EXPERIMENTAL PERFORMANCE EVALUATION
OF A RADIAL-INFLOW TURBINE OVER A
RANGE OF SPECIFIC SPEEDS

by Milton G. Kofskey and Charles A. Wasserbauer

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1966
EXPERIMENTAL PERFORMANCE EVALUATION OF A RADIAL-INFLOW TURBINE OVER A RANGE OF SPECIFIC SPEEDS

By Milton G. Kofskey and Charles A. Wasserbauer

Lewis Research Center
Cleveland, Ohio
EXPERIMENTAL PERFORMANCE EVALUATION OF A RADIAL-INFLOW TURBINE OVER A RANGE OF SPECIFIC SPEEDS
by Milton G. Kofskey and Charles A. Wasserbauer
Lewis Research Center

SUMMARY

An experimental investigation was made to determine the effect of specific speed on efficiency for a 4.59-inch radial-inflow turbine. The range of specific speeds investigated (72 to 108) at equivalent design speed and pressure ratio was obtained by changing volume flow, based on rotor exit conditions. Changes in volume flow were accomplished by the use of stators having throat areas nominally 50, 75, 100, and 125 percent of design. The turbine was operated with air as the working fluid.

Maximum total and static efficiencies were obtained over the specific speed range of about 80 to 90. The peak total and static efficiencies were 0.91 and 0.87, respectively, for the 75-percent configuration.

An understanding of the losses which contributed to the variation of turbine performance with specific speed at design blade-jet speed ratio was made possible by an analysis which determined the magnitude of the various losses for each configuration. Stator loss was the predominant contributor to the decrease in efficiency as specific speed was reduced from a value of 86. Rotor incidence and viscous losses were the primary contributors to the decrease in performance when specific speed was increased above the value of 90. Stator exit static-pressure measurements showed that, at equivalent design speed and pressure ratio, rotor reaction increased as specific speed increased.

Rotor exit total-pressure and flow angle surveys indicated that low losses were obtained near the hub region of the rotor for all configurations at equivalent design speed and pressure ratio. Comparatively high losses were obtained near the outer wall. These increased losses may have resulted from tip leakage effects with blade unloading, as well as from centrifugation of low-momentum fluid to this region.
INTRODUCTION

The current Brayton-cycle space-power technology program at the Lewis Research Center includes the experimental investigation of factors which influence the performance of small radial-inflow turbines. One such factor is the specific speed parameter, which relates the operating variables of turbine rotative speed, volume flow based on exit conditions, and ideal specific work to turbine geometry and aerodynamic performance.

Reference 1 shows specific speed - efficiency correlations for a number of radial-inflow turbines of various sizes and for a wide range of inlet conditions. This reference shows that high efficiency is attainable for a specific speed range from 65 to 105, with a significant reduction in efficiency outside this range. However, turbine size, rotor tip clearance, and Reynolds number effects are present in the specific speed - efficiency correlations but are not examined separately. Therefore, the experimental investigation described herein was conducted to determine the specific speed effect on performance for a particular turbine size with rotor tip clearance and Reynolds number held constant.

Two approaches were considered to achieve the range of specific speeds. One was to design and fabricate an optimized stator and rotor configuration for each specific speed point, and the other was to use several stators with one rotor configuration. The second approach was chosen because it would minimize time and cost of the program; however, less than optimum turbine configurations may have resulted, especially at the extremes of the specific speed range.

The 4.59-inch-tip-diameter radial-inflow turbine of reference 2 was chosen as the research turbine. The design specific speed for this turbine is 95.6. Three additional stators having throat areas nominally 50, 75, and 125 percent of design were fabricated. The four configurations cover a specific speed range of 68 to 107 at equivalent design speed and pressure ratio. Each configuration was investigated over a range of turbine pressure ratios at equivalent design speed.

This report presents the performance of the subject turbine for each configuration and shows the specific speed effect on turbine efficiency. Results are presented in terms of equivalent weight flow and efficiency at equivalent design speed over a range of pressure ratios. Internal flow characteristics are presented in terms of static pressure variation through the turbine and radial variation of exit flow angle and loss distribution at the rotor exit.

SYMBOLS

\[ g \quad \text{gravitational constant, 32.174 ft/sec}^2 \]
\[ H' \quad \text{isentropic specific work based on total-pressure ratio, ft-lb/lb} \]
\( \Delta h \) specific work, Btu/lb

\( J \) mechanical equivalent of heat, 778.029 ft-lb/Btu

\( N \) turbine speed, rpm

\( N_s \) specific speed, \( NQ^{1/2}/(H'^{3/4}) \), ft\(^{3/4}\)/sec\(^{1/2}\)

\( p \) pressure, psia

\( Q \) volume flow (based on exit conditions), cu ft/sec

\( \text{Re} \) Reynolds number, w/\( \mu r_t \)

\( r \) radius, ft

\( U \) blade velocity, ft/sec

\( V \) absolute gas velocity, ft/sec

\( V_j \) ideal jet-speed corresponding to total- to static-pressure ratio across turbine, \( \left(\frac{2g\Delta h_{id}}{\rho_{id}}\right)^{1/2} \), ft/sec

\( W \) relative gas velocity, ft/sec

\( w \) weight flow, lb/sec

\( \alpha \) absolute rotor exit gas flow angle measured from axial direction, deg

\( \gamma \) ratio of specific heats

\( \delta \) ratio of inlet total pressure to U. S. standard sea-level pressure, \( \rho'_1/\rho^* \)

\( \epsilon \) function of \( \gamma \) used in relating parameters to that using air inlet conditions at U. S. standard sea-level conditions, \( \frac{0.740}{\gamma} \left(\frac{\gamma + 1}{2}\right)^{\gamma/\gamma-1} \)

\( \eta_s \) static efficiency (based on total- to static-pressure ratio across turbine)

\( \eta_{tot} \) total efficiency (based on total- to total-pressure ratio across turbine)

\( \theta_{cr} \) squared ratio of critical velocity at turbine inlet to critical velocity at U. S. standard sea-level temperature, \( (V_{cr}, 1/V_{cr})^2 \)

\( \mu \) gas viscosity, lb/(ft)(sec)

\( \nu \) blade-jet speed ratio (based on rotor inlet tip speed), \( U_t/V_j \)

Subscripts:

\( \text{cr} \) condition corresponding to Mach number of unity

\( \text{id} \) ideal

\( \text{w} \) outer wall
TURBINE DESCRIPTION

The 4.59-inch-tip-diameter radial-inflow turbine described in reference 2 was selected for this investigation. Air equivalent design values are as follows:

Equivalent weight flow, \( w_c \sqrt{\theta_{cr}/\delta} \), lb/sec ........................................... 0.616
Equivalent specific work, \( \Delta h/\theta_{cr} \), Btu/lb ........................................... 11.9
Equivalent speed, \( N\sqrt{\theta_{cr}} \), rpm ........................................... 29,550
Equivalent total- to static-pressure ratio, \( p'_1/p_3 \) ........................................... 1.540
Total to total efficiency, \( \eta_{tot} \) ........................................... 0.880
Total to static efficiency, \( \eta_S \) ........................................... 0.824
Blade-jet speed ratio, \( \nu \) ........................................... 0.697
Specific speed, \( N_S = NQ^{1/2}/(H')^{3/4} \), ft \((min)(sec^{1/2}) \) ........................................... 95.6

The range of specific speeds at equivalent design speed and pressure ratio was obtained by changing volume flow by using stators with different throat areas. This was done by essentially changing the stator blade angle. Three additional stators having nominal throat areas of 50, 75, and 125 percent of design were used to obtain nominal specific speeds of 68, 83, and 107.

Figure 1 shows the four stators and the rotor used in the investigation. The measured stator throat areas were 49.6, 75.3, 96.1, and 126.1 percent of design. Hereafter, each stator and rotor combination will be referred to as the 50-, 75-, 100-, and 125-percent configuration. One stator blade of each configuration had an elongated leading edge to block the flow from entering the small area end of the inlet scroll. A description of the 100-percent configuration including velocity diagrams is given in reference 2. The 100- and 125-percent stators each have 14 blades, whereas the 50- and 75-percent stators have 18 blades each. In order to maintain acceptable stator-blade
surface velocities, the 50-, 75-, and 125-percent stators have slightly different shapes than the 100-percent stator.

It may be noted that, although the throat area of the 100-percent stator was 3.9 percent smaller than design, results (as reported in ref. 2) showed that equivalent design weight flow was obtained at equivalent design speed and pressure ratio. Attainment of equivalent design weight flow results from the flow check procedure, in which the rotor throat area is increased by cutting back rotor trailing edges until equivalent design weight flow is obtained.

The rotor has 11 blades and 11 splitter vanes. These splitter vanes are used over the initial third of the rotor, thereby increasing the solidity in this region. The resultant decrease in loading was required at the hub to prevent low blade pressure-surface gas velocities.
APPARATUS, INSTRUMENTATION, AND METHODS

The test facility, instrumentation, and method of calculating performance parameters were the same as those described in reference 2, except that air was used as the working fluid. Figure 2 shows a cross-sectional sketch of the turbine test section and the instrument measuring stations. A varying area scroll was used to obtain uniform inlet conditions at the stator inlet. A center body was used at the rotor exit to obtain measurement of exit static pressure at the hub and at the outer wall. Radial surveys of total pressure, total temperature, and flow angle were made at the rotor exit.

The 100-percent configuration was tested at nominal inlet conditions of 16.0 pounds per square inch absolute and 540° R and resulted in a weight flow of 0.657 pound per second at equivalent design speed and pressure ratio. A nominal Reynolds number of 277,000 was calculated from this result; Reynolds number is defined herein as $Re = \frac{w \rho r_t}{\mu}$. In order to eliminate the effects of changes in Reynolds number on turbine efficiency, this parameter was held constant for all configurations at equivalent design speed and pressure ratio. Thus, the inlet total pressure was adjusted for the other configurations until a weight flow of approximately 0.657 pound per second was obtained. Table I shows the values of inlet total pressure and temperature and the pressure ratio.

Table I

<table>
<thead>
<tr>
<th>Station</th>
<th>Total Pressure</th>
<th>Total Temperature</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.12</td>
<td>120°C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.14</td>
<td>140°C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.16</td>
<td>160°C</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. - Turbine test section and instrumentation.
TABLE I. - EXPERIMENTAL OPERATING CONDITIONS

<table>
<thead>
<tr>
<th>Configuration, percent design</th>
<th>Inlet total pressure, psia</th>
<th>Inlet total temperature, °R</th>
<th>Pressure-ratio range</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>13.0</td>
<td>536</td>
<td>1.28 to 2.13</td>
</tr>
<tr>
<td>100</td>
<td>16.0</td>
<td>540</td>
<td>1.30 to 2.16</td>
</tr>
<tr>
<td>75</td>
<td>19.2</td>
<td>540</td>
<td>1.29 to 2.26</td>
</tr>
<tr>
<td>50</td>
<td>27.2</td>
<td>542</td>
<td>1.31 to 2.32</td>
</tr>
</tbody>
</table>

The turbine was rated on the basis of both total and static efficiency. Turbine inlet and exit total pressures were calculated from weight flow, static pressure, total temperature, and flow angle. In the calculations of turbine inlet total pressure, the flow was assumed to be normal to the plane defined by station 1. The exit total temperature was determined from turbine power measurements.

RESULTS AND DISCUSSION

The results of this investigation are presented in two sections. The first section includes overall results in terms of equivalent weight flow and efficiency for a range of pressure ratios at equivalent design speed with cold air as the working fluid. The effect of specific speed on turbine efficiency is then shown. The second section discusses the internal flow characteristics of the turbine as determined from exit radial surveys of angle and total- and static-pressure measurements through the turbine at equivalent design speed and pressure ratio.

Turbine Performance

Weight flow. - Figure 3 shows the variation of equivalent weight flow $\dot{w} = \sqrt{\theta_{cr}/\delta}$ with inlet total- to exit static-pressure ratio at equivalent design speed. Equivalent weight flows of 0.752, 0.615, 0.519, and 0.367 pound per second were obtained for the 125-, 100-, 75-, and 50-percent configurations at the equivalent design pressure ratio of 1.54. The variation of weight flow with increasing pressure ratio indicated that the flow was subsonic over the entire range of pressure ratios covered. The figure also shows that near choked flow conditions were obtained for the 50-percent configuration at the pressure ratio of 2.32. The combination of near choked flow conditions obtained for the 50-percent configuration and the flattening of the weight-flow curves with decreasing stator-throat area indicates that the velocity level through the stator blade row was increasing with decreasing stator throat area.

Figure 4 presents the variation of equivalent weight flow with stator throat area for equivalent design speed and pressure ratio. Equivalent weight flow is expressed as...
1.2

Inlet total-to-exit static-pressure ratio, \( p_2/p_1 \)

Percent configuration

- Design
- 125
- 100
- 75
- 50

Figure 3. - Variation of weight flow with pressure ratio and stator throat area at equivalent design speed.

a percentage of the experimental equivalent weight flow obtained with the 100-percent configuration. The dashed line shown on the figure represents the case where equivalent weight flow is directly proportional to stator throat area. Comparison of the experimental curve with the ideal case shows that the weight flow increases at a lower rate than the rate of area increase. This indicates that the stator pressure ratio \( p_2/p_1 \) increased with increasing stator throat area and, therefore, rotor reaction increased. This change in rotor reaction resulted from the variation of stator to rotor throat area ratios of the four configurations. The change in rotor reaction among the four configurations is discussed further in the section Internal Flow Characteristics.

Efficiency. - Figure 5 shows the variation of total and static efficiency with blade-jet speed ratio for each configuration. The highest efficiencies, at design blade-jet speed ratio, were obtained with the 75-percent configuration. Total and static
efficiencies were 0.91 and 0.87, respectively, for this configuration. These values are significantly higher than the total and static efficiencies of 0.89 and 0.83 obtained with the 100-percent configuration. At design blade-jet speed ratio, the lowest efficiencies, total and static, were obtained with the 125-percent configuration. These values were 0.85 and 0.77 for the total and static efficiencies, respectively.

The level of rotor exit velocity, as indicated by the difference between total and static efficiency, decreases with decreasing stator throat area. For example, at the design blade-jet speed ratio of 0.697, approximately 8 points in efficiency are attributed to rotor kinetic energy for the 125-percent configuration, while only 3.0 points in efficiency are attributed to rotor exit kinetic energy for the 50-percent configuration. This decrease in rotor exit velocity with decreasing stator throat area results from the change in the stator to rotor throat area ratio among the four configurations. The figure also indicates the variation of rotor exit kinetic energy with blade-jet speed ratio.
of figures 5(a) and (d) shows that there was a greater rate of change in exit kinetic energy with increasing blade-jet speed ratio (decreasing turbine pressure ratio $p_1/p_3$) for the 125-percent configuration than for the 50-percent configuration. This effect results from the larger variation of weight flow with pressure ratio for the 125-percent configuration than for the other configurations, as shown in figure 3 (p. 8).

Figure 6 shows the variation of total and static efficiency with specific speed for all four configurations investigated. The dashed line represents the variation of efficiency with specific speed at the design blade-jet speed ratio of 0.697. The upper plot in figure 6 shows that the highest total efficiency value of 0.91 was obtained at a specific speed of approximately 86. This efficiency value is 2.0 points higher than the efficiency of 0.89, which was obtained at the design specific speed value of 95.6 for the 100-percent or reference turbine configuration. It may be noted that the design blade-jet speed ratio curve (dashed line) passes through the peak efficiency point for all but the 50-percent configuration. The heavy curve shown in the figure represents the envelope of the efficiency curves for all configurations. This curve shows that maximum total efficiency is obtained in the specific speed range of about 80 to 90.

The lower plot in figure 6 shows the variation of static efficiency with specific speed for the four configurations. The highest efficiency value of 0.87 was also obtained at a specific speed of approximately 86. This value of efficiency is about 3.0 points higher than that obtained for the 100-percent or reference turbine configuration at the design specific speed of 95.6. The lowest peak static efficiency of 0.77 was obtained at a
specific speed of 111. It should be pointed out, however, that part of this decrease in static efficiency results from using the same rotor with each stator. Since the 125-percent configuration passes the largest volume flow of the four configurations, the rotor exit kinetic energy would be expected to be higher for this configuration.

The variation of static efficiency with specific speed for design blade-jet speed ratio (dashed line) shows the same trend as the envelope curve represented by the heavy line. The highest efficiency of 0.87 was obtained at a specific speed of 86, and the lowest efficiency of 0.77 was obtained at a specific speed of 108. It may be noted that both total and static efficiencies obtained at design blade-jet speed ratio occur at or very close to the peak efficiency points for the 75- and 100-percent configurations and at lower values of efficiency for the other configurations. From these results, it appears that radial-inflow turbines should be designed for a specific speed range of about 80 to 90 for the attainment of high efficiency.

Loss distribution. - In order to obtain an understanding of the losses which contributed to the variation of turbine performance with specific speed at design blade-jet speed ratio, an analysis was made to determine the magnitude of the various losses for each configuration. The method used involved the determination of velocity diagrams for each configuration from measured turbine work, weight flow, inlet conditions of pressure and temperature, speed, stator throat area, and results of rotor exit surveys of total pressure and flow angle. Design loss distribution between the stator and rotor was used to proportion the measured overall turbine loss for the 100-percent configuration. Stator losses for the other configurations were then assumed to vary in proportion to the average of inlet and outlet kinetic energy as determined from the velocity diagrams.

Rotor incidence losses were determined through adjustment of the actual incidence angle, which resulted in an effective relative whirl velocity different from the velocity diagram value. The adjustment depends upon the blade speed, the number of blades, the rotor diameter, and the volume flow at the rotor inlet. The use of the effective relative whirl velocity is analogous to the use of the slip factor for centrifugal impellers. The remaining losses were attributed to rotor viscous losses. Figure 7 shows the results of these calculations. The various losses, expressed in terms of efficiency, are shown as a function of specific speed. The magnitude of the exit kinetic-energy loss is shown by the difference between total and static efficiency values obtained from figure 5 (p. 9) at design blade-jet speed ratio.

Figure 7 shows that rotor incidence loss increases as specific speed increases above 90. This increase in rotor incidence loss results from an increase in the stator exit flow angle (as measured from tangential) and a decrease in the stator exit velocity with increasing specific speed. Rotor viscous losses also increase substantially with increasing specific speed. The increase in rotor loss results from the increased relative velocity level through the rotor. Part of the increase in rotor viscous loss can be attributed to the manner in which the range of specific speeds was obtained. Figure 7
shows that there is no significant change in stator viscous losses as the specific speed is increased above a value of 86. This would indicate that the combined losses resulting from the velocity level through the stators and the boundary-layer blockage did not change to any large degree.

