -        -        -        -        -        -        -
391  -        -        -        -        -        -        -        -        -        -
401  -        -        -        -        -        -        COEFN    -        -        -
411  -        -        -        -        -        -        -        -        -        -
421  -        -        -        -        -        -        -        -        -        -
431  -        -        -        -        -        -        -        -        -        -
441  -        -        -        -        -        -        -        -        -        -
451  -        -        -        -        -        -        -        -        -        -
461  -        -        -        -        -        -        -        -        -        -
471  -        -        -        -        -        -        -        -        -        -
481  -        -        -        -        -        -        -        -        -        -
491  -        -        -        -        -        -        -        -        -        -
501  -        -        -        -        -        -        -        -        -        -
511  -        -        -        -        -        -        -        -        -        -
521  -        -        -        -        -        -        -        -        -        -
531  -        -        -        -        -        -        -        -        -        -
541  -        -        -        -        -        -        -        -        -        -
551  -        -        -        -        -        -        -        -        -        -
561  -        -        -        -        -        -        -        -        -        -
571  -        -        -        -        -        -        -        -        -        -
581  -        -        -        -        -        -        -        -        -        -
591  -        -        -        -        -        -        -        -        INLOOK   NCASES
601  SWITCH
611
621
631
641
651
661
671
681
691
```

TABLE IV. - Continued. COMMON ALLOCATION

ALLOCATION FOR THE ARRAY B

1	DELT	SIMP	AEXIT	TMAX	FLOW	AREA	RESQRD	TRSFER	OLDDEL	A1
11	A2	ACOEF1	ACOEF2	ACOEF3	H2	SPACES	ERLOG	E2	KSUB	TABLT
21	REVOLV	FILE	RATMOS	NSTART	CHAMP	EPAR	EXMODE	EMONE	DNSITY	PSI
31	SIGNAL		PRESS	TM	SQRDK	GK2M	GKM	VMACH	DONE	TRU
41	NBODYS	MBODYS	ZN	QMAX	RSQRD	SINALF	SINBET	COSALF	COSBET	PMAGN
51	INDERR	SINTRU	COSTRU	SINCL	CINCL	SINV	COSV	RATIO	Q	PAR
61	-	-	COMPA	-	-	FORCE	-	-	DRAG	-
71	-	XIFT	-	-	OBLAT	-	-	QX	-	-
81	RADIAL	CIRCUM	ZORMAL	P	-	-	AMC	-	-	AM
91	AMSQRD	VX	VY	VZ	V	VSQRD	VATM	-	-	VQ
101	VQSQRD	R	-	-	-	-	-	-	-	XWHOLE
111	-	-	-	-	-	ORBELS	-	-	-	-
121	-	BNAME	-	-	-	-	-	-	-	EFMRS
131	-	-	-	-	-	-	BMASS	-	-	-
141	-	-	-	-	RBCRIT	-	-	-	-	-
151	-	-	BODYL	-	-	-	-	-	-	-
161	-	-	TIM	-	-	-	-	-	-	TDEL
171	-	-	-	-	-	-	IBODY	-	-	-
181	-	-	-	-	NEFMRS	-	-	-	-	-
191	-	-	RB	-	-	-	-	-	-	-
201	-	-	-	-	-	-	-	-	-	-
211	-	-	-	-	-	-	XP	-	-	-
221	-	-	-	-	-	-	-	-	-	-
231	-	-	-	-	-	-	-	-	-	-
241	VEFM	-	-	-	-	-	-	-	-	-
251	-	-	-	-	-	-	-	-	-	-
261	-	-	-	-	TDATA	-	-	-	-	-
271	-	-	-	-	-	-	-	-	-	-
281	-	-	-	-	-	-	-	-	-	-
291	-	-	-	-	-	-	-	-	-	-
301	-	-	-	-	-	-	-	-	-	-
311	-	-	-	-	-	-	-	-	-	-
321	-	-	-	-	-	-	-	-	-	-
331	-	-	-	-	-	-	-	-	-	-
341	-	-	-	-	-	-	-	-	-	-
351	-	-	-	-	-	-	-	-	-	-
361	-	-	-	-	-	-	-	-	-	-
371	-	-	-	-	-	-	-	-	-	-
381	-	-	-	-	-	-	-	-	-	-
391	PUSHO	EXITA	RAMC	-	-	-	-	PSIR	OUTPUT	RETURN
401	X	-	-	-	-	-	-	-	-	-
411	-	-	-	-	-	-	-	-	-	-
421	-	-	-	-	-	-	-	-	-	-
431	-	-	-	-	-	-	-	-	-	-
441	-	-	-	-	-	-	-	-	-	-