Decreasing specific speed below a value of 86 results in an increase in stator viscous losses. The figure shows that the losses increase from about 5.0 points in terms of efficiency at a specific speed of 86 to about 11.0 points at a specific speed of approximately 72. These losses may be associated with the increased velocity level through the stator and the increased boundary-layer blockage due to a larger ratio of wetted area to flow area.

Calculations also indicated that rotor incidence losses were insignificant below a specific speed of 90. Rotor viscous and exit kinetic-energy losses decreased with decreasing specific speed, since specific speed is proportional to the square root of the exit velocity when rotative speed, rotor throat area, and pressure ratio are constant.

**Internal Flow Characteristics**

The determination of turbine internal-flow characteristics for each configuration was based on the measured static-pressure distribution through the turbine, together with the results of a radial survey of turbine exit total pressure and flow angle.

Figure 8 shows the variation in stator exit static pressure with design stator throat area at equivalent design speed and pressure ratio for the four configurations investigated. Rotor reaction decreases and stator exit velocity increases with decreasing stator throat area, as was noted in the discussion of equivalent weight flow.
Figure 8. - Variation of stator exit static pressure with stator throat area at equivalent design speed and pressure ratio.

Figure 9. - Variation of turbine exit flow angle with radius ratio at equivalent design speed and pressure ratio.

The results of a radial survey of exit flow angle taken at equivalent design speed and pressure ratio are presented in figure 9 for the four configurations investigated. It may be noted that, as the stator throat area was reduced, the exit flow angle changed from predominately overturning (as denoted by negative angles) to underturning over the entire passage height. This trend in exit flow angle with configuration is to be expected since the rotor exit relative velocity decreases with decreasing stator throat area.

The variation of exit flow angle and exit total and static pressure with radius ratio indicated that there was a nonuniform work distribution from hub to outer wall for all four configurations, with minimum work occurring along the outer wall. This may be due to blade unloading which results from tip leakage and from centrifugation of low
momentum fluid to this region.

Local values of total efficiency were calculated on the basis of the change in tangential momentum through the rotor and the radial distribution of total pressure at the rotor exit. These results are plotted in figure 10 in terms of turbine loss \((1.0 - \eta_{\text{tot}})\) as a function of radius ratio. The figure shows that the largest radial variation in loss or efficiency is obtained with the 75- and 100-percent configurations. However, the magnitudes of the losses for these two configurations are substantially lower than those for the other two configurations. The curves show low losses along the hub region and comparatively high losses in the region near the outer wall for all configurations.

Calculations were made from experimental results to determine the radius ratio at which the weight flow was divided into equal parts. A radius ratio of approximately 0.77 was calculated for all configurations. This coincides with the design mean streamline, as shown in figure 10. At the mean streamline the calculated local loss is approximately equal to the experimental value as obtained from overall performance for each configuration.

**SUMMARY OF RESULTS**

An experimental investigation was made to determine the specific speed effect on performance for a 4.59-inch-tip-diameter radial-inflow turbine at equivalent design speed over a range of pressure ratios. Results are presented for operation at a nominally constant Reynolds number of 277 000 at equivalent design speed and pressure ratio. The effect of turbine size on performance was eliminated by use of the same rotor for each configuration. The range of specific speed values investigated at equivalent design speed and pressure ratio was obtained by changing volume flow through the use of stators having throat areas nominally 50, 75, 100, and 125 percent of design. From this investigation the following results were obtained:

1. Comparison of actual equivalent weight flow with an equivalent weight flow, which is directly proportional to stator throat area, showed that there was a deficiency in
weight flow for the 125-percent configuration and a surplus of weight flow for the 50- and 75-percent configurations at equivalent design speed and pressure ratio. This difference in weight flows was attributed to the corresponding changes in rotor reaction which result from the use of the same rotor for each configuration.

2. Maximum total and static efficiencies were obtained in the specific speed range of about 80 to 90. In this range, peak total and static efficiencies of 0.91 and 0.87 were obtained with the 75-percent configuration at a specific speed of 86. The lowest peak value of efficiency was obtained with the 125-percent configuration. For this case, the total efficiency was 0.85 at a specific speed of approximately 108. The corresponding static efficiency was 0.77 at a specific speed of 111.

3. An analysis of stator and rotor losses over the range of specific speeds investigated at equivalent design speed and pressure ratio showed the following:

(a) Turbine losses were at a minimum for the specific speed range of about 80 to 90.

(b) Stator viscous losses were the predominant factor in the decrease in total efficiency as the specific speed was decreased from a value of 86.

(c) Rotor incidence and viscous losses were the predominant factors in the decrease in total efficiency as specific speed was increased from 90.

4. Stator-exit static-pressure measurements obtained at equivalent-design speed and pressure ratio indicated that the highest rotor reaction was obtained for the 125-percent configuration. Rotor reaction decreased with decreasing stator throat area.

5. Radial surveys of rotor exit total pressure and flow angle at equivalent design speed and pressure ratio indicated that minimum losses occurred near the hub region and the losses increased substantially near the outer wall. These increased losses may have resulted from tip leakage effects with blade unloading and from centrifugation of low-momentum fluid toward the outer wall. Comparison of losses between configurations showed that minimum losses were obtained from hub to outer wall for the 75-percent configuration.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, August 26, 1966,
120-27-03-13-22.
REFERENCES


\( m_i \) 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, °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

\( \alpha \) angle between thrust and velocity vectors (sketch (a)), deg

\( \beta \) angle of rotation of thrust out of orbit plane (sketch (a)), deg

\( \eta \) power efficiency factor

\( \mu \) \( k^2 m_r \)

\( \rho \) atmospheric density, kg/m\(^3\)
\( \omega \) argument of pericenter, radians
\( \Omega \) 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

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

In $x,y,z$ component form,

$$ V'_x = V_x + \omega y $$
$$ V'_y = V_y - \omega x $$
$$ V'_z = V_z $$

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 $\alpha$ and $\beta$, 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 $\vec{A}$, $\vec{P}$, and $\vec{V}'$ form an orthogonal set. Thus,

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

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

$$ \vec{F} \cdot \vec{V}' = F V' \cos \alpha $$
$$ \vec{F} \cdot \vec{A} = FA \sin \alpha \sin \beta $$
$$ \vec{F} \cdot \vec{P} = FP \sin \alpha \cos \beta $$

Solving for $\vec{F}$ yields

$$ \vec{F} = \frac{F}{P^2} \left( V' \cos \alpha \vec{A} \times \vec{P} + A \sin \alpha \sin \beta \vec{P} \times \vec{V}' + P \sin \alpha \cos \beta \vec{F} \right) $$
or, in \( x,y,z \) component form,

\[
F_x = \frac{F}{p^2} \left[ \dot{V}' \cos \alpha (A_y P_z - A_z P_y) + A \sin \alpha \sin \beta (P_y V_z' - P_z V_y') + P \sin \alpha \cos \beta P_x \right]
\]

(B7a)

\[
F_y = \frac{F}{p^2} \left[ \dot{V}' \cos \alpha (A_z P_x - A_x P_z) + A \sin \alpha \sin \beta (P_z V_x' - P_x V_z') + P \sin \alpha \cos \beta P_y \right]
\]

(B7b)

\[
F_z = \frac{F}{p^2} \left[ \dot{V}' \cos \alpha (A_x P_y - A_y P_x) + A \sin \alpha \sin \beta (P_x V_y' - P_y V_x') + P \sin \alpha \cos \beta P_z \right]
\]

(B7c)

Aerodynamic Lift and Drag Resolution Along \( x,y,z \) Axes

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

\[
\vec{D} = -D \frac{V_x'}{V'} - D \frac{V_y'}{V'} - D \frac{V_z'}{V'}
\]

(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:

\[
\vec{L} \cdot \vec{V}' = 0
\]

(B9a)

\[
\vec{L} \cdot \vec{A} = L A \sin \beta
\]

(B9b)

\[
\vec{L} \cdot \vec{P} = L P \cos \beta
\]

(B9c)

Solving for \( \vec{L} \) yields

\[
\vec{L} = \frac{L}{p^2} \left( A \sin \beta \vec{P} \times \vec{V}' + P \cos \beta \vec{P} \right)
\]

(B10)

or, in \( x,y,z \) component form,

\[
L_x = \frac{L}{p^2} \left[ A \sin \beta (P_y V_z' - P_z V_y') + P \cos \beta P_x \right]
\]

(B11a)

\[
L_y = \frac{L}{p^2} \left[ A \sin \beta (P_z V_x' - P_x V_z') + P \cos \beta P_y \right]
\]

(B11b)

\[
L_z = \frac{L}{p^2} \left[ A \sin \beta (P_x V_y' - P_y V_x') + P \cos \beta P_z \right]
\]

(B11c)
From spherical trigonometry used in reference to the celestial sphere shown in sketch (c), the following relations may be derived for the position coordinates:

\begin{align*}
  x &= r(\cos \Omega \cos u - \sin \Omega \sin u \cos i) \\
  y &= r(\sin \Omega \cos u + \cos \Omega \sin u \cos i) \\
  z &= r(\sin u \sin i)
\end{align*}

where

\begin{align*}
  r &= \frac{p}{1 + e \cos v} \\
  u &= \omega + v
\end{align*}

and \( v \) can be obtained from

\begin{equation}
  \cos v = \frac{\cos E - e}{1 - e \cos E}
\end{equation}

and

\begin{equation}
  M = E - e \sin E
\end{equation}

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 \( \dot{r} = \sqrt{\frac{\mu}{p}} \sin v \).
\[ \dot{x} = - \sqrt{\frac{\mu}{p}} (N \cos i \sin \Omega + Q \cos \Omega) \]  
\[ \dot{y} = \sqrt{\frac{\mu}{p}} (N \cos i \cos \Omega - Q \sin \Omega) \]  
\[ \dot{z} = \sqrt{\frac{\mu}{p}} (N \sin i) \]

where

\[ N = e \cos \omega + \cos u \]  
\[ Q = e \sin \omega + \sin u \]
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

\[
X_{1}^{2} = X_{2} - X_{1} = \frac{1}{6} (k_{1} + 2k_{2} + 2k_{3} + k_{4}) \tag{D1}
\]

where

\[ X = \text{a dependent variable} \]

\[ X_{1}^{2} = \text{increment in the dependent variable} \]

\[ h_{2} = \text{increment in the independent variable} \]

\[
k_{1} = h_{2} \dot{X}_{2}(t_{1}, X_{1})
\]

\[
k_{2} = h_{2} \dot{X}_{2}\left(t_{1} + \frac{h_{2}}{2}, X_{1} + \frac{k_{1}}{2}\right)
\]

\[
k_{3} = h_{2} \dot{X}_{2}\left(t_{1} + \frac{h_{2}}{2}, X_{1} + \frac{k_{2}}{2}\right)
\]

\[
k_{4} = h_{2} \dot{X}_{2}(t_{1} + h_{2}, X_{1} + k_{3})
\]

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 Lagrange interpolation formula gives the derivative at any time \( t_{0} \leq t \leq t_{2} \):

\[
\dot{X} = \dot{X}_{0} \left(\frac{t - t_{1})(t - t_{2})}{h_{1}(h_{1} + h_{2})}\right) - \dot{X}_{1} \left(\frac{(t - t_{0})(t - t_{2})}{h_{1}h_{2}}\right) + \dot{X}_{2} \left(\frac{(t - t_{0})(t - t_{1})}{h_{2}(h_{1} + h_{2})}\right) \tag{D2}
\]

Integration of this equation from \( t_{1} \) to \( t_{2} \) yields

\[
X_{1}^{2} = \frac{1}{6} \left[ \left(\frac{h_{2}}{h_{1}}\right)^{2} \left(\frac{-h_{2}}{1 + \frac{h_{2}}{h_{1}}}\right) \dot{X}_{0} + \frac{h_{2}}{h_{1}} (h_{2} + 3h_{1}) \dot{X}_{1} + \left(2h_{2} + \frac{h_{2}}{h_{1}}\right) \dot{X}_{2} \right] \tag{D3}
\]
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 \( \bar{X} \) to obtain the relative error \( \delta_2 \). Thus,

\[
\delta_2 = \left| \frac{X'_1 - X'_2}{\bar{X}} \right|^2
\]

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

\[
\delta = Ah^5
\]

(\text{where } A \text{ is a suitable coefficient}) or in the logarithmic form

\[
\log \delta = A' + 5 \log h
\]

where

\[
A' = \log A
\]

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

\[
A'_3 = A'_2 + \frac{A'_2 - A'_1}{t_2 - t_1} (t_3 - t_2)
\]

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

\[
A'_3 = A'_2 + \frac{(A'_2 - A'_1) h_3}{h_2}
\]

and on this basis \( \delta_3 \) would be predicted to be

\[
\log \delta_3 = A'_2 + 5 \log h_3
\]

It is desired that \( \delta_3 \) should approximate \( \delta \), the reference error; therefore,

\[
\log h_3 = \frac{1}{5} (\log \delta - A'_1)
\]

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 \( \delta \). 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 ($\delta > \delta_{\text{limit}}$), a reduced step size must be used. It may be obtained from the reference error $\delta$ as

$$h_3 = \exp\left[\frac{1}{6} (\log \delta - A'_2)\right]$$  \hspace{1cm} (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 $h_2 = h_1$,

$$X'_{\text{l}} = \frac{h_1}{3} (\dot{x}_0 + 4\dot{x}_1 + \dot{x}_2)$$  \hspace{1cm} (D10)

Failures. Should two consecutive predictions of the same step fail to produce an error $\delta$ less than $\delta_{\text{limit}}$, 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.
## GLOSSARY OF VARIABLES

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>COMMON LOCATION</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (700)</td>
<td>C(11)</td>
<td>ARRAY CONTAINING THE INITIAL DATA AND THE PROGRAM CONTROL VARIABLES</td>
</tr>
<tr>
<td>A1</td>
<td>B(10)</td>
<td>ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(1)</td>
</tr>
<tr>
<td>A2</td>
<td>B(11)</td>
<td>ERROR CONTROL PARAMETER DEFINED BY EQ. (D6A) AT T(2)</td>
</tr>
<tr>
<td>ACOEF1</td>
<td>B(12)</td>
<td>INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)</td>
</tr>
<tr>
<td>ACOEF2</td>
<td>B(13)</td>
<td>INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)</td>
</tr>
<tr>
<td>ACOEF3</td>
<td>B(14)</td>
<td>INTERPOLATION POLYNOMIAL COEFFICIENT FOR VARIABLE STEP SIZE, EQ. (D3)</td>
</tr>
<tr>
<td>AEXIT1 (10)</td>
<td>A(103)</td>
<td>ENGINE EXIT AREAS FOR AT MOST 10 STAGES, M**2</td>
</tr>
<tr>
<td>AEXIT</td>
<td>B(3)</td>
<td>AEXIT1(NSTAGE)</td>
</tr>
<tr>
<td>AK (3)</td>
<td>A(51)</td>
<td>RUNGE KUTTA COEFFICIENTS, SET IN STDATA</td>
</tr>
<tr>
<td>ALPHA</td>
<td>A(49)</td>
<td>ANGLE BETWEEN VELOCITY AND THRUST VECTORS, SEE SKETCH (A)</td>
</tr>
<tr>
<td>ALT</td>
<td>A(4)</td>
<td>VEHICLE ALTITUDE ABOVE EARTH, M</td>
</tr>
<tr>
<td>AM</td>
<td>B(90)</td>
<td>TOTAL VEHICLE ANGULAR MOMENTUM PER UNIT MASS, M**2/SEC</td>
</tr>
<tr>
<td>AMASS (30)</td>
<td>A(347)</td>
<td>PERMANENT LIST OF BODY MASSES IN ORDER OF PNAME LIST, SET IN STDATA, MASSES FROM ELIPS DATA BEGIN AT AMASS (21), SUN MASS UNITS</td>
</tr>
<tr>
<td>AMC (3)</td>
<td>B(87)</td>
<td>X, Y, Z COMPONENTS OF ANGULAR MOMENTUM PER UNIT MASS, **Z)/SEC</td>
</tr>
<tr>
<td>AMSQRD</td>
<td>B(91)</td>
<td>SQUARE OF TOTAL ANGULAR MOMENTUM PER UNIT MASS, M<strong>4/SEC</strong>2</td>
</tr>
<tr>
<td>AREA1 (10)</td>
<td>A(113)</td>
<td>AERODYNAMIC REFERENCE AREAS FOR AT MOST 10 STAGES, M**2</td>
</tr>
<tr>
<td>AREA</td>
<td>B(6)</td>
<td>AREA1(NSTAGE)</td>
</tr>
<tr>
<td>ASYMP</td>
<td>A(7)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>ATMN</td>
<td>A(21)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>AU</td>
<td>A(29)</td>
<td>ASTRONOMICAL UNIT, M</td>
</tr>
<tr>
<td>AW (4)</td>
<td>A(55)</td>
<td>RUNGE KUTTA COEFFICIENTS, SET IN STDATA</td>
</tr>
<tr>
<td>AZI</td>
<td>A(35)</td>
<td>INITIAL AZIMUTH ANGLE, USED WHEN IMODE = 4, SEE SKETCH (B), DEGREES</td>
</tr>
</tbody>
</table>
ARRAY CONTAINING INTERNAL PARAMETERS NOT UNDER USER CONTROL

ANGLE BETWEEN VELOCITY-THRUST PLANE AND ORBIT PLANE, SEE SKETCH(A)

BODY MASSES SELECTED FROM AMASS LIST IN SEQUENCE CORRESPONDING TO BNAME LIST

ORDERED LIST OF BCD BODY NAMES

ORIGINAL UNORDERED LIST OF BCD BODY NAMES READ IN AT INPUT

AUXILIARY ORDERED LIST OF BCD BODY NAMES

TOTAL DRAG COEFFICIENT

INDUCED DRAG COEFFICIENT

SMALLEST CRITICAL RADIUS (RBCRIT(J)) WITHIN WHICH OBJECT LIES

COSINE OF INCLINATION

CIRCUMFERENTIAL COMPONENT OF TOTAL PERTURBATIVE ACCELERATION, M/SEC**2

LIFT COEFFICIENT

SEE TABLE II

STORAGE ARRAY FOR COEFFICIENTS USED TO COMPUTE ALPHA, CL, CDI, CD OR OTHER PARAMETERS

COMPONENTS OF TOTAL PERTURBATIVE ACCELERATION ALONG X, Y, Z AXES.

SEE TABLE II

SEE TABLE II

COSINE OF ALPHA

COSINE OF BETA

COSINE OF TRU

COSINE OF THE ARGUMENT OF LATITUDE

ARRAY WHERE SAVED DATA IS STORED FOR LATER USE. ARRAYS A, XPRIM, AND XPRIMB MAY BE SAVED.

SEE TABLE II

OUTPUT CONTROL PARAMETER USED IN STEP

INITIAL STEP SIZES FOR AT MOST 10 STAGES, SEC

DELT1(INSTAGE)

ATMOSPHERIC DENSITY, KG/M**3

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(15)  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 (8), 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

MBODY5  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  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 PNAAL 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
PUSH0 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
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVS</td>
<td>Revolution counter, used only for output</td>
</tr>
<tr>
<td>REVOLV</td>
<td>Rotation rate of reference body when ATMN=BNAME(1), RAD/SEC</td>
</tr>
<tr>
<td>RMASS1 (10)</td>
<td>Initial masses for at most 10 stages, KG</td>
</tr>
<tr>
<td>ROTATE</td>
<td>Rotation rate of a reference body, RAD/SEC</td>
</tr>
<tr>
<td>RSQRD</td>
<td>Radius squared of object to origin, M**2</td>
</tr>
<tr>
<td>SIGNAL</td>
<td>See Table II</td>
</tr>
<tr>
<td>SIMP1 (10)</td>
<td>Specific impulses for at most 10 stages, SEC</td>
</tr>
<tr>
<td>SIMP</td>
<td>SIMP1(NSTAGE)</td>
</tr>
<tr>
<td>SINALF</td>
<td>Sine of alpha</td>
</tr>
<tr>
<td>SINBET</td>
<td>Sine of beta</td>
</tr>
<tr>
<td>SINTRU</td>
<td>Sine of true</td>
</tr>
<tr>
<td>SINGL</td>
<td>Sine of inclination</td>
</tr>
<tr>
<td>SINV</td>
<td>Sine of the argument of latitude</td>
</tr>
<tr>
<td>SPACES</td>
<td>Number of equal time units until next output</td>
</tr>
<tr>
<td>SPD</td>
<td>Seconds per day, set in STDATA, SEC/DAY</td>
</tr>
<tr>
<td>SQRDK1</td>
<td>Gravitational constant of the Sun, AU<strong>3/DAY</strong>2</td>
</tr>
<tr>
<td>SQRDK</td>
<td>Gravitational constant of the Sun, M<strong>3/SEC</strong>2</td>
</tr>
<tr>
<td>STEPMX</td>
<td>See Table II</td>
</tr>
<tr>
<td>STEPS</td>
<td>See Table II</td>
</tr>
<tr>
<td>STEPNO</td>
<td>Count of unsuccessful integration steps (those which do not pass error control test)</td>
</tr>
<tr>
<td>SWLOOK</td>
<td>See Table II</td>
</tr>
<tr>
<td>TABLT</td>
<td>Time measured relative to the Julian date of takeoff, days</td>
</tr>
<tr>
<td>TABLE (200)</td>
<td>Array of input parameters and their common store locations</td>
</tr>
<tr>
<td>TAPE3</td>
<td>See Table II</td>
</tr>
<tr>
<td>TB (10)</td>
<td>Flight times for at most 10 stages, SEC</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>TDATA (6,3,7) B(265)</td>
<td>COEFFICIENTS FROM EPHEMERIDES TAPE TO BE USED IN DETERMINING POSITIONS AND POSSIBLY VELOCITIES OF TURBATING BODIES, ONE SET FOR EACH OF 7 BODIES</td>
</tr>
<tr>
<td>TDEL (7) B(170)</td>
<td>ONE-HALF OF TIME SPACING BETWEEN TWO ADJACENT ENTRIES OF LIKE BODY NAME ON EPHEMERIDES TAPE, READ FROM TAPE FOR EACH BODY</td>
</tr>
<tr>
<td>TFILE A(6)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>TIM (7) B(163)</td>
<td>TIME FOR SET OF EPHEMERIS DATA, READ FROM EPHEMERIDES TAPE, ONE FOR EACH BODY</td>
</tr>
<tr>
<td>TKICK A(15)</td>
<td>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</td>
</tr>
<tr>
<td>TM B(34)</td>
<td>ATMOSPHERIC TEMPERATURE TIMES THE RATIO OF MOLECULAR TO ACTUAL MOLECULAR WEIGHT, DEGREES KELVIN</td>
</tr>
<tr>
<td>TMAX B(4)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>TMIN A(18)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>TOFFT A(24)</td>
<td>FRACTIONAL PART OF JULIAN DATE OF TAKEOFF, DAYS</td>
</tr>
<tr>
<td>TRSFER B(8)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>TRU B(40)</td>
<td>TRUE ANOMALY, RAD</td>
</tr>
<tr>
<td>TTEST A(54)</td>
<td>SEE TABLE II</td>
</tr>
<tr>
<td>ITOL A(45)</td>
<td>TIME TOLERANCE WITHIN WHICH PROBLEM TIME MINUS TMAX MUST LIE TO END STAGE</td>
</tr>
<tr>
<td>U A(59)</td>
<td>ECCENTRIC ANOMALY, RAD</td>
</tr>
<tr>
<td>V B(95)</td>
<td>VELOCITY OF OBJECT RELATIVE TO THE ORIGIN, M/SEC</td>
</tr>
<tr>
<td>VATM (3) B(97)</td>
<td>X,Y,Z COMPONENTS OF THE RELATIVE VELOCITY, VQ, M/SEC</td>
</tr>
<tr>
<td>WEFM (3,8) B(241)</td>
<td>X,Y,Z COMPONENTS OF OBJECT VELOCITY RELATIVE TO ALL BODIES, M/SEC</td>
</tr>
<tr>
<td>VEL A(37)</td>
<td>INITIAL RELATIVE VELOCITY, USED WHEN IMODE=4, SKETCH (8), M/SEC</td>
</tr>
<tr>
<td>WMACH B(138)</td>
<td>MACH NUMBER OF OBJECT</td>
</tr>
<tr>
<td>VQ B(100)</td>
<td>VELOCITY OF OBJECT RELATIVE TO ATMOSPHERE, M/SEC</td>
</tr>
<tr>
<td>VQSQRD B(101)</td>
<td>SQUARE OF VQ, M<strong>2/SEC</strong>2</td>
</tr>
<tr>
<td>VSQRD B(96)</td>
<td>SQUARE OF V, M<strong>2/SEC</strong>2</td>
</tr>
<tr>
<td>VX B(92)</td>
<td>X COMPONENT OF VELOCITY, M/SEC</td>
</tr>
<tr>
<td>VY B(93)</td>
<td>Y COMPONENT OF VELOCITY, M/SEC</td>
</tr>
<tr>
<td>VZ B(94)</td>
<td>Z COMPONENT OF VELOCITY, M/SEC</td>
</tr>
<tr>
<td>X (100) B(401)</td>
<td>WORKING SET OF INTEGRATION VARIABLES</td>
</tr>
<tr>
<td>XDOT (100) B(501)</td>
<td>TIME DERIVATIVES OF THE SET X</td>
</tr>
</tbody>
</table>
XIYF (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 DISCRIMINATION 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
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,

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

(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

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

and its derivative

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

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:

<table>
<thead>
<tr>
<th>Word number</th>
<th>Data</th>
<th>Mode</th>
<th>Definitions and/or units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Planet name</td>
<td>BCD</td>
<td>Six characters (first six)</td>
</tr>
<tr>
<td>2</td>
<td>Julian date</td>
<td>Floating point</td>
<td>Date of midpoint of fit, Julian date</td>
</tr>
<tr>
<td>3</td>
<td>Delta T</td>
<td></td>
<td>Number of days on each side of midpoint</td>
</tr>
<tr>
<td>4</td>
<td>$x$</td>
<td></td>
<td>$a_{\text{day}}$</td>
</tr>
<tr>
<td>5</td>
<td>$y$</td>
<td></td>
<td>$a_{\text{day}}^4$</td>
</tr>
<tr>
<td>6</td>
<td>$z$</td>
<td></td>
<td>$a_{\text{day}}^4$</td>
</tr>
<tr>
<td>7</td>
<td>$\sigma$</td>
<td></td>
<td>$a_{\text{day}}^3$</td>
</tr>
<tr>
<td>8</td>
<td>$\delta$</td>
<td></td>
<td>$a_{\text{day}}^3$</td>
</tr>
<tr>
<td>9</td>
<td>$a$</td>
<td></td>
<td>$a_{\text{day}}^2$</td>
</tr>
<tr>
<td>10</td>
<td>$b$</td>
<td></td>
<td>$a_{\text{day}}^2$</td>
</tr>
<tr>
<td>11</td>
<td>$c$</td>
<td></td>
<td>$a_{\text{day}}$</td>
</tr>
<tr>
<td>12</td>
<td>$\sigma'$</td>
<td></td>
<td>$a_{\text{day}}^5$</td>
</tr>
<tr>
<td>13</td>
<td>$\delta'$</td>
<td></td>
<td>$a_{\text{day}}^5$</td>
</tr>
<tr>
<td>14</td>
<td>$\sigma$</td>
<td></td>
<td>$a_{\text{day}}^4$</td>
</tr>
<tr>
<td>15</td>
<td>$\delta$</td>
<td></td>
<td>$a_{\text{day}}^4$</td>
</tr>
<tr>
<td>16</td>
<td>$a$</td>
<td></td>
<td>$a_{\text{day}}^3$</td>
</tr>
<tr>
<td>17</td>
<td>$b$</td>
<td></td>
<td>$a_{\text{day}}^3$</td>
</tr>
<tr>
<td>18</td>
<td>$c$</td>
<td></td>
<td>$a_{\text{day}}^2$</td>
</tr>
<tr>
<td>19</td>
<td>$\sigma'$</td>
<td></td>
<td>$a_{\text{day}}^2$</td>
</tr>
<tr>
<td>20</td>
<td>$\delta'$</td>
<td></td>
<td>$a_{\text{day}}$</td>
</tr>
<tr>
<td>21</td>
<td>$\sigma$</td>
<td></td>
<td>$a_{\text{day}}$</td>
</tr>
<tr>
<td>22</td>
<td>$\delta$</td>
<td></td>
<td>$a_{\text{day}}$</td>
</tr>
</tbody>
</table>

*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:

Set 1

\[
\begin{align*}
\text{1st word:} & \quad \text{FORTRAN compatible} \\
\text{2nd word:} & \quad \text{file number, fixed point in decrement} \\
\text{3rd word:} & \quad \text{planet name, code in BCD, first six characters} \\
\text{4th word:} & \quad \text{Julian date, floating point} \\
\text{23rd word:} & \quad z \\
\end{align*}
\]

Set 2

\[
\begin{align*}
\text{24th word:} & \quad \text{planet name, code in BCD, first six characters} \\
\text{25th word:} & \quad \text{Julian date, floating point} \\
\text{44th word:} & \quad z \\
\end{align*}
\]

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

Set 12

\[
\begin{align*}
\text{234th word:} & \quad \text{planet name} \\
\text{235th word:} & \quad \text{Julian date, floating point} \\
\text{254th word:} & \quad z \\
\text{End-of-record gap} \\
\end{align*}
\]

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, \(7.29211565 \times 10^{-5}\) 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:

\[ X = \text{x-component of position in } x,y,z \text{ coordinate system, m} \]
\[ Y = \text{y-component of position in } x,y,z \text{ coordinate system, m} \]
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 "ALP" 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

ORLATN = (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:

\[
\text{TB} = \text{burning time for 1st stage, 2nd stage, etc., sec}
\]

\[
\text{FLOW} = \text{propellant flow rate for 1st stage, 2nd stage, etc., kg/sec}
\]

\[
\text{SIMP} = \text{vacuum specific impulse of 1st stage, 2nd stage, etc., sec}
\]

\[
\text{AREA} = \text{aerodynamic reference area of 1st stage, 2nd stage, etc., m}^2
\]

\[
\text{AEXIT} = \text{engine exit area for 1st stage, 2nd stage, etc., m}^2
\]

\[
\text{RMASS} = \text{initial mass or jettison mass for 1st stage, 2nd stage, etc., kg}
\]

\[
\text{DELT} = \text{initial integration step size for 1st stage, 2nd stage, etc., sec}
\]

\[
\text{IDENT} = \text{input identification number 1st stage, 2nd stage, etc.}
\]

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\text{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

\[
\text{BETA} = \text{angle } \beta, \text{ deg (see sketch (a) (p. 4))}
\]

\[
\text{COEFN} (I) = \text{angle-of-attack schedule, } \alpha = \alpha(t) \text{ (see instruction (16))}
\]

\[
\text{ICC} = \text{fixed-point integer (see instruction (16))}
\]

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):

\[
\text{ATMN} = \text{name of body that has atmosphere, in "ALF" format, (Earth)}
\]

\[
\text{RATM} = \text{radius above which atmospheric forces are not to be considered, m}
\]

\[
\text{ROTATE} = \text{atmospheric-rotation rate, radians/sec } (7.29211585 \times 10^{-5} \text{ for Earth})
\]

\[
\text{BETA} = \text{angle } \beta, \text{ deg (see sketch (a))}
\]
COEFN (I) = angle-of-attack schedule, \( \alpha = \alpha(t) \), \( C_L/\sin \alpha \), \( C_D,0 \), 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 \( \alpha(t), C_L/\sin \alpha, C_D,0, \) 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:

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

The coefficients \( a_{1,j}, a_{2,j}, \ldots \) should then be loaded into the COEFN(J) array as:

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

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:

\[
\text{COEFN(J)} = \alpha \text{ coefficients, } C_L/\sin \alpha \text{ coefficients, } C_D,0 \text{ coefficients, etc.}
= t_1, a_{11}, a_{12}, \ldots, t_n, N_M, l, b_{11}, b_{12}, \ldots, N_M, k, \text{ etc.}
\]

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:

\[
\text{ICC(1)} = \text{fixed-point value of } J \text{ where } \alpha \text{ coefficients begin}
\]

\[
\text{ICC(2)} = \text{fixed-point value of } J \text{ where } C_L/\sin \alpha \text{ coefficients begin}
\]

\[
\text{ICC(3)} = \text{fixed-point value of } J \text{ where } C_D,0 \text{ coefficients begin}
\]

\[
\text{ICC(4)} = \text{fixed-point value of } J \text{ where } C_D,i/C_L^2 \text{ coefficients begin}
\]

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; in appendix D
- \( ERLIMT \) = maximum value of \( \delta \) 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 \( EREF = 1 \times 10^{-6}, ERLIMT = 3 \times 10^{-6}. \)

(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) ≥ 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 $\text{DATA} = 101$, 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 $\text{DATA} = 300$ group if an ephemerides tape is required. This group is followed by the $\text{DATA} = 1$ 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 $\text{DATA} = 101$, $\text{DATA} = \text{INLOOK}$, or $\text{DATA} = \text{IDENT (NSTAGE)}$ 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 $\text{DATA} = 300$ 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:

<table>
<thead>
<tr>
<th>Takeoff time</th>
<th>Position and velocity</th>
<th>Spherical</th>
</tr>
</thead>
<tbody>
<tr>
<td>(completely fill in one and only one block)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rectangular</td>
<td>Orbit elements</td>
</tr>
<tr>
<td></td>
<td>X =</td>
<td>E =</td>
</tr>
<tr>
<td></td>
<td>Y =</td>
<td>OMEGA =</td>
</tr>
<tr>
<td></td>
<td>Z =</td>
<td>NODES =</td>
</tr>
<tr>
<td></td>
<td>VX =</td>
<td>INCL =</td>
</tr>
<tr>
<td></td>
<td>VY =</td>
<td>MA =</td>
</tr>
<tr>
<td></td>
<td>VZ =</td>
<td>P =</td>
</tr>
<tr>
<td></td>
<td>IMODE = 2</td>
<td>IMODE = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference and perturbing bodies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{BODYCD} = )</td>
</tr>
<tr>
<td>(\text{TAPE} = 3 = 0)</td>
</tr>
<tr>
<td>(\text{ELIPS} = )</td>
</tr>
<tr>
<td>(\text{BEGIN} = )</td>
</tr>
<tr>
<td>(\text{END} = )</td>
</tr>
<tr>
<td>(\text{TFILE} = )</td>
</tr>
<tr>
<td>(\text{IMODE} = 3)</td>
</tr>
</tbody>
</table>

**INPUT CHECK LIST**

<table>
<thead>
<tr>
<th>Takeoff time</th>
<th>Position and velocity</th>
<th>Spherical</th>
</tr>
</thead>
<tbody>
<tr>
<td>(completely fill in one and only one block)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rectangular</td>
<td>Orbit elements</td>
</tr>
<tr>
<td></td>
<td>X =</td>
<td>E =</td>
</tr>
<tr>
<td></td>
<td>Y =</td>
<td>OMEGA =</td>
</tr>
<tr>
<td></td>
<td>Z =</td>
<td>NODES =</td>
</tr>
<tr>
<td></td>
<td>VX =</td>
<td>INCL =</td>
</tr>
<tr>
<td></td>
<td>VY =</td>
<td>MA =</td>
</tr>
<tr>
<td></td>
<td>VZ =</td>
<td>P =</td>
</tr>
<tr>
<td></td>
<td>IMODE = 2</td>
<td>IMODE = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output control</th>
<th>Error control</th>
<th>Restart feature</th>
<th>Parameter search</th>
<th>Atmosphere and coefficients</th>
<th>Oblateness</th>
<th>Stage data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EREF =</td>
<td>NSAVE =</td>
<td>LOOKX =</td>
<td>ATMN =</td>
<td>OBLATN =</td>
<td>TB =</td>
</tr>
<tr>
<td></td>
<td>MODOUT =</td>
<td>RRLIMIT =</td>
<td>RECALL =</td>
<td>XLOOK =</td>
<td>RATM =</td>
<td>ROTATE =</td>
</tr>
<tr>
<td></td>
<td>STEPS =</td>
<td>CLEAR =</td>
<td>LOOKSW =</td>
<td>COEFN =</td>
<td>AREA =</td>
<td>AREA =</td>
</tr>
<tr>
<td></td>
<td>DELMAX =</td>
<td></td>
<td>SWLOOK =</td>
<td>ICC =</td>
<td>DELT =</td>
<td>DELT =</td>
</tr>
<tr>
<td></td>
<td>STEPMX =</td>
<td></td>
<td>INLOOK =</td>
<td>BETA =</td>
<td>\</td>
<td>\</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>END =</td>
<td>\</td>
<td>\</td>
<td>\</td>
</tr>
</tbody>
</table>