451	-	-	-	-	-	-	-	-	-	-
461	-	-	-	-	-	-	-	-	-	-
471	-	-	-	-	-	-	-	-	-	-
481	-	-	-	-	-	-	-	-	-	-
491	-	-	-	-	-	-	-	-	-	-
501	XDOT	-	-	-	-	-	-	-	-	-
511	-	-	-	-	-	-	-	-	-	-
521	-	-	-	-	-	-	-	-	-	-
531	-	-	-	-	-	-	-	-	-	-
541	-	-	-	-	-	-	-	-	-	-
551	-	-	-	-	-	-	-	-	-	-
561	-	-	-	-	-	-	-	-	-	-
571	-	-	-	-	-	-	-	-	-	-
581	-	-	-	-	-	-	-	-	-	-
591	-	-	-	-	-	-	-	-	-	-
601	XINC	-	-	-	-	-	-	-	-	-
611	-	-	-	-	-	-	-	-	-	-
621	-	-	-	-	-	-	-	-	-	-
631	-	-	-	-	-	-	-	-	-	-
641	-	-	-	-	-	-	-	-	-	-
651	-	-	-	-	-	-	-	-	-	-
661	-	-	-	-	-	-	-	-	-	-
671	-	-	-	-	-	-	-	-	-	-
681	-	-	-	-	-	-	-	-	-	-
691	-	-	-	-	-	-	-	-	-	-
701										
711										
721										
731										
741										
751										
761										
771										
781										
791										

TABLE IV. - Continued. COMMON ALLOCATION

```
ALLCCATICN FOR THE ARRAY C

  1  NCASE     TAPE3     CLEAR     NSAVE     RECALL
 11  IMCCE     NEQ       NSTAGE    ALT       END       TFILE     ASYMPT    LOOKX     LOOKSW    SWLOOK
 21  XTOL      XLOCK     EREF      ERLIMT    TKICK     STEPMX    STEPS     IMIN      DELMAX    MODOUT
 31  ATMN      RATM      DTOFFJ    TOFFT     RE        OBLAIJ    OBLATD    OBLATH    AU        ETOL
 41  CONSU     CCNSTU    LAT       LONG      AZI       ELEV      VEL       LSTAGE    ROTATE    OBLAIN
 51  STEPGO    STEPNO    DEL       SPD       TTOL      GASFAC    SQRDK1    REVS      ALPHA     BETA
 61  AK        -         -         TTEST     AW        -         -         -         U         IND
 71  -         -         TB        -         -         -         -         -         -         -
 81  -         -         RMASS1    -         -         -         -         -         -         -
 91  -         -         FLOW1     -         -         -         -         -         -         -
101  -         -         SIMP1     -         -         -         -         -         -         -
111  -         -         AEXIT1    -         -         -         -         -         -         -
121  -         -         AREA1     -         -         -         -         -         -         -
131  -         -         IDENT     -         -         -         -         -         -         -
141  -         -         DELT1     -         -         -         -         -         -         -
151  -         -         BODYCD    -         -         -         -         -         -         -
161  -         -         ICC       -         -         -         -         -         -         -
171  -         -         CDI       CL        CD        PUSH      ELIPS     -         -         -
181  -         -         -         -         -         -         -         -         -         -
191  -         -         -         -         -         -         -         -         -         -
201  -         -         -         -         -         -         -         -         -         -
211  -         -         -         -         -         -         -         -         -         -
221  -         -         -         -         -         -         -         -         -         -
231  -         -         -         -         -         -         -         -         -         -
241  -         -         -         -         -         -         -         -         -         -
251  -         -         -         -         -         -         -         -         -         -
261  -         -         -         -         -         -         -         -         -         -