\(a\)The following standard data are loaded by \text{SUBROUTINE STDATA}:  
\[\begin{align*}
\text{DTOFFJ} &= 2440 \ 000.0 \\
\text{IMODE} &= 1 \\
\text{BODYCD}(1) &= \text{(ALF5)EARTH} \\
\text{RMASS}(1) &= 1.0 \\
\text{MODOUT} &= 4 \\
\text{STEPS} &= 1.0 \\
\text{EREF} &= 1 \times 10^{-6} \\
\text{ERLIMIT} &= 3 \times 10^{-6} \\
\text{STEPMX} &= 100.0 \\
\text{LOOKSW} &= 711 \\
\text{TFILE} &= 1.0 \\
\end{align*}\]

\(b\)At input 300, setting \text{TAPE} 3 = 0 is necessary to make an ephemeris tape.
APPENDIX H

PROGRAM LISTING

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 TAPE 3 =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 1 IS MOVED TO THE INITIAL
C DATA LOCATION. THE MAIN INPUT STATION IS STATEMENT B(input).
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(6000), B(7000), C(4000),
1 1b(100), 141(1000)
C
C EQUIVALENCE
11A , C ( 11111111111), CLEAR , C ( 31111111111),
1 21D , C (21111111111), (STAGINGA ( 31111111111),
3 NCASE, A ( 6000), (STAGINGA ( 31111111111), RECALL, C ( 51111111111),
C ITR, A ( 63111111111), TAPE3 , C ( 21111111111), TABLE , C ( 191111111111)
C
C 1 CALL INPUT (99,C,TABLE)
C IF (TAPE3) 3,2,3 
2 CALL TAPE
3 NCASE = NCASE + 1
4 WRITE OUTPUT TAPE 6,12,NCASE
5 FORMAT(3(3H RECALLED INITIAL DATA FROM STAGE1,8H OF CASE4,1H,)
6 CALL INPUT (111,C,TABLE)
7 IF (SENSE) 13,14
8 WRITE OUTPUT TAPE 6,15
9 FORMAT(3H RECALLED INITIAL VIA SENSE SWA)
10 CALL EXIT
11 IF ((STAGE) 11,9,11
12 DO 11 LSTAGE=1,10
13 CONTINUE
14 LSTAGE = 10
15 CALL ORDER
16 CALL STAGE
17 CALL STAGE
18 GO TO 1
END

SUBROUTINE AERO
C SUBROUTINE AERO COMPUTES THE LIFT AND DRAG ACCELERATIONS. AS IN SUBROUT-
C LINE THRUST, THESE VECTORS ARE REFERENCED TO THE RELATIVE WIND VELOCITY.
C COEFFICIENTS OF LIFT, INDUCED DRAG, AND THRUST 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/SINALPHA, AND COD ARE ASSUMED AS FITTED QUADRATIC EQUA-
C IONS IN THE COEF ARRAY. GASPC = THE SUM SPECIFIC HEAT RATIO + STANDARD
C ACCELERATION OF GRAVITY X UNIVERSAL GAS CONSTANT. FOR EARTH, GASPC =
C 20.064881 METERS / SEC**2 / KELVIN DEGREE**2/2.
C COMMON C

C DIMENSION A(6000), B(7000), C(4000),
1 1VATM(3), P(3), XIFT(3), DRAG(3), PAR(3), X(100)
C
C EQUIVALENCE
1 11A , C ( 11111111111), (ALPHA , A ( 4911), (AREA , B ( 6111,)
2 11B , C (21111111111), (BETA , A ( 5011), (CD , A ( 16511),
3 111CDI , A ( 100311), (CL , A ( 16411), (COSLALPHA , B ( 4911),
4 CDI/SINALPHA , B ( 4911), (SINALPHA , B ( 2911), (DRAG , B ( 6911),
5 GASPC , A ( 4611), (PVAL , B ( 8411), (PAR , B ( 6011),
6 PMAGN , B ( 5011), (QV , B ( 5911), (QV , B ( 10211),
7 (SINALPHA , B ( 4611), (SINALPHA , B ( 4711), (TM , B ( 3911),
8 VATM , B ( 9711), (WATM , B ( 3911), (VQ , B ( 10011),
9 VUSQRGT , B ( 10111), (X , B ( 40111), (K , B ( 7211
C
C Q = 0.5*SINALPHA*V*VUSQRGT
C QA = QRANGE/X2
C VMACH = SQRT(VUSQRGT/1M1/GASPC
C
C COMPUTE THE X,Y,Z COMPONENTS OF LIFT.
C IF (ALPHA) 41,2
1 CL = 0.0
2 CDI=0.0
3 GO TO 4
4 CL = QUAD(VMACH**2)+SINALPHA
5 AA = QVAL+CL/PMAGN
6 AB = SINALPHA
7 GO D 3,1,3
8 XIFTIK = AA+AB+PAR(3)+CDI/SINALPHA
9 CDI=QUAD(VMACH)+31*CL
C
C COMPUTE THE X,Y,Z COMPONENTS OF DRAG.
4 CD = CDI*QUAD(VMACH)+D
5 AC = CDI*QVAL+QV
6 DO 5 K=1,3
7 DRAG(K) = AC+VATM(K)
8 RETURN
END
FUNCTION ARCTAN (Y, X)

THE FORTRAN II LIBRARY ATAN(F+ OR - Z=TAN(-THETA)) USES A SINGLE
ARGUMENT WITH ITS SIGN TO GIVE THETA IN THE FIRST (+Z) OR FOURTH
(-Z) QUADRANT.

THE ARCTAN FUNCTION MAY BE USED IF + OR - Z IS DERIVED FROM A
FRACTION SO THAT ARCTAN (Y, X) = TAN(-1 (+OR-Y/SIN(THETA)))+OR-X=
COS(THETA)). THUS THE ARCTAN (Y, X) GIVES THETA IN ITS PROPER
QUADRANT FROM -180 DEGREES TO +180 DEGREES.

IF (X) 2,1,2
1 ARCTAN=SIGNF15.57079632,Y)
GO TO 4
2 ARCTAN=ATANF1Y/X)
IFIX 3,1,4
3 ARCTAN=ARCTAN+SIGNF13.141592651Y)
4 RETURN

END

SUBROUTINE CONVT1 (V, AMC)

THIS ROUTINE COMPUTES -- (1) ANGULAR MOMENTUM, AMC(4)
(2) ANGULAR MOMENTUM SQUARED, AMC(5)
(3) X, Y, Z COMPONENTS OF ANG. MOM., AMC(IJ)
(4) VELOCITY, V(4)
(5) VELOCITY SQUARED, V(5)

COMMON C

DIMENSION A(4000), B(4000), C(4000),
1 AMC(3), V(5), R8(3), IND(3)
2
C EQUIVALENCE
3
1 11 A, C ( 111), ( 111), ( 1111), (IND), A ( 60),
2R8, B ( 193)
C
DO 1 J1=1,3
1 J2=IND(1J)
J3=IND(2J)
ARMC(J3) = R0(J3) + V(J3) + R8(J1) + V(J1)
AMC(5) = AMC(I))**2 + AMC(2) + 2 + AMC(3)**2
AMC(4) = SQRTF(AMC(5))
V(5) = V(4)**2 + 2 + 2 + V(3)**2
V(4) = SQRTF(V(5))
RETURN
END

SUBROUTINE CONVT2

THIS ROUTINE CONVERTS RECTANGULAR COORDINATES INTO ORBIT ELEMENTS.
RECTANGULAR COORDINATES- POSITION COMPONENTS, X, AND VELOCITY COMPONENTS, VX.
THE ORBIT ELEMENTS ARE IN THE ORBELS ARRAY-
(1) ECCENTRICITY
(2) ARGUMENT OF PERICENTER
(3) MEAN ANOMALY
(4) INCLINATION
(5) LONGITUDE OF ASCENDING NODE
(6) SEMILATUS RECTUM

COMMON C

DIMENSION A(4000), B(4000), C(4000),
1 AMC(3), ORBELS(6), R8(3)
2
C EQUIVALENCE
3
11 A, C ( 111), ( 111), ( 1111), (IND), A ( 60),
2AMC, R8, B, C ( 1111), ( 1111), ( 1111), ( 1111), ( 511),
3EPAR, R8, B ( 261), ( 463), ( 163), ( 363), ( 163),
4R, V, B ( 1023), ( 163), B ( 163), ( 163), ( 163),
5TRU, V, B ( 400), ( 951), ( 951), ( 951), ( 951),
6VX, V, B ( 923), ( 923), ( 923), ( 923), ( 923),
C
ORBELS(6) = AMSQR/GK2M
P=SQRTF(IN(1)+VX**2+R8(1)+**2*R8(3)**2)
TRU=ARCTANAM/GK2M*AMC(1)/VX+P+V*X**2+R8(3)+V*X, ORBELS(6)-R)
1 IFAMC(1) 2,1,2
1 ORBELS(3) = 0.
GO TO 3
2 ORBELS(3) = ARCTANAMC(1)*AMC(2)
3 ORBELS(4) = ARCTANAMC(AMC(1)**2 + AMC(2)**2 AMC(3))
SNODE = SINF(ORBELS(3))
COSD = COSF(ORBELS(3))
AA = R8(1) + SNODE + R8(2)**2 + SNODE
AB = R8(3) + SNODE + COSF(ORBELS(4)) + R8(2)**2 + R8(1)**2 + SNODE
ORBELS(2) = ARCTANAB, AA + TRU
ORBELS(1) = SQRTFASFF + ORBELS(6) + V/SQR/GK2M2/R)**3
EPONE = SQRTF(EZM)**2 + 2
EPAS = SQRTFASFF(EZM)**2
SNTRU = SINFTRU
COSTRF = COSFTRU
EPAS = SQRTFASFF ORBELS(1)**2 + TRU + ORBELS(1)**2 + COSTRU0 + EPAS
1 IFEPAS(1) 9,6,6
5 ORBELS(5) = LGP(EPONE+EPAS)**2(EPONE+EPAS) + ETHETA
GO TO 7
6 ORBELS(1)**2 + 2, ARCTANEPAS, EPONE - ETHETA
7 RETURN
END

48
SUBROUTINE ERRORZ

THIS SUBROUTINE COMPUTES THE RELATIVE ERRORS BETWEEN THE R-K AND LOW-ORDER INTEGRATION SCHEMES. IT ALSO COMPUTES THE ERROR COEFFICIENT, A, AND SAVES THE ERROR DATA WHEN EREF HAS A - SIGN. THE BRANCH ON IMODE DETERMINES WHICH SET OF NORMALIZING FACTORS ARE TO BE USED.

COMMON

DIMENSION A(1600), B(1000), C(4000),
1 RELER 11, XPRIM(200), XINC (100)

EQUIVALENCE

1 A (111), B (101), C (111),
2 B (111), DELT, A (111), E2 (101),
3 RELER A (111), IMODE, A (111), INERR, B (511),
4 R, B (101), STEPNO, A (411), STEPDO, A (421),
5 V, T (951), XINC, B (601), XPRIM, C (711)

EQUIVALENCE (RELER, XINC)

RELERR1 = XINC1/1XPRIM1 + 1.0/10.
RELERR12 = XINC1/2/XPRIM1/210.

DO 10 J=1,4
RELERR131 = XINC131/1XPRIM131 + 1.0/10.
RELERR132 = XINC132/1XPRIM132 + 1.0/10.

SELECT MAXIMUM ERROR, COMPUTE ERROR COEFFICIENT, POSSIBLY SAVE ERROR DATA.

END

SUBROUTINE EQUATE

THIS SUBROUTINE IS CALLED FROM NODY TO EVALUATE THE DERIVATIVES OF THE VARIABLES OF INTEGRATION. EITHER RECTANGULAR COORDINATES OR ORBIT ELEMENTS MAY BE USED AS THE VARIABLES OF INTEGRATION, BUT IN THE CASE OF THE LATER, THE CORRESPONDING RECTANGULAR COORDINATES MUST FIRST BE FOUND.


COMMON

DIMENSION A(1600), B(1000), C(4000),
1 XPRIM (100,2), V (3), QX (3),
2 RB (3), MFRMS (8), X (100),
3 XPRIM (100,2), FORC (3), X (100),
4 DRAG (3), OBLAT (3), COMPX (3),
5 XDOT (100)

X ORBIT ELEMENTS

1 TIME
2 MASS
3 ECCENTRICITY
4 ARGUMENT OF PERICENTER
5 ARGUMENT OF ASC. NODE
6 INCLINATION
7 MEAN ANOMALY
8 SEMILATUS RECTUM

COMMON

DIMENSION A(1600), B(1000), C(4000),
1 XPRIM (100,2), V (3), QX (3),
2 RB (3), MFRMS (8), X (100),
3 XPRIM (100,2), FORC (3), X (100),
4 DRAG (3), OBLAT (3), COMPX (3),
5 XDOT (100)
EQUIVALENCE 11A, C     ( 1,1,1)IAMSUD, B     ( 911),(ASYMP, A     ( 71),
21A, C     ( 1,1,1)IENAME, B     ( 551),
31CIRCUM, B     ( 821)(COMP, B     ( 631),(CUC5TA, A     ( 321),
41CSTRO, B     ( 531)(COV, B     ( 5711),(URAG, B     ( 691),
51EMUNE, B     ( 2B1),(EPAR, B     ( 261),(ETUL, A     ( 301),
61EXTMODE, B     ( 271),(FLOW, B     ( 911),(FORME, A     ( 611),
71GRTM, B     ( 381),(UMB, B     ( 371),(IMUSE, A     ( 111),
81KSB, B     ( 191),(KMODYS, B     ( 421),(INEMRE, B     ( 181),
91NVAENT, B     ( 241),(DQALT, A     ( 4011),(ORLAT, B
EQUIVALENCE 11FISH, B     ( 331),(PUHDO, B     ( 3911),(CQ, B     ( 7B1),
21RADIAL, B     ( 811),(ERATUS, B     ( 2311),(H B     ( 1931),
31B     ( 1021),(RSMQ, B     ( 5511),(SINCE, B     ( 541),
41SINTUS, B     ( 9211),(SINV, B     ( 5611),(SPU, A     ( 461),
51STBATL, B     ( 2011),(TUFFT, A     ( 2411),(TSRSL, B     ( 81),
61TEST, A     ( 5431),(C     ( 5911),(TV, B     ( 451),
71VSURD, B     ( 9611),(VA, B     ( 9211),(KGET, B     ( 501),
81XIF, B     ( 7211),(EXPRM, C     ( 7111),(EXPRM, C     ( 811),
91X     ( 4011),(JORMAL, B     ( 8311),(IZN, B     ( 431)

C TAHAT=XI1J/3P+TOFFT
IMODU=IMODE
I JU TO 12,6,161,IMODE
C
C STATEMENTS 1 TO 16 FIND THE RECTANGULAR POSITION AND VELOCITY FROM ORBIT
ELEMENTS AND TRUE ANOMALY. THE TRUE ANOMALY IS FOUND FROM ITERATIVE
C SOLUTION OF KEPLER'S EQUATION.
2 EJ = XI3) + 2 2EM = 1.0 +
EMON = X13J.1 EPAR=SQRTFABSF(E2M))
VCLCL=GM/4/1=QTFX1B1)
C
C COMPUTE SIN AND COSINE OF TRUE ANOMALY.
C PART A, E=1
3 IF (EMON) 10,4,5
4 SINTU = 0.
COSTU = 1.
GO TO 14
C PART a, E IS GREATER THAN 1
5 DO 7 J=1,161
DELU = XI7) -UXI311)*SINFU) ECOU = XJ9)*COSF(U)
DELU = DELU/((1.0 -ECOSU) +ECOSU**3))
U = U/DELU
6 IF IABSFUDELU)=COSTU) 9,9,7
7 CONTINUE
ASYMFI = 1.0
IF (EMON) 823,8
B CALL EPHMR
GO TO 23
9 COSU = C0SFU)
DEMI = 1.XI3)*COSU COSTRU = (ECOSU-XJ311)/DEMI
SINTU = EPAR*SINFU)/DEMI
GO TO 14
C PART C, E IS LESS THAN 1
10 DO 12 J=1,19
DELU = X171)-UXI31)*SINFU) ECOU = XJ9)*COSF(U)
DELU = DELU/((1.0 -ECOSU +O.1*ECOSU**3))
U = U/DELU
11 IF IABSFUDELU)=COSTU) 13,13,12
12 CONTINUE
WRITE OUTPUT TAPE 6,55,VO,DELU
CALL EEP
C 13 COSU = C0SFU)
DEMI = 1-XI3)*COSU COSTRU = (ECOSU-XJ311)/DEMI
SINTU = EPAR*SINFU)/DEMI
14 PDVR = 1.XI3)*GOSTU
C
C COMPUTE POSITION AND VELOCITY FROM ORBIT ELEMENTS AND TRUE ANOMALY.
C ALSO, CLEAR THE PERTURBATING ACCELERATIONS.
15 SOME = SINF(X14))
COMEGA = COSF(X14))
SNODEU = SINF(X15))
CNODEU = COSF(X15))
SINC3U = SINF(X16))
CINC3U = COSF(X16))
SINV=3INTU+COMEGA+GOSTU+SOMEU
COSV=3CSTTRU+COMEGA-SINTU+SOMEU
AK=COSV*CNODEU+SNODEU*SINC3U
BL=COSV*SNODEU+SNODEU*CINC3U
CL=SNODEU*COSV+SNODEU*CINC3U
DL=SNODEU*SNODEU+SNODEU*CINC3U
EL = XJ3)*SUMAS+2N
FL = XJ3)*LUMAS+CDV
AS=E+CNODEU*F+SNODEU*CINC3U
B12=CNODEU*CINC3U-EL)*J06E
R = XJ0)*PDVR
ASPO = X11)
SINV=3INTU+INCL
RB11) = X11)
RB(2) = X11)
KR(3) = X11)
VK(11) = VEICL1+4S
VK(12) = VEICL1+82
VK(13) = VEICL1+FI*SIGCL
GO TO 18
TEST FOR PRESENCE OF PERTURBING BODIES.
IF (IMODS) 20,21,20
20 CALL EPHEMRS
21 IF (XABSFLIMODE) 21 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 (ETOTL-ABSFLIMODE) 26 23,23
23 IF (IMODE) 5,4,24
24 IMODE=-3
25 IF (INSTM) 25,54,25
26 IF (IMODE) 26 27,27
27 CALL TESTR
C
C TEST FOR OBLETNESS PERTURBATION COMPUTATION.
28 IF (OBLATENESS) 28 30,29,30
29 CALL OBLATE
C
C TEST FOR PRESENCE OF THRUST.
30 XDOTT(2) = -FLOW IF (R-RATMOS) 31,31,32
31 CALL ICAD
GO TO 33
32 PRESS=0.
33 IF (PUSH) 34,34,37
36 IF (40 TO NOONE) GO TO 38
37 CALL THRUST
ASSIGN 41 TO NOONE
38 IF (PRESS) 39,42,39
39 GO TO NOONE, 140,41
40 CALL AERO
C
C SUM COMPONENTS OF THE PERTURBING ACCELERATION.
42 DO 43 J=1,3
43 COMP(J) = -Q(J+2)+OBALT(J)+FORCE(J)+DRA(J)
44 GO TO 47,47,47
45 AA = GZM/R/RSQRO
GO 46 46,46,46
46 XDOTT(X+2) = X(J+2)
47 GO TO 54
C
C COMPUTE THE DERIVATIVES OF THE ORBIT ELEMENTS. (AFTER RESOLVING
C PERTURBING ACCELERATION INTO CIRCUMFERENTIAL, RADIAL, NORMAL COMPONENTS)
47 CIRCUM=COMP(3)+COSV*INCL-COMPA(1)+BI-COMPA(2)*sin
RADIAL=COMPA(1)+AR+COMPA(2)+C1+COMPA(3)+SINV
ZORMAL=COMPA(1)+SINQ*SINCL-COMPA(1)+CNGDE*INCL+COMPA(3)*Cincl
ZVClCIRC=EM1*EPAR/X(8)
ROVPl = 1./PDVR + 1.
RUVA = EM2/RDVR
XDOTT(8) = 2.*VClCIRC*CIRC
IF (X(3)) 48,48,48
48 CSGROD = CIRCUM*CIRCUM
RASQROD = RAUJAL+RADIAL
DEM(1) = (4.45QROD+RASQROD)*VClCIRC
C
C TEST FOR IN-PLANE PERTURBATION.
IF (DEM(1)) 55 57,56,57
56 XDOTT(3) = 0.
XDOTT(4) = 0.
XDOTT(7) = 0.
GO TO 50
57 VDVZ=VClCIRC/42.
XDOTT(3) = SQRTF(42.)*CSQROD+RASQROD)*VClCIRC
XDOTT(4) = VDVZ(+12.)*CSQROD+RASQROD/DEM(1)+RADIAL
XDOTT(7) = -VDVZ(+12.)*CSQROD+RASQROD/DEM(1)+RADIAL
GO TO 50
49 XDOTT(3) = (1+SINTRU+RADIAL+PDVR-ROVA)/X(3)+CIRCUM*VClCIRC
XDOTT(4) = (SINTRU/X(3)+DVPPI=CIRCUM-COSTRU RADIAL/X(3)+VClCIRC
XDOTT(7) = (2.*VClCIRC)*RADIAL-(SINTRU/X(3)+
INVPI=CIRCUM+J)
50 IF (SINCL) 51,92,91
51 XDOTT(5) = SIN/SINCL-JORMAL/VClCIRC/PDVR
GO TO 53
52 XDOTT(5) = 0.
53 XDOTT(6) = CUSV*KORMAL/3PDVR/VClCIRC
54 RETURN
55 FORMAT(1H10KELPERS EQUATION CONVERGENCE FAILURE, U=65.81.76, DELU=
165.83)
END

SUBROUTINE EPHMRS

This subroutine is called to compute the positions of the perturbing bodies relative to the vehicle. Occasionally, this routine is called for the purpose of translating the origin in which case (TRANSFER=1) the relative velocities are also calculated. If a body's position is to be computed from an elliptic approximation, subroutine ELIPSE is called. Otherwise, the position will be calculated in EPHMRS from the precision tape ephemeris. The do loop encompasses almost the entire EPHMRS subroutine and, in effect, ELIPSE too.

COMMON C

DIMENSION A(4000), B(7001), C(4000),
1 X(3,10), Y(3,10), Z(3,10), R(3,10), T(3,10),
2 NEPHMRS, TDATA, X1(3,3), TDEL(7), BMAS5(10), VEFM(3,1), DATA(I)
3 , TDAT(11,7)

EQUIVALENCE

11 A , C ( 111) , XAU, A ( 291) , EB , C ( 1111)
2 IMASS, B ( 1371) , (TDOFFJ, A ( 231) , NEPHMRS, B ( 1301)
3 I BODY , B ( 1771) , (MODYS, B ( 421) , NEPHMRS, B ( 1651)
4 40# , B ( 781) , 40# B ( 1021) , BE , B ( 1351)
5 SSSQRM , B ( 351) , SSSP, A ( 441) , (TAFLY , B ( 201)
6 XDATA , B ( 2651) , (TDEL B ( 1701) , (TIM , B ( 1351)
7 7TTRSF, B ( 811) , (VEFM B ( 2411) , (XPM , B ( 2711)

EQUIVALENCE (1DF, 1FB), (TDAT1, TDAT1)

PART 2. SET INDEX. FIND POSITION IF ELLIPSE IS USED (NEPHMRS = 20 OR UP).
1 DO 19 J=1, MODYS
2 IB = J+1
3 IF (TRANSFER) 12, 12, 17

PART 3. TAPE EPHMERIS IS TO BE USED. FIND DIFFERENCE (DT) BETWEEN
4 CURRENT PROBLEM TIME (TDOFF + TAFLY + MIDPOINT TIME) OF CURRENTLY
5 STORED TAPE DATA. THEN SEE IF CURRENT DATA IS OKAY. TTOL = TIME INTERVAL
6 ON EITHER SIDE OF TIM FOR WHICH CURRENT DATA IS GOOD.
7 TTOL = T(TIM) - (TDOFF) IF (TRANSFER) 10, 10, 3

PART 4A. CURRENT DATA NOT OKAY. READ IN NEXT DATA SET. IF DT IS -,
8 BACK UP THE TAPE 2 RECORDS BEFORE CURRENT DATA.
9 IF (NEPHMRS) = 20 OR UP.
10 CONTINUE

PART 4B. IF THIS DATA IS FOR A BODY IN THE NAME LIST, STORE IT.
11 IF (TRANSFER) 12, 12, 15
12 READ TAPE 3, (DATA(1), 1=1,21)

PART 5. CURRENT DATA IS OKAY. GET POSITION FROM THE POLYNOMIAL
13 P = A + B X + C X**2 + D X**3 + E X**4 + F X**5.
14 DO 10 K=1, 3
15 XP(K, JBI) = TDAT(A(1), K, JB)
16 DO 10 K=1, 3
17 XP(K, JBI) = XP(K, JBI) + DT*TDAT(A(K, K, JB)
18 CONTINUE

PART 6. COMPUTE DISTANCE FROM REFERENCE AND FROM ROCKET.
19 DO 12 K=1, 3
20 XP(K, JB1) = XP(K, JB1) - XP(K, JB1)*SIGNFIAU*FIB
21 CONTINUE

PART 7. COMPUTE PERTURBING ACCELERATIONS (QX).
22 IF 4194304==2**22 IS REMOVED
23 TO PREVENT OVERFLOW. 0.40**2**11 AND 850093492==2**33 RESTORE THE SCALE.
24 PRSQR = (R011, JB1**2 + R2(S, JB1)**2 + R4(3, JB1)**2 + 21494304)
25 MRELS = SQRTF(FPRSQR)
26 RQSQ = (XP1, JBI)**2 + X2(JB1)**2 + X3(JB1)**2 + 21494304
27 RCUBE = RQSQ + SQRTFSQ RQSQ
28 PRSRC = PDRSQ + MRELS
29 RJB1 = RRLIL*2048.
30 CONTINUE

PART 8. COMPUTE VELOCITY FROM V = B + 2(4X + 30X**2 + 4X**3 + 5X**4)
31 AND FROM REFERENCE BODY VELOCITY (VEFM(1)).
32 DO 15 K=1, 3
33 VEPMK(JBI) = VEPMK(JBI) + VB(K, JB1)*CRUHE1/7
34 CONTINUE

PART 9. RETURN
35 CONTINUE
36 RETURN
37 END
SUBROUTINE EXTRA
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 A1600, B1700, C14000, D14000
C EQUIVALENCE (A, C, B, C, D)
C SIGNAL = 0.
C QMAX = 0.
RETURN
END

SUBROUTINE EXTRA
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 (JO1)
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). THE ELLIPSE DATA IS READ FROM
C INPUT CARDS AND ORGANIZED IN SUBROUTINE ORDER. TPO IS TIME SINCE PERIHELION
C PASSAGE, ZM IS MEAN ANOMALY, U IS ECCENTRIC ANOMALY.
C COMMON C
C DIMENSION A1600, B1700, C14000, D14000
C 1 XP13, B1, VEP3(3, B), TDATA(1211)
C EQUIVALENCE (A, C, B, C, D)
C SIGNAL = 0.
C JO1 = 0.
C RETURN
END

SUBROUTINE ELIPSE (JO1)
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). THE ELLIPSE DATA IS READ FROM
C INPUT CARDS AND ORGANIZED IN SUBROUTINE ORDER. TPO IS TIME SINCE PERIHELION
C PASSAGE, ZM IS MEAN ANOMALY, U IS ECCENTRIC ANOMALY.
C COMMON C
C DIMENSION A1600, B1700, C14000, D14000
C 1 XP13, B1, VEP3(3, B), TDATA(1211)
C EQUIVALENCE (A, C, B, C, D)
C SIGNAL = 0.
C JO1 = 0.
C 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 SIGNIFICANT PART.
ENTRY EXADD
COMMON -206
Q1 COMMON 1
Q2 COMMON 1
TEMPI COMMON 1
TEMP2 COMMON 1
HTR
BCI 1,EXADD
SXDI =-4.1
SXDI =-4.2
EXADD SXDI =-4.4
C1 = 1.4
FAD* = 3.4
STO Q1
FAD* = 2.4
STO Q2
FAD* = 2.4
STO Q1
STO TEMP1
CLA Q1
FAD Q2
STO TEMP2
FAD TEMP2
FAD TEMP1
STO Q1
FSB TEMP2
STO* 4.4
TRA 4.4
END

SUBROUTINE ICAD
SUBROUTINE ICAD DETERMINES THE ATMOSPHERIC TEMPERATURE, PRESSURE, AND
DENSITY AS A FUNCTION OF ALTITUDE ABOVE THE EARTH IN ACCORDANCE WITH
THE 1962 U.S. STANDARD ATMOSPHERE (ICAO TO 20 KM.). A SHORT FAP
PROGRAM FOLLOWS ICAD WHICH PROVIDES A MEANS OF LOADING DATA INTO MACHINE.
IT MUST BE LOADED DIRECTLY AFTER ICAD. IF THE LENGTH OF ICAD
IS CHANGED, 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 EARTH'S 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 TM IS TM AT TABLE H.
C REP P IS THE PRESSURE IN MILLIBARS AT TABLE H.
C TM IS THE TEMPERATURE TIMES MOLECULAR WEIGHT / ACTUAL
C MOLECULAR WEIGHT. DEGREES KELVIN.
C PRESS IS PRESSURE IN MILLIBARS.
C DENSITY IS DENSITY IN KILOGRAMS PER CUBIC METER.
C HEIGHT IS EITHER GEOPOTENTIAL ALTITUDE OR GEOMETRIC ALTITUDE IN METERS.
C
COMMON C
C
C DIMENSION A(600), B(1000), C(4000),
1 TABLEH(231), TM(231), REP(231), ALM(231), RB(31)
C
C EQUIVALENCE
1 14A = C ( 111), (ALT , A ( 411), (B , C ( 1111),
1 DENSITY , A ( 291), (OBLATN, A ( 401),
1 PRESS , B ( 331), (R , B ( 1021), (RB , B ( 1931),
1 WRE , A ( 291), (TABLT, B ( 201), (TM , B ( 341),
1 RESQRO , B ( 77)
1 EQUIVALENCE (TABLEH(241), TM(1), TABLEH(47), ALM(1), TABLEH(70), REP)
C
C IF (OBLATN) 102,101,102
101 ALT = R - RE
GO TO 103
102 ALT = R-6356783.2/B/SQRT(1.993305783+.006693421685*(R/313)**2)
103 IF (ALT=9000.) 105,104
104 HEIGHT = ALT
GO TO 106
105 HEIGHT = ALT/(1.0+ALT/635766.)
106 K=K
C
C FIND THE HEIGHT IN A TABLE OF BASE DATA. DATA ARE
C ARRANGED IN DECREASING ALT WITH 21 REGIONS. ABOVE THAT, PRESSURE AND
C DENSITY ARE SET = 0. TEMPERATURE IS SET TO 3000.
1 IF (K=22) 2,6
2 IF (HEIGHT-TABLEH(K)) 5,3,3
3 K = K+1
GO TO 1
4 K = K-1
5 IF (K) 7,3,6
6 HINC = HEIGHT - TABLEH(K)
IF (HINC) 4,8,8
7 K = 1
8 IF (ALM(K)) 9,100,9
C
C CONTROL COMES HERE FOR NONISOTHERMAL LAYERS
9 TM = TMRIK(1) + ALMKI*H INC
IF ALT = 0.0 TO 108
107 PRESS = REFPKI*(TMRIK(1)/TM + 0.0341631947/ALMKI)
GO TO 10
108 IF (K-1) = 109,110,109
109 KC = K
C1 = REPH(1) + ALMKI*H INC
C2 = TMRIK(1)/ALMKI
C3 = 1./C1*C2
C4 = -0.0341631947*REPH(1)*C3
110 PRESS = REFPKI*EXPFC4*H INC/CI+C2+HEIGHT
10 DENSITY = PRESS/TM/2.07053072
GO TO 13
C CONTROL COMES HERE FOR ISOTHERMAL LAYERS
100 IF (K-1) = 11,12,12
11 TM = TMRIK(1)
PRESS = REFPKI*EXPFC4*H INC/CI+C2+HEIGHT
GO TO 10
C CONTROL COMES HERE FOR EXTREME ALTITUDES
12 PRESS = 0.0
DENSITY = 0.0
TM = 3000.
13 RETURN
END

REML THIS IS THE FAP PROGRAM WHICH LOADS ICAO DATA INTO MACHINE.
REML THE 256 IN ORG 256 WAS FOUND BY SUBTRACTING 22 FROM THE DEC LOCATION
REML OF REF PI FROM FAP LISTING OF ICAO. THIS WAS FOUND TO BE 2781.
REML THUS, 278-22 = 256. DISCARD THE FIRST TWO BINARY CARDS AFTER ASSEMB
REML AND PLACE REMAINING CARDS IMMEDIATELY BEHIND ICAD BINARY DECK.
REML
REML REM A1 IS REF P(23)
REML REM A2 IS ALM(23)
REML REM A3 IS TMRIK(23)
REML REM A4 IS TABLE H(23)
REML
ORG 256
A1 DEC 0.1,1.1E8E-9,3.6502E-9,1.0957E-8,4.0304E-8,1.8838E-7
DEC 6.9604E-7,1.6212E-7,3.9292E-7,6.33943E-7,6.50317E-7,6.29218E-7
DEC 2.6664E-7,1.0076E-7,3.0979E-7,2.81077E-8,1.82099E-8,5.90008E-8
A2 DEC 1.10090E-8,6.80014E-7,5.4787E-7,2.20320E-7,1.01325E-7
DEC 0.005,0.03,0.0,0.006,-0.002,0,0.0028,0.001,0,0.006
A3 DEC 0.27006,6.65290,6.24020,6.21605,6.18305,6.15906,6.13506,6.10569,6.0569
DEC 1.21069,1.11105,6.98065,6.36580,6.23065,6.21605,6.18065,6.14065,6.10569
DEC 252.65,270.65,270.65,228.65,216.65,216.65,218.15
A4 DEC 1.830.75,4.65,4.65,4.65,3.5,3.5,4.65,1.75,5,1.665,1.665,1.665
DEC 1.155,1E5,9E5,79000,61000,52000,47000,32000,20000
DEC 0.000,0.0
END