271  -         -         -         -         -         -         -         -         -         -
281  -         -         -         -         -         -         -         -         -         -
291  -         -         -         -         -         -         PNAME     -         -         -
301  -         -         -         -         -         -         -         -         -         -
311  -         -         -         -         -         -         -         -         -         -
321  -         -         -         -         -         -         REFER     -         -         -
331  -         -         -         -         -         -         -         -         -         -
341  -         -         -         -         -         -         -         -         -         -
351  -         -         -         -         -         -         AMASS     -         -         -
361  -         -         -         -         -         -         -         -         -         -
371  -         -         -         -         -         -         -         -         -         -
381  -         -         -         -         -         -         RCRIT     -         -         -
391  -         -         -         -         -         -         -         -         -         -
401  -         -         -         -         -         -         -         -         -         -
411  -         -         -         -         -         -         COEFN     -         -         -
421  -         -         -         -         -         -         -         -         -         -
431  -         -         -         -         -         -         -         -         -         -
441  -         -         -         -         -         -         -         -         -         -
451  -         -         -         -         -         -         -         -         -         -
461  -         -         -         -         -         -         -         -         -         -
471  -         -         -         -         -         -         -         -         -         -
481  -         -         -         -         -         -         -         -         -         -
491  -         -         -         -         -         -         -         -         -         -
501  -         -         -         -         -         -         -         -         -         -
511  -         -         -         -         -         -         -         -         -         -
521  -         -         -         -         -         -         -         -         -         -
531  -         -         -         -         -         -         -         -         -         -
541  -         -         -         -         -         -         -         -         -         -
551  -         -         -         -         -         -         -         -         -         -
561  -         -         -         -         -         -         -         -         -         -
571  -         -         -         -         -         -         -         -         -         -
581  -         -         -         -         -         -         -         -         -         -
591  -         -         -         -         -         -         -         -         -         -
601  -         -         -         -         -         -         -         -         INLOOK    NCASES
611  SWITCH
621
631
641
651
661
671
681
691
701
711  XPRIM     -         -         -         -         -         -         -         -         -
```

TABLE IV. - Continued. COMMON ALLOCATION

721	-	-	-	-	-	-	-	-	-	-
731	-	-	-	-	-	-	-	-	-	-
741	-	-	-	-	-	-	-	-	-	-
751	-	-	-	-	-	-	-	-	-	-
761	-	-	-	-	-	-	-	-	-	-
771	-	-	-	-	-	-	-	-	-	-
781	-	-	-	-	-	-	-	-	-	-
791	-	-	-	-	-	-	-	-	-	-
801	-	-	-	-	-	-	-	-	-	-
811	-	-	-	-	-	-	-	-	-	-
821	-	-	-	-	-	-	-	-	-	-
831	-	-	-	-	-	-	-	-	-	-
841	-	-	-	-	-	-	-	-	-	-
851	-	-	-	-	-	-	-	-	-	-
861	-	-	-	-	-	-	-	-	-	-
871	-	-	-	-	-	-	-	-	-	-
881	-	-	-	-	-	-	-	-	-	-
891	-	-	-	-	-	-	-	-	-	-
901	-	-	-	-	-	-	-	-	-	-
911	XPRIMB	-	-	-	-	-	-	-	-	-
921	-	-	-	-	-	-	-	-	-	-
931	-	-	-	-	-	-	-	-	-	-
941	-	-	-	-	-	-	-	-	-	-
951	-	-	-	-	-	-	-	-	-	-
961	-	-	-	-	-	-	-	-	-	-
971	-	-	-	-	-	-	-	-	-	-
981	-	-	-	-	-	-	-	-	-	-
991	-	-	-	-	-	-	-	-	-	-
1001	-	-	-	-	-	-	-	-	-	-
1011	-	-	-	-	-	-	-	-	-	-
1021	-	-	-	-	-	-	-	-	-	-
1031	-	-	-	-	-	-	-	-	-	-
1041	-	-	-	-	-	-	-	-	-	-
1051	-	-	-	-	-	-	-	-	-	-
1061	-	-	-	-	-	-	-	-	-	-
1071	-	-	-	-	-	-	-	-	-	-
1081	-	-	-	-	-	-	-	-	-	-
1091	-	-	-	-	-	-	-	-	-	-
1101	-	-	-	-	-	-	-	-	-	-
1111	DELT	SIMP	AEXIT	TMAX	FLOW	AREA	RESQRD	TRSFER	OLDDEL	A1
1121	A2	ACOEF1	ACOEF2	ACOEF3	H2	SPACES	ERLOG	E2	KSUB	TABLT
1131	REVOLV	FILE	RATMOS	NSTART	CHAMP	EPAR	EXMODE	EMONE	DNSITY	PSI
1141	SIGNAL		PRESS	TM	SQRDK	GK2M	GKM	VMACH	DONE	TRU
1151	NBODYS	MBODYS	ZN	QMAX	RSQRD	SINALF	SINBET	COSALF	COSBET	PMAGN
1161	INDERR	SINTRU	COSTRU	SINCL	CINCL	SINV	COSV	RATIO	Q	PAR
1171	-	-	COMPA	-	-	FORCE	-	-	DRAG	-
1181	-	XIFT	-	-	OBLAT	-	-	QX	-	-
1191	RADIAL	CIRCUM	ZORMAL	P	-	-	AMC	-	-	AM
1201	AMSQRD	VX	VY	VZ	V	VSQRD	VATM	-	-	VQ
1211	VQSQRD	R	-	-	-	-	-	-	-	XWHOLE
1221	-	-	-	-	-	ORBELS	-	-	-	-
1231	-	BNAME	-	-	-	-	-	-	-	EFMRS
1241	-	-	-	-	-	-	BMASS	-	-	-
1251	-	-	-	-	RBCRIT	-	-	-	-	-
1261	-	-	BODYL	-	-	-	-	-	-	-
1271	-	-	TIM	-	-	-	-	-	-	TDEL
1281	-	-	-	-	-	-	IBODY	-	-	-
1291	-	-	-	-	NEFMRS	-	-	-	-	-
1301	-	-	RB	-	-	-	-	-	-	-
1311	-	-	-	-	-	-	-	-	-	-

TABLE IV. - Continued. COMMON ALLOCATION

1321	-	-	-	-	-	-	XP	-	-	-
1331	-	-	-	-	-	-	-	-	-	-
1341	-	-	-	-	-	-	-	-	-	-
1351	VEFM	-	-	-	-	-	-	-	-	-
1361	-	-	-	-	-	-	-	-	-	-
1371	-	-	-	-	TDATA	-	-	-	-	-
1381	-	-	-	-	-	-	-	-	-	-
1391	-	-	-	-	-	-	-	-	-	-
1401	-	-	-	-	-	-	-	-	-	-
1411	-	-	-	-	-	-	-	-	-	-
1421	-	-	-	-	-	-	-	-	-	-
1431	-	-	-	-	-	-	-	-	-	-
1441	-	-	-	-	-	-	-	-	-	-
1451	-	-	-	-	-	-	-	-	-	-
1461	-	-	-	-	-	-	-	-	-	-
1471	-	-	-	-	-	-	-	-	-	-
1481	-	-	-	-	-	-	-	-	-	-
1491	-	-	-	-	-	-	-	-	-	-
1501	PUSHO	EXITA	RAMC	-	-	-	-	PSTR	OUTPOT	RETURN
1511	X	-	-	-	-	-	-	-	-	-
1521	-	-	-	-	-	-	-	-	-	-
1531	-	-	-	-	-	-	-	-	-	-
1541	-	-	-	-	-	-	-	-	-	-
1551	-	-	-	-	-	-	-	-	-	-
1561	-	-	-	-	-	-	-	-	-	-
1571	-	-	-	-	-	-	-	-	-	-
1581	-	-	-	-	-	-	-	-	-	-
1591	-	-	-	-	-	-	-	-	-	-
1601	-	-	-	-	-	-	-	-	-	-
1611	XDOT	-	-	-	-	-	-	-	-	-
1621	-	-	-	-	-	-	-	-	-	-
1631	-	-	-	-	-	-	-	-	-	-
1641	-	-	-	-	-	-	-	-	-	-
1651	-	-	-	-	-	-	-	-	-	-
1661	-	-	-	-	-	-	-	-	-	-
1671	-	-	-	-	-	-	-	-	-	-
1681	-	-	-	-	-	-	-	-	-	-
1691	-	-	-	-	-	-	-	-	-	-
1701	-	-	-	-	-	-	-	-	-	-
1711	XINC	-	-	-	-	-	-	-	-	-
1721	-	-	-	-	-	-	-	-	-	-
1731	-	-	-	-	-	-	-	-	-	-
1741	-	-	-	-	-	-	-	-	-	-
1751	-	-	-	-	-	-	-	-	-	-
1761	-	-	-	-	-	-	-	-	-	-
1771	-	-	-	-	-	-	-	-	-	-
1781	-	-	-	-	-	-	-	-	-	-
1791	-	-	-	-	-	-	-	-	-	-
1801	-	-	-	-	-	-	-	-	-	-
1811										
1821										
1831										
1841										
1851										
1861										
1871										
1881										
1891										
1901										
1911	TABLE	-	-	-	-	-	-	-	-	-

TABLE IV. - Continued. COMMON ALLOCATION

1921	-	-	-	-	-	-	-	-	-	-
1931	-	-	-	-	-	-	-	-	-	-
1941	-	-	-	-	-	-	-	-	-	-
1951	-	-	-	-	-	-	-	-	-	-
1961	-	-	-	-	-	-	-	-	-	-
1971	-	-	-	-	-	-	-	-	-	-