SUBROUTINE NBODY
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,INC,ECC,ETC. ARRAYS ARE AS FOLLOW.
C
C X ORBIT ELEMENTS RECTANGULAR COORDINATES
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
C COMMON C
C DIMENSION A(4000), B(7000), C(4000),
C 1 XPRIM (100,2), XPRIM (100,2), XDOTPM (100,2),
C 2 X (100), XINC (100), XINC (100),
C 3 XDOT (100), RB (33), XX (100),
C 4 AMC (3), AX (3), AX (3),
C 5 XMOML (4), VY (3), VY (3),
EQUIVALENCE  
IA ,C ( 111),(A1 ,B ( 101),(A2 ,B ( 11))  
1ACOEFL.B ( 121),(ACOEFL.B ( 131),(ACOEFL.B ( 141)  
1AK ,B ( 511),(AMC .B ( 871),(AMGdob.B ( 911)  
1AI ,B ( 901),(ASYMPT.A ( 711),(ASEM .A ( 551)  
1B ,C ( 1111111),(GONST.A ( 3211),(DELN .A ( 111)  
1DECM .B ( 3911),(E2 ,B ( 1811),(EMODE .B ( 281)  
1ETRILMF.A ( 1411),(ETO .A ( 3011),(EMODE .B ( 271)  
1FK4 ,B ( 3611),(H2 ,B ( 1511),(IMODE .A ( 111)  
91(INDERR.B ( 5111),(KSUB .B ( 1911),(IMODUS.B ( 421)  
EQUIVALENCE  
INEQ ,A ( 2111),(NSTART.B ( 2411),(FODDEL.B ( 91)  
1QMAX.B ( 4411),(RATIO.B ( 5811),(RBD.B ( 1931)  
1REV5.A ( 4811),(IR .B ( 10211),(STEPK .A ( 161)  
41STEPG0 ,A ( 4111),(STEPG .A ( 4211),(TRANS.B ( 81)  
1TRU .B ( 4011),(ITEST .A ( 5411),(TVGT .A ( 961)  
61VX ,B ( 9211),(XODT .B ( 5011),(XINC .B ( 6011)  
71XPRIM .C ( T1111),(XPRIM.C ( 91111),(XHOME.B ( 11011)  
81X ,B ( 4011),(ERLOG .B ( 1711),(EREF .A ( 131)  
91Q ,B ( 5911),(OUTPUT.B ( 3991)  
C  
PART 1. SET UP THE STARTING SEQUENCE FOR ERROR CONTROL AND DELAY CHECKING  
THE ERROR UNTIL TWO STEPS ARE COMPLETED. THE ASSIGNED GO TO NSTART AND IBEGIN CONTROL STARTING.  
NEQ = NEQ  
DO 2 J=1,NEW  
XPRIM(J,2) = XPRIM(J,1)  
XPRIM(J,2) = XPRIM(J,1)  
2 X(J) = XPRIM(J,1)  
NSTART = 0  
TRANS = 0  
DEL = DELT  
DEL = DELT/2.  
220 CALL UQUETE  
IF (OUTP0T) = 222,221,222  
221 CALL OUTPUT  
222 DO 3 J=1,3  
XHOME(J) = VX(J)  
3 XHOME(J) = RB(J)  
C CHANGE INTEGRATION VARIABLES IF IMODE IS -.  
IF (IMODE) = 4,5,5  
4 CALL TESTTR  
GO TO 1  
5 CALL TESTRA  
IF (TRANS) = 1,201,1  
205 ASSIGN 21 TU NSTART  
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART 7A.  
IF (RB(J)) = 7,8,8  
6 IF(VX(J)) = 7,8,8  
7 ASSIGN 37 TU NREV1  
ASSIGN 35 TU NREV2  
GO TO 9  
8 ASSIGN 33 TU NREV1  
ASSIGN 37 TU NREV2  
9 DO 10 J=1,NEQ  
XODT(J,1) = XODT(J)  
XINC(J) = 0.  
10 CONTINUE  
11 KSUB + 1  
ASSIGN 16 TO N  
C  
PART 2. RUNGE-KUTTA SUBINTERVAL SCHEME. EQUATE PRODUCES THE NECESSARY  
DERIVATIVES XDOT(J).  
12 DO 13 J=1,NEQ  
X(J) = XDOT(J) + DELT  
XINC(J) = XINC(J) + A(KSUB)*XK(J)  
13 X(J) = XPRIM(J,2) + AKKSUB)*XK(J)  
14 CALL UQUETE  
15 GO TO N,16,17,18,20)  
C  
PART 3. SUBINTERVALS 2, 3, AND 4. TO STATEMENT 19 FINISH A  
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) + A(KSUB)*XDOT(J) + DELT  
19 CONTINUE  
C  
PART 4. BEGIN A NEW RUNGE-KUTTA STEP. THIS ALSO GIVES DERIVATIVES  
FOR THE LOWER ORDER INTEGRATION CHECK.  
ASSIGN 20 TO N  
GO TO 14  
20 GO TO NSTART(1,27,23,21)  
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  
XODT(J,J)=XINC(J)  
XINC(J)=0.  
XODT(J,J)= XODT(J)  
22 CONTINUE  
GO TO 11  
C
PART 50. MAX ERROR TEST—STARTING ONLY—CHECK THE MAX ERROR AND EITHER ENTER RUNNING MODE OR REPEAT START WITH SMALLER STEP.

DO 24 J=2,NEQ
24 XINC(J)=XINC(J)+OLDINC(J)+3.-|XDOTPM(J,1)+XDOTPM(J,2)*4.|
24 XDOTDL(J)=DDEL

260 CALL ERRORZ
25 IF (E2-ERLIMT) 26,26,56
26 ASSIGN 27 TO NSTRT
ASSIGN 11 TO IBEGIN
A1 = A2
GO TO 31

C PART 6. RUNNING PHASE PROGRAM.
C PART 2A. CHECK THE INTEGRATION BY INTEGRATING OVER THE LAST C RUNGE KUTTA STEP BUT USE DOTS FOR LAST TWO INTERVALS, OLDEL C AND DELT RESPECTIVELY. STATEMENT 28 IS THE LOWER INTEGRATION C HUNG RUNGE-KUTTA INCREMENTS. ERRORZ COMPUTES THE MAXIMUM RELATIVE C ERROR AND STATEMENT 29 TESTS THIS ERROR AGAINST THE LIMIT VALUE.

DO 27 J=Z,NEQ
27 XINCIJI = XINCIJ+l*
27 XOOTPMJl=|XOOTPM(J,1)+XOOTPM(J,2)*4.|
27 XOOTDLJ=HFACT
28 XINC(J) = ACOEF1*XOOTP(J,1) + ACOEF2*XOOTP(J,2) + ACOEF3*XINC(J)
14ACOEF3=DOT(J)

280 CALL ERRORZ
29 IF (E2-ERLIMT) 30,30,57

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.

DO 30 J=1,Z
30 H2 = DELT
31 HMAX = MAXF(J,OMAX)
31 IF (R(BIZ2) < 3.34,34
32 GO TO NREV2; (37,35)
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 REV = REV + 1.
37 IF (XABS(F)) < 11,42,42

C PART 7B. IN ORBIT ELEMENTS. ADJUST ARGUMENT OF PERICENTER AND MEAN ANOMALY TO = OR + PI TO MAINTAIN ACCURACY IN SIN-COS ROUTINES.

38 IF (EMONE) < 19,42,42
39 DO 41 J=1,3
41 ADJ2=IPF(XPRIM(J,2)+6.28318532+SIGN(F)*5.234);)
40 ADJ3 = -ADJ2+0.28172
40 CALL EXAO(I(XPRIM(J,2),XPRIM(J,2),ADJ3)
43 CONTINUE

C PART 7C. ADVANCE THE REMAINING PARAMETERS, FIND NEW STEP SIZE. AND TEST FOR AN ORIGIN TRANSLATION.

42 DO 43 K=1,3
43 KWHOLE(K)=INT(K)
44 WHOLE(K)=R(K)
44 DO 44 J=1,NEQ
44 XDOTPM(J,1) = XDOTPM(J,1)
44 XDOTPM(J,2) = XDOTPM(J,2)
44 XPRIM(J,1) = XPRIM(J,2)
44 XPRIM(J,2) = XPRIM(J,1)
44 XINC(J) = 0.
44 CONTINUE
45 OLDEL = DELT
45 CALL STEP
45 IF (COND) 47,47,47
45 IF (XABS(F)) < 11,42,42
47 IF (XABS(F)) < 11,42,42

C PART 7D. INF IN TEMPORARY RECTANGULAR COORDINATES, TEST FOR RETURN TO ORBIT ELEMENTS. FIRST, E IS FOUND. IF TIME HAS NOT ADVANCED SUFFICIENTLY, INTEGRATION CONTINUES IN RECTANGULAR VARIABLES (STATE, AB). STATEMENT 49 DETERMINES IF KEPLER'S EQUATION CAUSED IMO = 3. IF NOT, AN E CLOSE TO 1 CHECK IS MADE IN STATEMENT 50. IF IT DID, RECTANGULAR VARIABLES WILL BE USED IF THE LIMIT IS TOO SMALL (STATEMENT 52). OR IF E IS 5 OR GREATER (STATEMENT 53) OR IF THE PATH LIES CLOSE TO AN ASYMPTOTE (STATEMENT 55).

48 CALL CONVI(VX,AMC)
48 EMONE=EMONE*4.9
48 IF (XPRIM(J) < 1,40,40
49 CALL TESTTR
49 IF (XABS(F)) < 11,42,42
C PART 8. COMES HERE WHEN ERROR TEST FAILED-BoTH STARTING AND RUN.
C RETRIEVE DLL POINT AND RECOMPUTE WITH SMALLER INTERVAL.
C IF TWO CONSECUTIVE TRIES FAIL (STATEMENT 91) THE STARTING SEQUENCE OCCURS.
56 ASSIGN 1 TO IBEGIN
57 DO 58 J=1,NEQ
XPRI(M,J,2) = XPRI(M,J,1)
XPRI(M,J,1) = XPRI(M,J,2)
KODE(J) = KODE(J) + 1
59 CONTINUE
58 CONTINUE
STEPNO = STEPNO + 1.
K2 = DELT
DELT = SIGN(FLXP1(ERLOG-42)/5.1,DEL)
A = AL
59 IF (FAIL-STEPGO) 60,61,60
60 FAIL = STEPGO
GO TO IBEGIN, (111,1)
61 ASSIGN 1 TO IBEGIN
IF (STEPNO-STEPGO) = STEPMK, 62,62,45
62 GO TO IBEGIN, (111,1)
C PART 10. PRINT OUT THE ERROR INFO. IF EREF HAS A - SIGN. THEN RETURN.
67 IF (EREFL) 68,69,126
68 WRITE OUTPUT TAPE 6,70
REWIND =
GO 69 IN=1,INDERR
READ TAPE 4, BEX
69 WRITE OUTPUT TAPE 6,71,BEX
REWIND =
INDERR = 0
70 FORMAT(IN, 12X, 4HTIME, 6X, 4HDEL1, 7X, 2HMA2, 8X, 2HE2, 7X, 4HRASS, 6X,
1X,4HE', 4X, 2HNUP, 4X, 2HNUPES, 2X, 3X, 6HINC, 2X, 5X, 4HMAY, 6X, 3HP, 4X,
24X, 1HWP, 1
71 FORMAT(F5.1, 1H3F3, 1P11G10.2, 1D2)
72 RETURN
END
SUBROUTINE ORDER
T HIS ROUTINE TAKES THE BODY LIST READ FROM CARDS AND SORTS THEM IN
ORDER SO THAT THE DISTANCE FROM THE REFERENCE TO EACH BODY IS
DEPENDENT UPON ALREADY COMPUTED DISTANCES ONLY.
C ELLIPSE DATA ARE READ INTO A BLOCK OF 120 STORES RESERVED FOR
TEN ELLIPSES. ONE ELLIPSE IS READ INTO A 12 STORE BLOCK.
THE SINES AND COSINES OF THE 3 ANGLES ARE COMPUTED AND STORED IN
THE TDATA ARRAY ALONG WITH THE REST OF THE ELLIPSE DATA.
A BLOCK IS ARRANGED AS FOLLOWS:
C 1) NAME OF BODY IN BCD; ONLY & CHARACTERS.
C 2) NAME OF REFERENCE BODY IN BCD; SAME RESTRICTION.
C 3) MASS OF THE BODY IN SUN MASSES.
C 4) RADIUS INSIDE OF WHICH COORDINATES WILL BE TRANSLATED TO THIS BODY.
C 5) SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
C 6) ECCENTRICITY OF THE ORBIT.
C 7) LONGITUDE OF ASCENDING NODE.
C 8) 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 AAMASS = MASS OF EACH BODY, SUN MASSES. ORDER OF PNAME.
C AAMASS = 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 BODYS = THE LIST OF BCD BODY NAMES WITH THE REFERENCE BODY AT TOP.
C IBODY = I = BODY CARD INDEX. CORRESPONDS TO BODY CD, SAME RESTRICTION.
C IBODY = ARRAY OF SUBSCRIPTS. WHEN A DISTANCE IS FOUND FROM EPHEMERIS, IT MAY
C MAY BE ADDED (OR SUBTRACTED) FROM THE BODY POSITION GIVEN BY
C XPRI(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 NAME = ARRAY OF SUBSCRIPTS. INVERSE OF BNAME. GIVEN NEW LOCATION OF
C NAME LIST IN TERMS OF BODY. NOT IN COMMON.
C NRODS = COUNTED INTERALLY. TOTAL NUMBER OF EPHEMERIDES NRODS+11.
C NRODS = COMPUTED INTERALLY. TOTAL NUMBER OF EPHEMERIDES NRODS+11.
C NAME = ARRAY OF SUBSCRIPTS. GIVES DLL LOCATION OF NAME IN PNAME LIST.
C IN TERMS OF THE EPFRS LIST. STORED IN COMMON.
C NREFER = ARRAY OF SUBSCRIPTS. LOCATES THE REFERENCE BODY IN BODY.
C ORDER OF THE ARRAY CORRESPONDS TO BODYS, NOT IN COMMON.
C NREFER = ARRAY OF SUBSCRIPTS. LIKE NREFER BUT REFER TO BNAME LIST.
C NOT IN COMMON.
C PNAME = A PERMANENT LIST OF BCD BODY NAMES. 1 WORD EACH 16 CHARACTERS
C MAX. USED TO IDENTIFY MASS, REFERENCE NAMES, ETC. THE LIST IS A
C 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 BODIES THAT ARE THE REFERENCES OF
C DISTANCES GIVEN IN EPHEMERIDES (TAPES OR ELLIPS). CORRESPONDS TO
C PNAME LIST. COMMON C
C COMMON C
C DIMENSION A(4000), B(4000), C(4000)
C 1 AAMASS (30)
C 2 AMASS (30)
C 3 BNAME (81)
C 4 BNAME (81)
C 5 NAME (81)
C 6 NAME (81)
C 7 NREFER (81)
C 8 NREFER (81)
C 9 NAME (30)
C 10 NAME (30)
C 11 REFER (30)
C 12 REFER (30)
C 13 BODY (7)
C 14 BODY (7)
C 15 TOL (71)
C 16 TOL (71)
C 17 ELIPS (71)
C 18 ELIPS (71)
THIS SECTION SEES WHAT ELLIPSE DATA WAS READ FROM CARDS AND PUTS THE NAMES IN PLACE SO THAT DATA WILL BE USED IF NEEDED. ELLIPSE DATA HAS PRIORITY OVER TAPE DATA BECAUSE LAST DATA IN LIST IS THAT ACTUALLY USED.

FUNCTION COMPARF(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.

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  
PHNAME(KOUNT) = ELIPS(K)  
REFER(KOUNT) = ELIPS(K+1)  
AMASS(KOUNT) = ELIPS(K+2)  
RCRIT(KOUNT) = ELIPS(K+3)  
2 CONTINUE

PART 0. THROW AWAY BLANKS AND DUPLICATES IN BNAME LIST.  
ALSO COUNT THE BODIES.

IF (TRANSFER) = 4,3,4
3 BNAME(1) = BODYCO(1)  
4 DO 5 K=1,8  
5 BNAME(K+1) = BODYCU(K)  
L = 1  
BODY(1) = 0.  
DO 8 J=1,9  
BODY(J) = 0.  
8 DO 6 K=1,1  
IF (COMPARE(BNAME(1), BODY(K-1))) = 6,7,6  
6 CONTINUE  
BODY(1) = BNAME(1)  
L = 1  
7 BNAME(1) = 0.  
8 CONTINUE  
NBODYS = L-1  
MBODYS = NBODYS-1

PART 1. FIND THE REFERENCE BODY FOR EACH BODY IN THE LIST OF BODIES READ FROM CARDS. CLEAR NREFER AND BNAME.

DO 13 KL=1,NBODYS  
NREFER(KL) = 0  
NPMTR(KL) = 0  
BNAME (KL) = 0.  
DO 12 KP=1,30  
IF (COMPARE(BODY(KL),PMTR(KP))) = 12,9,12  
9 NPMTR(KP) = KP  
DO 11 KK = 1,8  
IF (COMPARE(REFER(KP),BODY(KL))) = 11,10,11  
10 NREFER(KP) = KK  
11 CONTINUE  
12 CONTINUE  
13 CONTINUE

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  
NREFER(KP) = NREFER(KP)  
IF (NPMTR(KP)) = 18,17,18  
17 MISPEL = MISPEL + 1  
18 IF(NREFER(KP)) = 20,19,20  
19 KZEROS = KZEROS + 1  
20 CONTINUE  
21 IF (KZEROS) = 24,22,24  
22 IF (MISPEL) = 24,23,24  
23 IF (NBODYS) = 24,22,24

PART 3. REPORTS ERRORS IN BODY LIST.

WRITE OUTPUT TAPE 4.25, NBODYS,MISPEL,KZEROS(BODY(KL),K),NBODYS  
WRITE OUTPUT TAPE 6.25, (NREFER(KP),K),NBODYS  
WRITE OUTPUT TAPE 6.27, (K),NBODYS  
WRITE OUTPUT TAPE 6.30, (K),NBODYS  
WRITE OUTPUT TAPE 6.31, (K),NBODYS  
WRITE OUTPUT TAPE 6.32, (K),NBODYS

PART 4. TRACES OUT "REFERENCE TO BODY" RELATIONSHIPS.

28 K = 2  
KN = 1  
NAME(KP) = 0  
29 IF (NREFER(KP)) = 20,31,30  
30 NAME(KP) = NREFER(KP)  
NAME(KP) = 0  
KN = NAME(KP)  
KA = KA + 1  
31 GO TO 29
PART 5. TRACES OUT **BODY TO REFERENCE** RELATIONSHIP

DO 34 KN = 1,NBOODYS
DO 34 K = 1,NBOODYS
IF (INNREFR(K) == NAME(IN)) 34,33,34
NAME(IN) = K
KK = KK + 1
34 CONTINUE

C C PART 6. INVERTS NAME TO MANE STORES BNAME, BMAS S, RBCRIT, AND A
C TEMPORARY NEFMRS.
DO 35 K = 1,NBOODYS
N = NAME(K)
NAME(IN) = K
NEF = NEFMRS(K)
BNAME(K) = NAME(IN)
BMAS(K) = BMAS(IN)
RBCRIT(K) = RBCRIT(IN)
NEFMRS(K) = NEF
35 CONTINUE

C PART 7. FINDS NNREFR REFERENCE FOR BNAME LIST, ALSO TEMP. IBOODY
DO 36 K = 1,NBOODYS
N = NAME(K)
NRF = NNREFR(N)
NNREFR(K) = MANE(NRF)
IBODY(K) = MANE(NRF)
36 CONTINUE

C PART B. FINDS IBODY FOR BACKYARD REFERENCE.
DO 39 K = 1,NBOODYS
N = NAME(K)
NNREFR(K) = MANE(NRF)
IBODY(K) = -K
39 CONTINUE

IBOODY LIST IS COMPLETE.

C PART 9. WRITES OUT EPHEMERIS LIST TO BE USED IN STORING DATA AND
C MAKES FINAL NEFMRS LIST.
DO 40 K = 1,NBOODYS
IF (NNREFR(K) == 24) 40,38
40 CONTINUE

C PART 10A. LOADS A FALSE (VERY EARLY) TAPE TIME TO FORCE TAPE
C READING BY THE EPHEMERIS ROUTINE. FILE = 0 UNLESS TAPE
47 IDEL(K) = 0
TIM(K) = 2400000.5
FILE = 10.
48 CONTINUE

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.
49 REVOLV = 0.
REVOLV = 0.
RAATMOS = 0.
49 CONTINUE

C PART 12. WRITES THE BNAME LIST ON TAPE 6.
C PART 12A. WRITES THE BNAME LIST ON TAPE 6.
56 IF (OUTPUT TAPE 6) 58,59,58
58 RETURN
END
SUBROUTINE ODLATE

THIS SUBROUTINE COMPUTES THE OBLATENESS ACCELERATIONS (ODLAT) DUE TO AN AXIALLY SYMMETRIC EARTH. THE 2ND, 3RD, AND 4TH SPHERICAL HARMONIC COEFF.