1981	-	-	-	-	-	-	-	-	-	-
1991	-	-	-	-	-	-	-	-	-	-
2001	-	-	-	-	-	-	-	-	-	-
2011	-	-	-	-	-	-	-	-	-	-
2021	-	-	-	-	-	-	-	-	-	-
2031	-	-	-	-	-	-	-	-	-	-
2041	-	-	-	-	-	-	-	-	-	-
2051	-	-	-	-	-	-	-	-	-	-
2061	-	-	-	-	-	-	-	-	-	-
2071	-	-	-	-	-	-	-	-	-	-
2081	-	-	-	-	-	-	-	-	-	-
2091	-	-	-	-	-	-	-	-	-	-
2101	-	-	-	-	-	-	-	-	-	-
2111	D	-	-	-	-	-	-	-	-	-
2121	-	-	-	-	-	-	-	-	-	-
2131	-	-	-	-	-	-	-	-	-	-
2141	-	-	-	-	-	-	-	-	-	-
2151	-	-	-	-	-	-	-	-	-	-
2161	-	-	-	-	-	-	-	-	-	-
2171	-	-	-	-	-	-	-	-	-	-
2181	-	-	-	-	-	-	-	-	-	-
2191	-	-	-	-	-	-	-	-	-	-
2201	-	-	-	-	-	-	-	-	-	-
2211	-	-	-	-	-	-	-	-	-	-
2221	-	-	-	-	-	-	-	-	-	-
2231	-	-	-	-	-	-	-	-	-	-
2241	-	-	-	-	-	-	-	-	-	-
2251	-	-	-	-	-	-	-	-	-	-
2261	-	-	-	-	-	-	-	-	-	-
2271	-	-	-	-	-	-	-	-	-	-
2281	-	-	-	-	-	-	-	-	-	-
2291	-	-	-	-	-	-	-	-	-	-
2301	-	-	-	-	-	-	-	-	-	-
2311	-	-	-	-	-	-	-	-	-	-
2321	-	-	-	-	-	-	-	-	-	-
2331	-	-	-	-	-	-	-	-	-	-
2341	-	-	-	-	-	-	-	-	-	-
2351	-	-	-	-	-	-	-	-	-	-
2361	-	-	-	-	-	-	-	-	-	-
2371	-	-	-	-	-	-	-	-	-	-
2381	-	-	-	-	-	-	-	-	-	-
2391	-	-	-	-	-	-	-	-	-	-
2401	-	-	-	-	-	-	-	-	-	-
2411	-	-	-	-	-	-	-	-	-	-
2421	-	-	-	-	-	-	-	-	-	-
2431	-	-	-	-	-	-	-	-	-	-
2441	-	-	-	-	-	-	-	-	-	-
2451	-	-	-	-	-	-	-	-	-	-
2461	-	-	-	-	-	-	-	-	-	-
2471	-	-	-	-	-	-	-	-	-	-
2481	-	-	-	-	-	-	-	-	-	-
2491	-	-	-	-	-	-	-	-	-	-
2501	-	-	-	-	-	-	-	-	-	-
2511	-	-	-	-	-	-	-	-	-	-

TABLE IV. - Concluded. COMMON ALLOCATION

2521	-	-	-	-	-	-	-	-	-	-
2531	-	-	-	-	-	-	-	-	-	-
2541	-	-	-	-	-	-	-	-	-	-
2551	-	-	-	-	-	-	-	-	-	-
2561	-	-	-	-	-	-	-	-	-	-
2571	-	-	-	-	-	-	-	-	-	-
2581	-	-	-	-	-	-	-	-	-	-
2591	-	-	-	-	-	-	-	-	-	-
2601	-	-	-	-	-	-	-	-	-	-
2611	-	-	-	-	-	-	-	-	-	-
2621	-	-	-	-	-	-	-	-	-	-
2631	-	-	-	-	-	-	-	-	-	-
2641	-	-	-	-	-	-	-	-	-	-
2651	-	-	-	-	-	-	-	-	-	-
2661	-	-	-	-	-	-	-	-	-	-
2671	-	-	-	-	-	-	-	-	-	-
2681	-	-	-	-	-	-	-	-	-	-
2691	-	-	-	-	-	-	-	-	-	-
2701	-	-	-	-	-	-	-	-	-	-
2711	-	-	-	-	-	-	-	-	-	-
2721	-	-	-	-	-	-	-	-	-	-
2731	-	-	-	-	-	-	-	-	-	-
2741	-	-	-	-	-	-	-	-	-	-
2751	-	-	-	-	-	-	-	-	-	-
2761	-	-	-	-	-	-	-	-	-	-
2771	-	-	-	-	-	-	-	-	-	-
2781	-	-	-	-	-	-	-	-	-	-
2791	-	-	-	-	-	-	-	-	-	-
2801	-	-	-	-	-	-	-	-	-	-
2811	-	-	-	-	-	-	-	-	-	-
2821	-	-	-	-	-	-	-	-	-	-
2831	-	-	-	-	-	-	-	-	-	-
2841	-	-	-	-	-	-	-	-	-	-
2851	-	-	-	-	-	-	-	-	-	-
2861	-	-	-	-	-	-	-	-	-	-
2871	-	-	-	-	-	-	-	-	-	-
2881	-	-	-	-	-	-	-	-	-	-
2891	-	-	-	-	-	-	-	-	-	-
2901	-	-	-	-	-	-	-	-	-	-
2911	-	-	-	-	-	-	-	-	-	-
2921	-	-	-	-	-	-	-	-	-	-
2931	-	-	-	-	-	-	-	-	-	-
2941	-	-	-	-	-	-	-	-	-	-
2951	-	-	-	-	-	-	-	-	-	-
2961	-	-	-	-	-	-	-	-	-	-
2971	-	-	-	-	-	-	-	-	-	-
2981	-	-	-	-	-	-	-	-	-	-
2991	-	-	-	-	-	-	-	-	-	-
3001	-	-	-	-	-	-	-	-	-	-
3011	-	-	-	-	-	-	-	-	-	-
3021	-	-	-	-	-	-	-	-	-	-
3031	-	-	-	-	-	-	-	-	-	-
3041	-	-	-	-	-	-	-	-	-	-
3051	-	-	-	-	-	-	-	-	-	-
3061	-	-	-	-	-	-	-	-	-	-
3071	-	-	-	-	-	-	-	-	-	-
3081	-	-	-	-	-	-	-	-	-	-
3091	-	-	-	-	-	-	-	-	-	-
3101	-	-	-	-	-	-	-	-	-	-
3111	-	-	-	-	-	-	-	-	-	-
3121	-	-	-	-	-	-	-	-	-	-
3131	-	-	-	-	-	-	-	-	-	-
3141	-	-	-	-	-	-	-	-	-	-
3151	-	-	-	-	-	-	-	-	-	-
3161	-	-	-	-	-	-	-	-	-	-
3171	-	-	-	-	-	-	-	-	-	-
3181	-	-	-	-	-	-	-	-	-	-
3191	-	-	-	-	-	-	-	-	-	-
3201	-	-	-	-	-	-	-	-	-	-

TABLE V. - ELEMENTS OF INTEGRATION VARIABLE ARRAY XPRIM

[XPRIM 9 to 100 are left for expansion.]

Integration variables	XPRIM							
	1	2	3	4	5	6	7	8
Rectangular variables	Time	Mass	x-Component of velocity	y-Component of velocity	z-Component of velocity	x-Component of position	y-Component of position	z-Component of position
Orbit elements	Time	Mass	Eccentricity	Argument of pericenter	Longitude of ascending nodes	Orbit inclination	Mean anomaly	Semilatus rectum

TABLE VI. - ASSUMED VALUES OF ASTRONOMICAL CONSTANTS

Constant	Assumed value	FORTRAN name	COMMON location
Astronomical unit, m	1.49599×10^{11}	AU	A(29)
Gravitational constant of the Sun, AU^3/day^2	2.959122083×10^{-4}	SQRDK1	A(47)
Equatorial Earth radius, m	6378165.	RE	A(25)
Earth oblateness coefficient, J	1.62345×10^{-3}	OBLATJ	A(26)
Earth oblateness coefficient, $\mathscr{D}$	7.875×10^{-6}	OBLATD	A(27)
Earth oblateness coefficient, H	-5.75×10^{-6}	OBLATH	A(28)
Earth radii per AU	4.26546512×10^{-5}	ERTOAU	[a]C(3)
Day, sec	86400	SPD	A(44)
Mass, reciprocal sun mass units:			
Sun	1.0	AMASS(1)	A(347)
Mercury	6,120,000	AMASS(2)	A(348)
Venus	408,645	AMASS(3)	A(349)
Earth	332951.3	AMASS(4)	A(350)
Mars	3,088,000	AMASS(5)	A(351)
Jupiter	1047.39	AMASS(6)	A(352)
Saturn	3500.0	AMASS(7)	A(353)
Uranus	22,869	AMASS(8)	A(354)
Neptune	18,889	AMASS(9)	A(355)
Pluto	400,000	AMASS(10)	A(356)
Moon	AMASS(4) × 81.375	AMASS(11)	A(357)
Earth-Moon	AMASS(4) + AMASS(11)	AMASS(12)	A(358)
Sphere-of-influence radii, m:			
Sun	1.0×10^{20}	RCRIT(1)	A(377)
Mercury	1.0×10^{8}	RCRIT(2)	A(378)
Venus	6.14×10^{8}	RCRIT(3)	A(379)
Earth	9.25×10^{8}	RCRIT(4)	A(380)
Mars	5.78×10^{8}	RCRIT(5)	A(381)
Jupiter	4.81×10^{10}	RCRIT(6)	A(382)
Saturn	5.46×10^{10}	RCRIT(7)	A(383)
Uranus	5.17×10^{10}	RCRIT(8)	A(384)
Neptune	8.61×10^{10}	RCRIT(9)	A(385)
Pluto	3.81×10^{10}	RCRIT(10)	A(386)
Moon	1.60×10^{8}	RCRIT(11)	A(387)

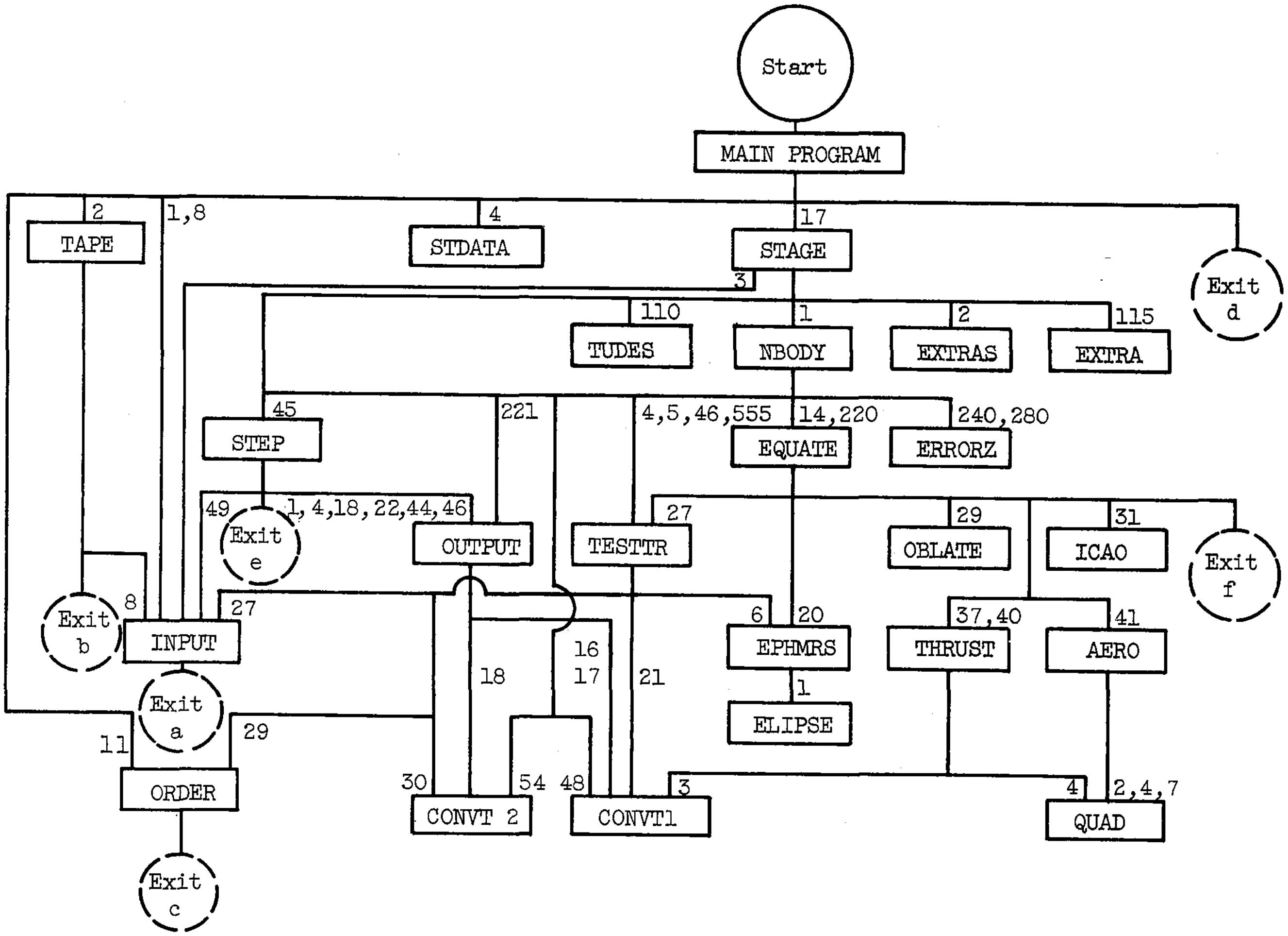
[a] Location relative to COMMON of SUBROUTINE TAPE (TAPE has a COMMON that is independent of all other subroutines).

TABLE VII. - LEWIS RESEARCH CENTER EPHEMERIS TAPE DATA

[The beginning date of all bodies except Mars is 2437 200.5 or Oct. 23, 1960. The beginning date for Mars is 2437 202.5 or Oct. 25, 1960.]

Body	Source	End date		Number of fits	Average, days/fit	Average, deg/fit	Source checked against	Average error	Maximum error
		Gregorian	Julian						
Venus	Themis	Oct. 31, 2000	2451 848.5	968	15	24	JPL	1.7	7.3
Earth-Moon barycenter	↓	Oct. 31, 2000	2451 848.5	962	15	15	JPL	1.8	9.5
Sun	↓	Nov. 24, 2000	2451 872.5	1821	8	8	JPL Themis	5.0 .06	21.0 3.0
Moon	JPL	Nov. 26, 1970	2440 916.5	1851	2	26	JPL	.14	9.5
Mars	JPL	July 26, 1998	2451 020.5	315	44	23	↓	1.1	7.2
Jupiter	Themis	March 2, 2060	2473 520.5	110	330	27	↓	1.6	9.5
Saturn	↓	↓	↓	44	825	27	↓	1.5	8.6
Uranus	↓	↓	↓	30	1211	14	↓	.95	6.5
Neptune	↓	↓	↓	31	1172	7	↓	.52	3.2
Pluto	↓	↓	↓	33	1101	4	Themis	.41	3.2

The error in the x-component of position, with similar equations for the y- and z-components, is given by $e_x = \left[(x' - x)/R\right]10^8$ where x' = merged ephemeris position component; x = check source position component; $R^2 = x^2 + y^2 + z^2$.



Exit	Cause of exit
a	End of data on input tape or incorrectly punched data cards
b	Illegal request of ephemeris data (misspelled names, wrong dates, etc.)
c	Illegal list of bodies (misspelled, unconnected references, etc.)
d	Sense switch 6 down (machine operator termination)
e	Number of permissible integration steps (STEPMX) exceeded
f	Nonconvergence of Kepler's equation after 15 iterations

Figure 1. - Block diagram of principal subprograms and program exits. Numbers on the block diagram are the FORTRAN calling statement numbers. A program may call only those programs located at a lower level and connected by a line. Logic decisions are not shown.