ARE OBLATJ, OELATH, AND OBLATO RESPECTIVELY.

COMMON C

DIMENSION A(600), B(700), C(4000),
1 RB(3), OBLAT(3)

EQUIVALENCE

1IA , A ( 1111), B ( 1111), C ( 1111) , G2M , B ( 36) ,
2OBLATJA ( 261) , OBLATD( A ( 27) , OBLATHA ( 28) ,
3( OBLAT - B ( 75) ) , R ( B ( 102) ) , ( RB + B ( 193) ) ,
4RE , A ( 25) , RSQRD , B ( 45) , ( RESQRD , B ( 71) )

AA = RB(3)/R
AB = AA*AA

IF ( ABSFIAAI-1.E-61 1,1,2
1 AAA = O.
2 AC = RE/SQRO/RSPRD
3 AD = GK2M/RSPRO/AC
4 AE = OBLATJ*AO
5 AF = OELATH*AU*RE/R
6 AG = OBLATJ. AO*AC
7 AH = AE*AA+AG*14.AB-1.714285714))R8(3-AF*l3.AE-O*61*R
8 RETURN

END

SUBROUTINE OUTPUT

ENTS AND RECTANGULAR COORDINATES ARE OUTPUTTED. IF THE OBJECT IS NOT WITH THIS IS THE ROUTINE WHICH FORMS THE BASIC DATA OUTPUT. BOTH ORBIT

EQUATION AND THE ROUTINE WHICH FORMS THE BASIC DATA OUTPUT. BOTH ORBIT ELEM-

IN AN ATMOSPHERE IPRESS=O.I. ONE LINE OF DATA IS DELETED. LIKEWISE,

ONLY THOSE PERTURBING BODIES PRESENT HAVE THEIR DISTANCE OUTPUTTED.

COMMON C

DIMENSION A(600), B(700), C(4000),
1 R ( 81) , ORBELS ( 61) , VATM ( 31) ,
2 BNAME ( 8) , RB(3,8) , DIRCOS(3,8) ,
3 XPRIM ( 2001) , RAMC ( 51) ,

EQUIVALENCE

1IA , A ( 1111) , B ( 1111) , C ( 1111) , ALPHA , A ( 491) , ALT , A ( 411) ,
2 IAM , B ( 871) , AM , B ( 901) , AREA , B ( 631) ,
3NAME , B ( 1221) , C ( 1111) , CDG , A ( 1651) ,
4ICL , A ( 1661) , C(SDFALF , B ( 481) , C(SSTRU , B ( 531) ,
5STOFFJ , A ( 231) , H2 , B ( 151) , ( IMODEA , A ( 111) ,
6NBODYS , B ( 421) , NBODYS , B ( 411) , ORBELS , B ( 1161) ,
7PRESS , B ( 331) , P , B ( 841) , ( PSI , B ( 301) ,
7BISR , A ( 3901) , ( PUSH , A ( 1601) , G , B ( 591) ,
9BAM , A ( 3951) , RB(3, B ( 1931) , RESV , A ( 481) ,

EQUIVALENCE

1IR , B ( 1021) , ( SINALF , B ( 461) , SINSTRU , B ( 521) ,
2DISTEGD , A ( 411) , ( STEPNO , A ( 421) , ( TABLE , B ( 201) ,
3SINU , B ( 901) , VATM , B ( 971) , ( VQ , B ( 1001) ,
4IV , B ( 951) , ( VX , B ( 921) , ( VY , B ( 931) ,
5VZ , B ( 981) , XPRIM , C ( 7111) , ( OUTPUT , B ( 3991) ,

DAYJ = DTUFFJ-2.461+TABLE
ALPHA = ALPHA+57.29577951
REV = REV+20(RB(3)-RE(11))/6.29318532*.5
16 CALL CONVTVX(AMC)
IMODE=IMODE
GO TO (Z,1,1), IMODE
1 CODE=6RRECTAN
18 CALL CONVT 2
GO TO 4
2 DO 3 K=1,6
3 ORBELS ( K ) = XPRIM ( K + 2]
CODE=SHORBIT
TRU=RACTANT(SINSTRU,COSTRJ)
4 PSI = ATANF(BRI1)+YR+BZ(2)+YR+RB(3)+VY+RB(3)+B/AM)+57.2957795
IF (OUTPO) 19,6,19
6 WRITE OUTPUT TAPE 6, 11,STEPG,STEPNO,ORBELS,VR,RB(1),
8NAME(1),CODE,IMODE,XPRIM,ORBELS,TRU,VX,VR(1),XRPRM,DAYJ,Q,
2RBL(3),VR(3),VR(2),REV,ALPHA,P5,ORBELS(4),ZVB,RB(3),
C IF WITHIN AN ATMOSPHERE COMPUTE DRAG, LIFT, G, ETC., AND PRINT EXTRA LINE.
10 IF (PRESS) 5,7,9
5 XIFT = G*AREA+G
G = ( PUSH=DRAG+CSSALF*XIFT+SINALF )*XPRIM/2(9.1965)
17 CALL CONVTIVAT,RAMC
PSIR = ATANF(BRI1)+YR+BZ(2)+YR+BATM(3)+YR+BZ(3)+BATM(3)+RAMC(4)+11,
157.2957795
IF (OUTPO) 7,14,7
14 WRITE OUTPUT TAPE 6, 12, ALT, PSIR, DRAG, VQ, G, PUSH
IF PERTURBATING BODIES ARE PRESENT, FIND THEIR DISTANCES AND PRINT THEM.

WRITE OUTPUT TAPE 6,10,8

FUNCTION QUAD (X,IC)

THIS ROUTINE COMPUTES AN VARIABLE, QUAD, AS A QUADRATIC FUNCTION OF X.

BELOWING TO A PARTICULAR REGION OF X, THE COEFN ARRAY IS ARRANGED AS ---
WHERE A1, B1, C1 ARE THE COEFFICIENTS TO BE USED FOR X BETWEEN X1 AND X2, ETC.
AND X1 IS LESS THAN X2, X2 IS LESS THAN X3, X3 IS LESS THAN X4, ETC.
IC IDENTIFIES WHICH DEPENDENT VARIABLE, QUAD, IS BEING SOUGHT.
IC(C)I(C) DEFINE THE STARTING LOCATIONS IN THE COEFN ARRAY FOR VARIABLES X.

COMMON C

DIMENSION A(600), B(700), C(4000),
1 COEFN(190), ICC(5)

EQUIVALENCE

1A C IC (111), B C (11111), COEFN A (407),
2ICC A (153)

I=C(C)C
1 IF (X-COEFN(I)) 2,3,3
2 I = 1
GO TO 1
3 IF((IC+4)) 5,5,4
4 I = 1
GO TO 3
5 QUAD = COEFN(I+1)*X+COEFN(I+2)*X+COEFN(I+3)
ICC(I)=I
RETURN
END

SUBROUTINE STAGE

THIS ROUTINE IS CALLED TO PREPARE DATA FOR USE IN NBOOY. STAGE DATA IS TAKEN FROM PERMANENT STORES AND LOADED INTO WORKING STORES. STAGE DATA MAY BE SET ASIDE FOR LATER USE (IF ON NSAVE-NSTAGE). WHEN IMODE IS 4, CONVERSION FROM EARTH-SPERICAL TO RECTANGULAR OR ORBIT ELEMENTS TAKES PLACE IN TUES.

COMMON C

DIMENSION A(6001), B(7001), C(40001),
1 XPRIM(200), XPRIM(200), T(10), FLOW(10), AEXIT(110), SIMPL(10),
2 AREA(1101), OELT(110), I100, FINV(1), FNT(6), FNDIV(1), FNCF(62)

EQUIVALENCE

1A C I (111), AEXIT B (33), AEXIT A (1033),
2AREA A (111), AREA B (63), B C (11111),
3DELT A (1333), DELT B (11), ID C (211111),
4DEL A (433), OELMAX A (191), IDNCR B (391),
5EREF A (133), ERLOG B (17), EEXIT B (392),
6FLOW B (53), FLOW A (833), IDENT A (1233),
7IMODE A (117), (LSTAGE A (387), MDOOUT A (203),
8INCASE C (111), INCASE A (600), INSAVE C (43),
9NSTAGE A (33), (PUSHD B (3913), (RMASS A (733)
EQUIVALENCE

1SIMPL A (933), SIM P B (27), T7 B (633),
2TABLE C (1911), TKICK A (153), TMAX B (43),
3TITOL A (451), (XPRIM C (911), (XPRIM C (711),
4RETURN B (4001), (OUTPUT B (399)}

C
PART 0. SAVE INITIAL DATA IF DESIRED. LOAD STAGE DATA INTO WORKING
STORAGE, ALLOW ADDITIONAL STAGE INPUT.

IF (DEL) 100,99,100
99 DEL = DELMAX-TRACK
100 IF (NSAVE=NSTAGE) 103,101,103
101 NCASES = NCASE
GO TO 102
102 DJU = DJU
IF (OUTPUT) 103,97,103
97 WRITE OUTPUT TAPE 6,98,NSTAGE+CASE
98 FORMAT (24M SAVED INITIAL DATA FOR STAGE 12,OH OF CASE 14,1H)
103 NSTAGE = NSTAGE+1
TMAX = XPRIM11+TMIN
XPRIM01 = 0.
XPRIM02 = XPRIM01*RMAX
IF (RMAX) 117,117,110
110 GOTO 119
117 XPRIM02 = RMAX
FLOW = FLOW+2
FLOOR = FLOOR+I
SIM = SIM+2
AEXIT = AEXIT+1
AREA = AREA+1
DELT = OEL+0.
1D = IDENT+1
CALL INPUT 100C TABLE
ERLOG = LOGFABSFLEREF+I
TTOL = 5E-8*ABSFLTMAX+I+E-0
PUSH0 = SIM+FLOW+8,0665
EXITA = AEXIT+100.
MODOUT = MODOUT+1
IF IDEL 105,104,105
104 DEL = -1*MINXSTAGE+100.
DELSTAGE = DELT
105 GO TO (109,104,109,104,109,107)
106 IF (DEL=DELMAX) 108,108,107
107 DEL = MODFUEL,DELMAX
108 IF (ID) 114,114,114
114 DEL = MIN(DEL,DEL)
109 IF (XABSFlMODE)-4,1,110,1
110 CALL TUNES
(MODE = XSIGNFIZIIMAX)
111 CALL NBOUY
112 CALL EXTRAS
C
C PART 9. COMES HERE FOR END OF SUB TRAJECTORY-
IF (DONE) 113,113,111
111 DONE = 0.
112 NSTAGE = NSTAGE+1
GO TO 100
113 DONE = 0.
115 CALL EXTRA
IF (RETURN) 103,116,100
116 RETURN
END

SUBROUTINE STEP
SUBROUTINE TESTS FOR THE END OF THE PROBLEM, COMPUTES STEP SIZE, AND
CONTROLS QUANTITY OF OUTPUT DATA. END OF PROBLEM OCCURS IF 
TIME = TMAX, 
STEPGO+STEPNO = STEPMX, OR CILOOKXI = XLOOK. THE LAST OPTION ALLOWS STOP-
PING ON A DEPENDENT VARIABLE. THE TEST FOR STOPPING AT XLOOK IS NOT MADE
UNTIL CILOOK(X) IS GREATER THAN SYLOOK. CONTROL ON QUANTITY OF OUTPUT IS
MOOOUT=I OUTPUT EVERY NTH STEPlN=STEPSI UNtiL TIME = THIN, THEN
2 OUTPUT AT INTERVALS 
OF UELMAX UNTIL TIME = TMAX. 
3 OUTPUT AT INTERVALS 
OF UELMAX UNTIL
TIME = THIN, THEN 
GO TO MODE 4 . 
4 OUTPUT EVERY NTH STEP 
UNTIL TIME = TMAX.

COMMON C
DlMENSION A16000, BITOD1, C14000,
1 XPRIM12000, DELT (10)

EQUIVALENCE
1A ,C ( 111111,1),F ( 101),F (2),A +B ( 1111 ),
1B ,C ( 111111,D,LONMAX,A ( 1011,DEL +A ( 4311 ),
31DEL +B ( 111,1,DONE +B ( 3911,F ( 1911 ),
41NO ( 1 C ) C ( 3111 ),ELOG +B ( 2711,F ( 1511 ),
51INLOOK +A ( 5991),LLOOK +A ( 911,F (LLOOK +A ( 811 ),
61MODOUT +A ( 2011),NSTAGE +A ( 371,F (DELT +A ( 1311 ),
71RATIO +B ( 5813),SIGNAL +B ( 3111,SPACES+10 ( 1611 ),
81STEPG +A ( 411),STEPX,A ( 1611,STEPNO +A ( 4211 ),
91STEPS +A +T ( 1713,SWUO+1071,TABLE +C ( 1911 ),
31MAX +B ( 413),TMIN +A ( 1811,TTOL +A +G1511 ),
21LOOK +A ( 1213),XPRIM +C ( 17111),TTOL +A ( 1111 ),
31START +B ( 2613),SWITCH +B ( 16111),OUTPUT +B ( 39111 ),
CHECK(A,B,C) = ABSF(A-B) = ABSF(A-C)
C
PART 1. TEST FOR END OF THE PROBLEM (MAXIMUM PROBLEM TIME OR MAXIMUM
NUMBER OF STEPS).
STEPGO = STEPGO + 1.
OUT = OUTPUT
IF (ABSFLMAX=XPRIM11111)-TTL) 1,1,1
1 DONE = 1.0
112 CALL OUTPUT
IF (OUTPUT) 26,111,26
111 WRITE OUTPUT TAPE 6,2,NSTAGE
2 FORMAT(12HSTAGE2121H COMPLETED.//)
GO TO 26
3 IF (STEPGO+STEPNO=STEPMX+1,1)
4 CALL OUTPUT
WRITE OUTPUT TAPE 6,5,STEPMX
5 FORMAT (22M STEPGO+STEPNO=STEPMX+1,1)
CALL EXIT
C PART 2. COMPUTE STEP SIZE (DELT) AND CONTROL OUTPUT.

7 N=1

A3 = (A2-A1)*RATIO+4
A1 = (ERLOG-A3)/5.

IF (ABS(F(AA)+BB,02)+ABS(F(SWITCH))) > & \epsilon_0 \delta_0

8 DELT = SIGN(FEXP(AA),DELT)

9 DELT = 3.*MC

10 MOUT = MOUT

GO TO (11,12,13,21),MOUT

11 IF (DELT+XPRIM(1)+3.*DELT-TMIN)) > 21,12,12

12 MOUT = 2

DEL = MIN - XPRIM(1)

GO TO 16

13 IF (DELT = XPRIM(1) - TMIN)) > 15,15,14

14 MOUT = 4

GO TO 21

15 DEL = DEL-MC

16 SPACES = INT(ABS(FIOX))

17 IF (SPACES) > 8,8,8

18 CALL OUTPUT

N=2

IF (ABS(DEL) = ABS(DEL)) > 19,16,16

19 DELT = SIGN(DEL,DELT)

GO TO 16

20 DEL = DEL/SPACES

GO TO 23

21 IF (MODFSTEPPGO,STEPS)) > 23,22,23

22 CALL OUTPUT

N=2

C

C PART 3. SEARCH FOR C(LOOKX) = XLOOK UNLESS LOOKX=0.

23 IF (LOOK = 27,42,27

24 LOOK X = LOOK X

LOOK SW = LOOK SW

OUTPUT = 1.

GO TO (44,45,46)

44 CALL OUTPUT

45 IF (SWITCH) > 32,28,33

46 IF (SW) > 29,29,29

47 XTOL = XTOL+ABS(F(LOOK))

IF (XTOL) > 26,30,31

48 SWITCH = -1.

49 ASSING = 0.

50 SWITCH = 1.

ASSING = 35 TU MODE

OVER = 0.

F = 0.

T=X.

33 SLOPE = (C(LOOKX)-LOOKX)/H2

GO TO MODE, (43,35)

43 IF (SLOPE = C(LOOKX)) > 35,41,41

35 ASSIGN 35 TU MODE

36 IF (OUT) > 63,44,63

46 OUTPUT = 0.

51 WRITE OUTPUT TAPE 6.64.

LOOK = 0.

WRITE OUTPUT TAPE 6.48,LOOK X,CILOOK X

FORMAT13HOC114H=CONVtRGENC TROUBLE.

GO TO 62

47 IF (OUT) > 62,50,62

50 WRITE OUTPUT TAPE 6.48,LOOK X,CILOOK X

48 FORMAT114(4H1) = 1P15.8,18H

60 T=1.

38 SIGN = CHECKF(LOOK,LOOKX,CILOOK X)

40 OVER = 3.

DONE = END

NSTAGE = NSTAGE

DEL = DELT(NSTAGE)

49 CALL INPUT(LOOKX,TABLE)

IF (DONE) > 110,110,110

110 IF (OUT) > 25,111,26

37 SIGN = CHECKF(LOOK,LOOKX,CILOOK X)

IF (SIGN) > 40,40,38

40 OVER = 3.

GO TO 400

38 IF (COVER) > 400,400,400

401 XGUESS = CILOOKX+SLOPE+DELT

IF (CHECKF(LOOK,LOOKX,XGUESS)) > 402,41,41

402 F = F+F1.

403 SIGN = SIGMA(ABS(F(LOOKX),F(LOOKX)))/SLOPE,SIGN(2)

41 ODX = CILOOK X

42 IF (ABS(FMAX-XPRIM(1)ABS(FDELT))) > 52,26,26

25 DELT = TMAX-XPRIM(1)

GO TO (26,24,24,26),MOUT

DEL = DEL=DELT

26 OUTPUT = OUT

RETURN
SUBROUTINE STOATA

THIS ROUTINE CLEARS THE XPRIM, XPRIMB ARRAYS AND LOADS A SET OF
STANDARD DATA INTO THE MACHINE. ANY VALUES SET HERE MAY BE OVERWRITTEN BY
INPUT 1 IN THE MAIN PROGRAM.