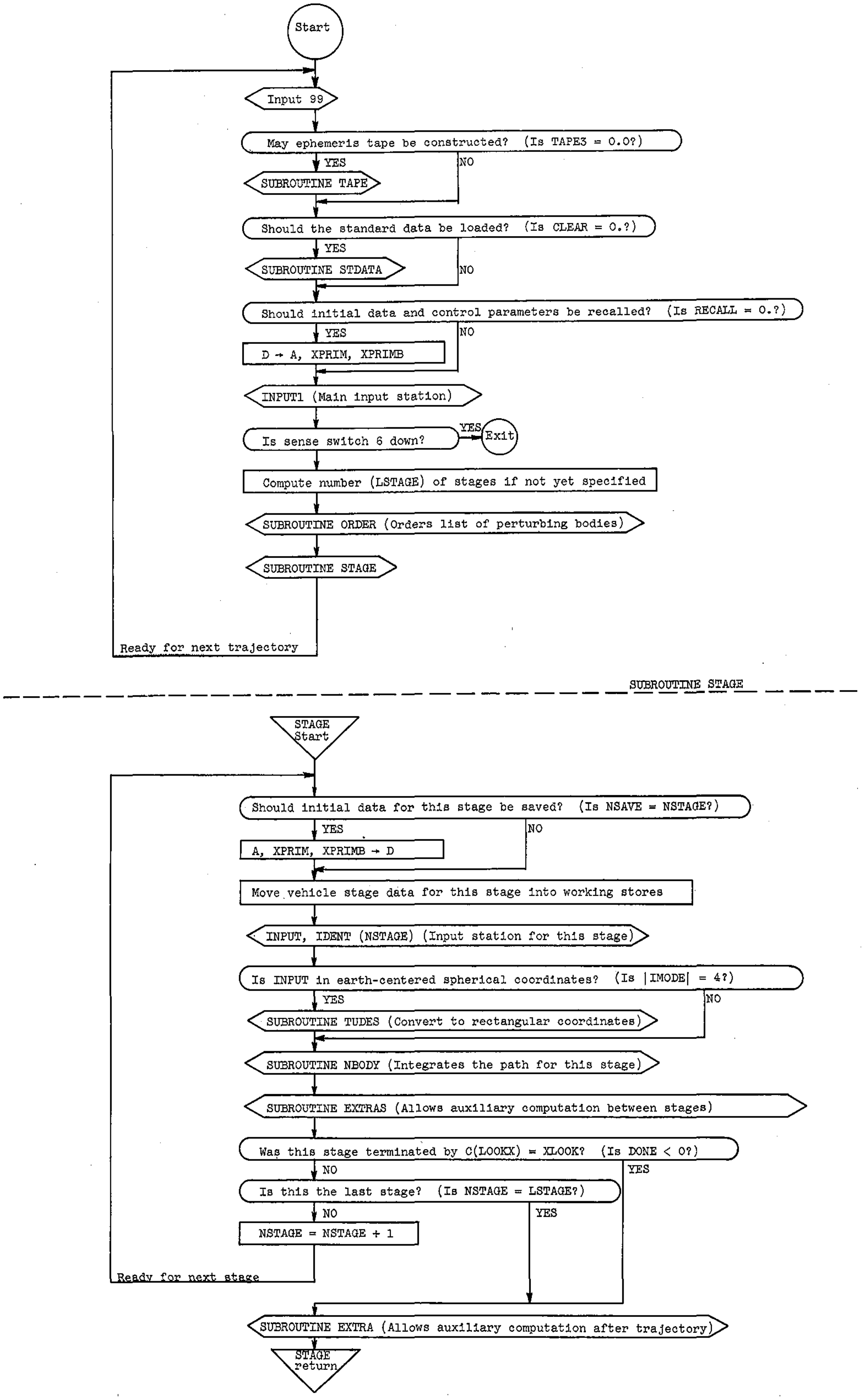


Figure 2. - Flow diagram of the main program and SUBROUTINE STAGE.

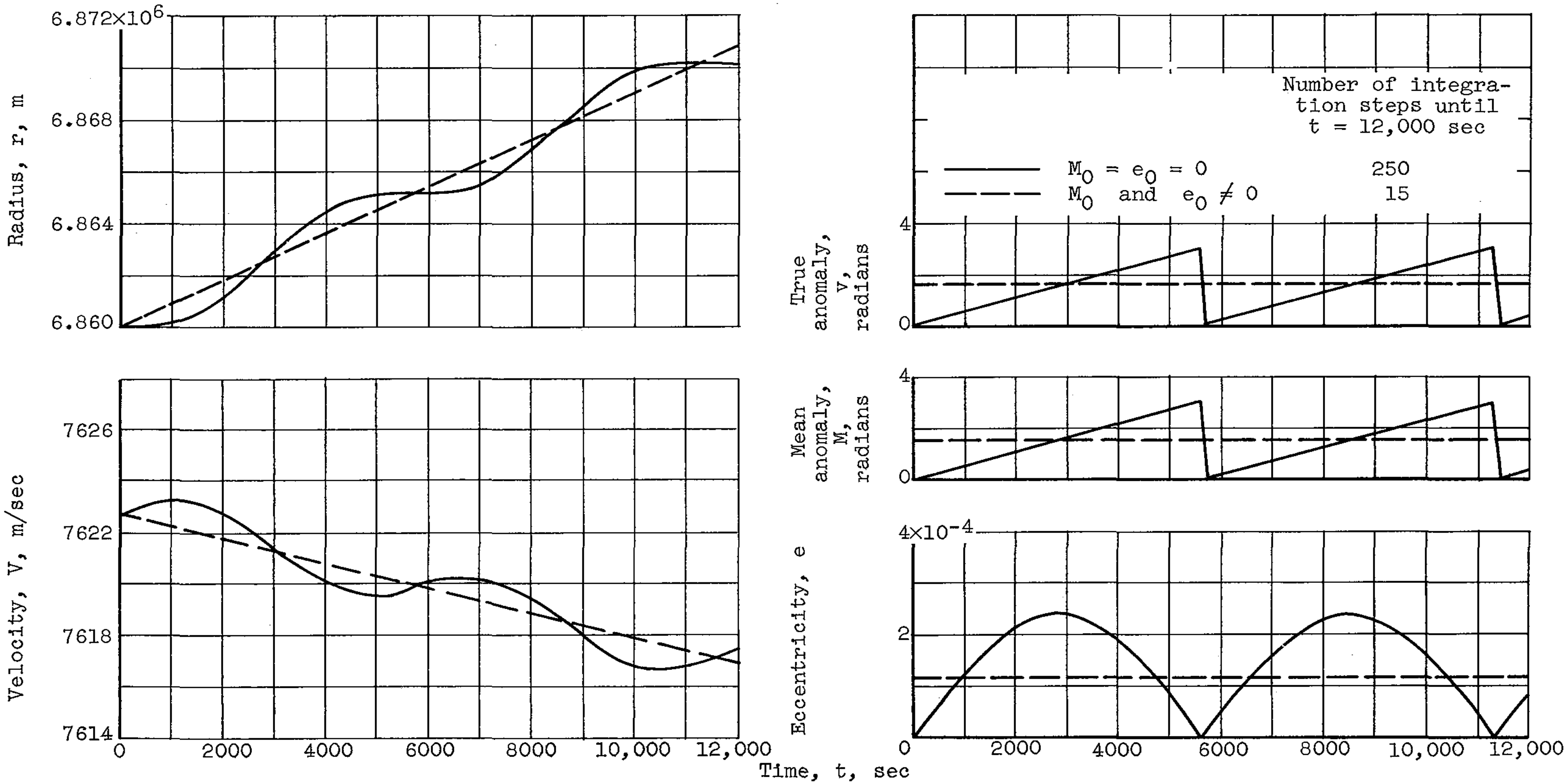


Figure 3. - Time histories of several trajectory parameters for Example I.

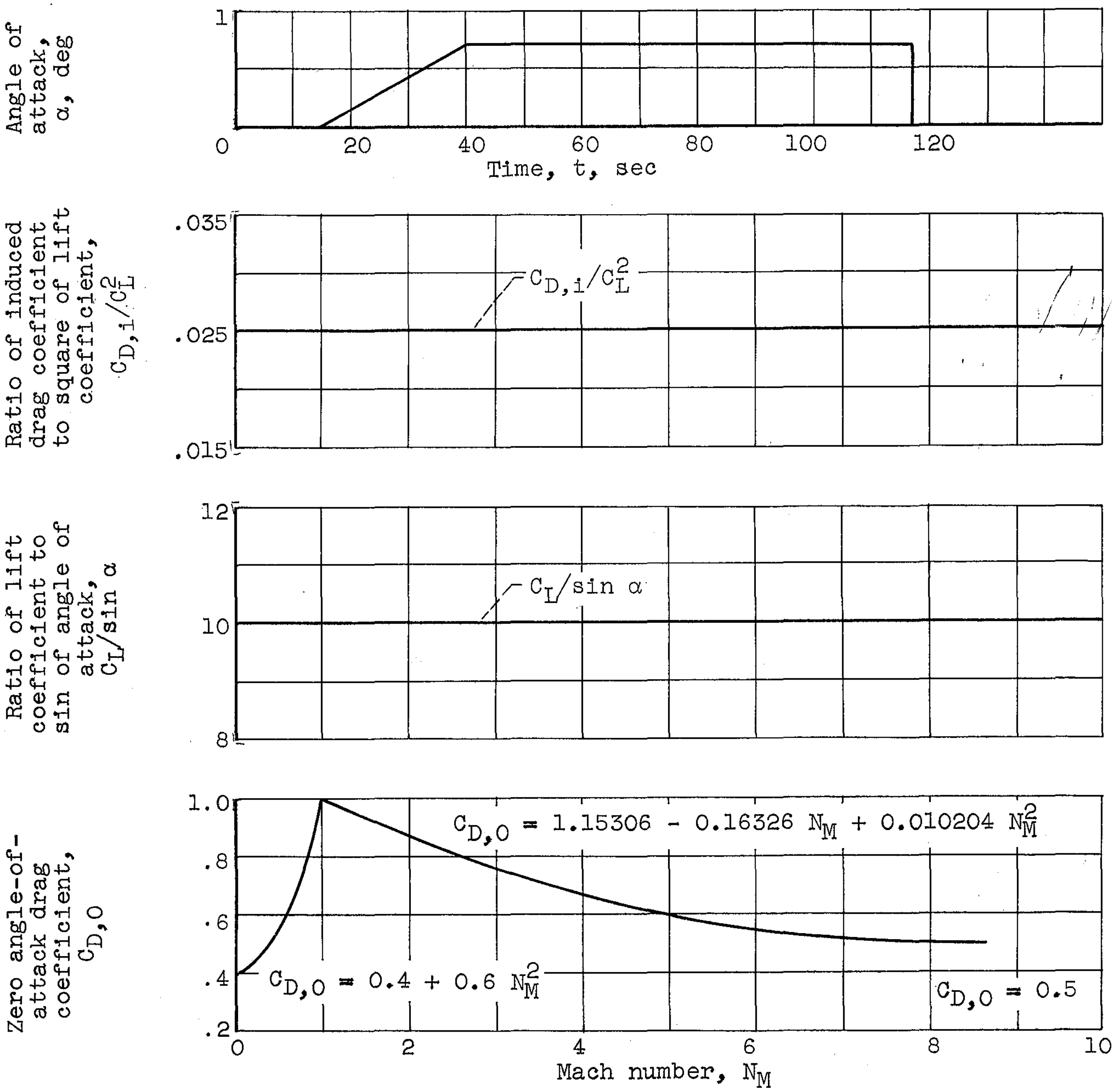


Figure 4. - Angle-of-attack schedule and variation of drag and lift coefficients with Mach number.

NASA-Langley, 1963