COMMON C

DIMENSION A(600), B(700), C(6000),
1 PNAME (112), AMASS (301), XPRIM (200),
2 CODEFN (190), IEC (4),
3 AK (3), XDOT (100), IND (3),
4 REFER (12), CRIT (301), AM (4),
5 RHASSI (10)

EQUIVALENCE
1IA ,C [111],(AK ,A [511],(AMASS ,A [347]),
13TFILE ,A [61],(XTOT ,A [501],(XPRIM ,C [71]),
14XTOL ,A [111],(RHASSI ,A [73])

CLEAR INITIAL CONDITIONS AND CONTROL PARAMETERS.
DO 1 J=1,1100
1 A(J) = 0.

THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
PNAME(1) = JHUSD
PNAME(2) = JHMERCU
PNAME(3) = JVENUS
PNAME(4) = JERTH
PNAME(5) = JHMAR
PNAME(6) = JHJUPITE
PNAME(7) = JHSATURN
PNAME(8) = JHURANUS
PNAME(9) = JHNEPTUN
PNAME(10) = JPLUTO
PNAME(11) = JMOON
PNAME(12) = JHERM

FILL OUT SUN REFERENCE LIST. INITIALIZE MASS ARRAY.
DO 2 K=1,10
AMASS(K) = 1.
2 REFER(K+1) = PNAME(1)
REFER(12) = PNAME(1)

FILL OUT EARTH REFERENCE LIST.
REFER(11) = PNAME(4)
REFER(4) = JZERD
REFER(11) = PNAME(4)
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 *1* OR *2*) TO CHANGE THE
C VARIABLES OF INTEGRATION (WHEN IMODE IS *-1*). 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),
XPRIM(100,2),XPRIMB(100,2),XWHOLE(6),YEFIM(3,9),XVI3),
ORBELS(0),ORBMASS(0),BNAME(0),BNAME(0),BNAME(1),ORBMASS(1),BNAME(1)
C
EQUIVALENCE
10 T ,C ( 1),JMC ,B ( 07),J(ASTYPI ,A ( 71),
20 C (1111),DBMASS ,B ( 137),BNAME ,B ( 122),
30 CHAMP + B ( 29),TDELT ,B ( 13),BNAME ,B ( 363),
40 IMODE + A (1),NBODY3 ,B ( 41),ORBELS ,B ( 110),
50 TRCRIJ + B ( 145),IRH + B ( 193),REV3 ,A ( 401),
60 K + B ( 102),TSWOK,B ( 359),TABLE C ( 1911),
70 TMAX + B ( 4),ITASPER ,B ( 9),ITR3 ,B ( 401),
80 TEST + A ( 54),VEFP + B ( 2433),VX + B ( 923),
90 XPRIM + C ( 711),XPRIMB ,C ( 911),XWHOLE ,B ( 1101),
100 EQUIVALENCE
11 IOUTPOT,B ( 399)
C IMODE = IMODE
IF (IMODE) 12,12,1
C
C IF IMODE IS *, TEST FOR TRANSLATION OF THE ORIGIN.
100 CHAMP = 1,E+30
DO 4 JB=1,4
IF (RIJ(B)=RUCRIT(JB)) 2,4,4
2 IF (CHAMP=RUCRIT(JB)) 4,4,3
3 CHAMP = RUCRIT(JB)
NCHAMP = JB
CONTINUE
IF (NCHAMP=1) 26,26,5
TSUPER = 1.0
8 BTEMP = BNAME
BNAMEN = BNAME(NCHAMP)
BNAMENBNAME(NCHAMP) = BTEMP
TEST = 0.
REV3 = 0.
IF (IOUTPOT) IO,4,6
9 WRITE OUTPUT TAPE 6,10,BNAME(NCHAMP),BNAME(1)
10 FORMAT (28MOORIGIN IS TRANSLATING FROM A6,A4 TO A6)
6 CALL EPHMR
DO 11K=1,3
VX(K) = VX(K)-YEFMR(NCHAMP)
BRI(K) = RB(K,NCHAMP)
XPRIM(K+2,1)=VX(K)
XPRIM(K+5,1)=RI(K)
C XPRIM(K+2,1) = 0.
XPRIM(K+5,1) = 0.
XWHOLE(K)=VX(K)
11 XWHOLE(K+33) = RB(K)
GO TO 20
C
C IF IMODE IS -, CHANGE THE VARIABLES OF INTEGRATION.
12 DO 13K=1,3
XPRIM(K+2,1)=XWHOLE(K)
XPRIM(K+5,1)=XWHOLE(K+3)
XPRIM(K+2,1) = 0.
XPRIM(K+5,1) = 0.
VX(K) = XWHOLE(K)
13 RB(K) = XWHOLE(K+3)
GO TO 16,14,15,IMODE
14 CODE = SURFTR
IMODE = 1
GO TO 18
15 IMODE = 3
GO TO 17
16 IMODE = 2
17 CODE = 6,RECTAN
18 NCHAMP = 1
9 WRITE OUTPUT TAPE 6,19,20
19 FORMAT (33MINTEGRATION MODE IS CHANGING TO A6)
20 GO TO (21,26,26),IMODE
21 CALL CONVT1(VX,AMC)
GR2M+ SURDOK+ORBMASS(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 (ABS(THR)-2.3/SQRT(ORBELS(1))) 24,24,23
23 ASYMPT = 1.0
GO TO 15
24 DO 25J=1,6
25 XPRIM(J,2,1) = ORBELS(J)
26 IF (ITASPER) 27,28,27
27 CALL INPUT (I01,C,TABLE)
29 CALL ORDER
28 RETURN
END
SUOROUTINE THRUST

THIS ROUTINE COMPUTES X, Y, Z THRUST ACCELERATIONS. THE THRUST VECTOR IS

ASSUMED COINCIDENT WITH THE LONGITUDINAL AXIS OF THE VEHICLE WHICH IS

ORIENTED TO THE RELATIVE WIND VELOCITY BY THE ANGLE OF ATTACK (ALPHA) AND

THE ROLL ANGLE (BETA). ALPHA IS ASSUMED TO BE A QUADRATIC FUNCTION OF TIME

WHEREAS BETA IS ASSUMED TO BE CONSTANT.

REVOLV IS THE EARTH'S ROTATION RATE IN RADIANS/SEC (7.29211585E-5) AND THE

FACTOR 8589934592 = 2**33 IS REMOVED TO PREVENT OVERFLOW.

COMMON C

DIMENSION AI(600), BI(700), CI(4000),
  1 FORCE(3), PAR(3), VATM(3), P(3), IND(3), RAMC(5), R(3), X(100)

C EQUIVALENCE
1A  ,C   ( 111), (EXIT), B   ( 31), (ALPHA), A   ( 49),
2B  ,C   ( 1111), (BETA), A   ( 501), (COSALPHA), B   ( 48),
2GOS大面积 ,B   ( 49), (EXIT), B   ( 392), (FLOW), B   ( 53),
4FORCE ,B   ( 663), (IND), A   ( 601), (PAR), B   ( 601),
5PMAGN ,B   ( 501), (PRESS), B   ( 331), (P), B   ( 84),
6PMAGH ,B   ( 3911), (PRESS), A   ( 1651), (FRAMC), B   ( 393),
7RATMOS,B   ( 231), (R), B   ( 1931), (REVOLV), B   ( 211),
8BETA ,B   ( 1023), (RESQR), B   ( 451), (SIMP), B   ( 21),
9SINH ,B   ( 491), (SINBETA), B   ( 473), (VATM), B   ( 97),
2COSH ,B   ( 1001), (COSBETA), B   ( 1011), (IX), B   ( 8),
C2IVY ,B   ( 931), (IV), B   ( 941), (IX), B   ( 601)

C EQUIVALENCE

SINBETA = SIN(BETA)/57.2957795
COSBETA = COS(BETA)/57.2957795
VATMI1 = VATM1 + REVOLV * RB
VATMI2 = VATM2 - REVOLV * RB
VATMI3 = VATM3
3 CALL CONVT1IVATM(RAMC)
4 ALPHA = QUADRI(A11, A12, A13)/57.2957795
SINALF = SIN(ALPHA)
COSALF = COS(ALPHA)
DO 1 J1 = 1, 3
J2 = IND(1)
J3 = IND(2)
1 PLJ1 = (VATMI1*RAMC(3) - VATMI2*RAMC(2))/BS99936592.
PMAGN = SQRT(PLJ1 + PLJ2 + PLJ3 + PLJ4)
PUSH = PLJ1 / PMAGN
TOPMAG = PUSH / PMAGN
K4 = SINEHY/VG
R5 = COSALF / RAMC(4)
DO 2 J1 = 1, 3
J2 = IND(1)
J3 = IND(2)
2 PARLJ1 = PLJ1 + VATMI3 - PLJ2 + VATMI1
FORCE(J1) = TOPMAG * SINALF * (COSBETA * K4 - PARLJ1) - R5 * (PARLJ2 *
  1 RAMC(3) - PARLJ3 * RAMC(2))
RETURN
END

SUBROUTINE TUDRES

THIS ROUTINE COMPUTES THE RECTANGULAR POSITION AND VELOCITY COMPONENTS

WITH RESPECT TO THE EARTH MEAN EQUINOX AND EQUATOR OF 1950.0 FROM THE

LATITUDE, LONGITUDE, AZIMUTH, ELEVATION, ALTITUDE, TOTAL VELOCITY, AND

TIME. ALSO, WHEN TKICK DOES NOT EQUAL ZERO, A NON-DRAG VERTICAL STEP OF

SIZE TKICK IS MADE IN CLOSED FORM (STATEMENTS 2 TO 4). THE INTEGRATION

WILL THEN BEGIN AT TIME EQUAL TO TIME + TKICK WITH THE ORIENTATION SPECIFIED

BY THE ABOVE FOUR ANGLES AND THE COMPUTED VALUES OF ALTITUDE AND VELOCITY.

FOR THE CLOSED FORM APPROXIMATION, A CONSTANT FLOW RATE (FLOW), VACUUM

SPECIFIC IMPULSE (SIMP) AND ENGINE EXIT AREA (EXIT) ARE ASSUMED KNOWN.

THE ATMOSPHERIC PRESSURE IS TAKEN TO BE THE SEA LEVEL VALUE.

COMMON C

DIMENSION AI(600), BI(700), CI(4000),
  1 SIN(41), COS(41), ANGLE(41), XPRIM(200)

C EQUIVALENCE
1A  ,C   ( 111), (EXIT), B   ( 31), (LAT), A   ( 41),
2LAT ,A   ( 351), (R), A   ( 1111), (TDOFF), A   ( 231),
2B  ,C   ( 351), (R), C   ( 1111), (TDOFF), A   ( 231),
3B  ,C   ( 351), (FLOW), B   ( 51), (GRZM), B   ( 361),
4LAT ,A   ( 331), (LONG), A   ( 361), (DIROTAT), A   ( 241),
5SINLAT, A   ( 403), (R), A   ( 251), (RESQR), B   ( 71),
6ROTA ,A   ( 391), (SIMP), B   ( 271), (SRD), A   ( 441),
7SINEG ,A   ( 41), (ESTEPN), A   ( 421), (ESTEPN), A   ( 421),
8BITOFF ,A   ( 241), (VEL), A   ( 371), (EXPRM), C   ( 7111),
9HOUTPUT ,B   ( 391)
C EQUIVALENCE (QLATLAT), (QLONG,LONG)
ALTI = 0.
VEL1 = VEL
DELI = 0.
DEL = 0.
ASSIGN 1 TO NGO
DAYS = DTEOFF - 2433282.5
GREEN = MODF((100.075526+0.98564734*DAYS+2,9015-13*DAYS+2
1+292115880-64*(1+0.00181815))57.2957755,360)
SINA(1) = SINFLAT/57.2957955
IF IFLATNI = 102,101,102
101 RADIUS = RE + ALT
GO TO 8
102 RADIUS = 6356783.25*SVRT(1.99330657B3+.006693421685*SINA(1)+2)+ALT
GO TO 8
1 XPRIM(6) = COSA(11)+COSA(11)+RADIUS
XPRIM(7) = SINA(11)+COSA(11)+RADIUS
XPRIM(8) = SINA(11)+RADIUS
RMASS = XPRIM(2)
XPRIM(2) = XPRIM(2)-FLOW+TICK
IF (OUTPOTF) IZ.I IZ
11 WRITE OUTPUT TAPE 6.A3,STEPG,STEPNO,LAT,LONG,ALT,ALT.
SINA(11),VEL,PLAOSS0,XPRIM(11)=XPRIM(11)+6.8
11 = G15.8,8,7H ELEV=G15.8,6H ALT=G15.8,6H TIME=G15.8,6H VEL=G15.8,6H
67 RAMSS=G15.8,4,2H=G15.8,5X,2H=G15.8,4X,2H=G15.8,6H
12 IF (TICK) IZ.50.I2
2 XPRIM(1) = XPRIM(1)+TICK
3 I = LOGF(RMSS0/XPRIM(2))
SIMPSL = SIMP-AEXIT/FL0W10332.75
VEL1 = VEL1+IMPSL9.8065981+GATICK
ALT1 = TLCA1=VEL1=0.49*RO6659+SIMPSL(1).81-XPRIM(2)/
1 (RMASS-XPRIM(2))
69 RADIUS = RADIUS + ALT
GREEN = GREEN + 7.29211586E-5*TICK=57.2957955
ASSIGN 5 TO NGO
GO TO 8
5 XPRIM(6) = COSA(11)+COSA(11)+RADIUS
XPRIM(7) = SINA(11)+COSA(11)+RADIUS
XPRIM(8) = SINA(11)+RADIUS
50 IF IFLATNI = 6.7.6
6 DEL = ATAN(C2-1.1)/(C3.1)+SINA(11)/COSA(11)=57.2957975-QLAT
7 DEL2 = RADIUS/57.2957975=COSA(1)+ROTATE=ROTATE+57.2957975
DEL = DEL + DEL2
ASSIGN 10 TO NGO
ANGLB(1) = GAT+DEL
ANGLB(2) = GAT+GREEN
ANGLB(3) = AZ1
ANGLB(4) = ELEV
DO 9 I=1,4
SINA(1) = SINFLANGEB(11)/57.2957955
9 COSA(1) = COSFLANGEB(11)/57.2957955
C1 = 5.*RES/RADIUS/RADIUS+OBLATJ
C2 = C1*SINA(11)+SINA(11)+6
C3 = C2*SINA(11)+SINA(11)+2
G = G2*M+RADIUS
GO TO NGO, (1.5.10)
10 COSA = COSA(11)+SINA(4)+COSA(4)+COSA(3)+SINA(1)
COSB = COSA(4)+SINA(3)
XPRIM(3) = VEL1*(COSA+COSA(2)-COS2*SINA(2)+XPRIM(7)+ROTATE
XPRIM(4) = VEL1*(COSA+COSA(3)+COS2*SINA(3)+XPRIM(6)+ROTATE
XPRIM(5) = VEL1*SINA(11)+SINA(4)+COSA(11)+COSA(3)+COSA(4)
RETURN
END

SUBROUTINE TAPE
C SUBROUTINE TAPE USES THE MASTER MERGED EPHEMERIDES TAPE (TAPE 9 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 FORKAN 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 EACH EPHEMERIS COMPILATION RECEIVES A SET OF INPUT 300 DATA. FIRST PIECE
C OF DATA WRITTEN ON A FILE IS THE FILE IDENTIFICATION NUMBER, FILE.
C FILE IS NUMBERED CONSECUTIVELY STARTING WITH FILE=1, SINCE MUON DATA IS IN
C TERMS OF EARTH RADI1, 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 TAPE IS CLEARED BY A FINAL CLEARING LOOP.
C FUNCTION COMPARE(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C NORMAL INPUT - ELIST, TENABLE, 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. THE EPHEMERIDES ARE FINISHED THE
C NAMES ARE ERRASED FROM THIS LIST.
C TMAKE- LIKE TMAKE BUT IS HELD FOR OUTPUT.
C TMAKE- THE BEGINNING DATE EXPRESSED AS A JULIAN DAY.
C TEND- ENDING DATE EXPRESSED AS A JULIAN DAY.
C INVAL- THE APPROX. NUMBER OF DAYS COVERED BY ONE SET OF COEFF. IT
C IS USED TO DECIDE WHICH DATA ARE TO BE ENTERED UGABLE. THE
C DOUBLE ENTRIES PERMIT FASTER OPERATION IF REVERSAL OF
C THE APPRAOCH FOR THIS DATA ARE ENTERED.
C EDATE- JULIAN ENDING DATE FOR THE MASTER EPHEMERIS.
C ERTAU- EARTH RADI1 PER A.U.
C
C COMMON C
DIMENSION C (700), TMAKE (12), LIST (30),
2 EDATE (12), INTERVAL (30), KTAG (12),
3 ELIST (12), TMAKE (12), INTERVAL (2),
4 PNAME (30), DATUM (252), DATUM (21),

C EQUIVALENCE
1 TMAKE (C (2)), (C (2)), (C (4)), (C (1)),
2 INTERVAL (C (171)), (C (299)), (C (30)), (PNAME (C (11)),
3 KTAG (C (91)), (C (231)), (C (85)), (DATUM (C (64)),
4 EDATE (C (127)), (C (159)), (C (156)), (DATUM (C (189))

C

B COMPARFIA, B1 = (A+B)-(A*B)
REWIND 3
DO 1 K=1,4000
1 (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) = JHSUN
PNAME(2) = GHUMUR
PNAME(3) = SVENUS
PNAME(4) = GHMARS
PNAME(5) = GHJUPIT
PNAME(6) = GHSATURN
PNAME(7) = GHANUR
PNAME(8) = GHNEPTUN
PNAME(9) = GHPLUTO
PNAME(10) = GHMUND
PNAME(11) = GHARTH

C

C PART 2. SET UP JULIAN DATES ENDING EACH EPHEMERIS.
EDATE(1) = 2451872.5
EDATE(3) = 2451884.5
EDATE(4) = 2451892.5
EDATE(5) = 2473520.5
EDATE(6) = 2473520.5
EDATE(8) = 2473520.5
EDATE(9) = 2473520.5
EDATE(10) = 2440916.5
EDATE(11) = 2451872.5
EDATE(13) = 2451848.5
INTVA = 30000
INTVAL(1) = 8
INTVAL(2) = 5
INTVAL(3) = 15
INTVAL(4) = 44
INTVAL(5) = 330
INTVAL(6) = 825
INTVAL(7) = 121
INTVAL(8) = 117
INTVAL(9) = 1101
INTVAL(10) = 2
INTVAL(11) = 15
FILE = 1.
ERTDAU = 4.26546512 E-5
MOON = 0

C

C PART 2A. CALL INPUT AND SEE IF TAPE IS TO BE MADE. INPUT MUST ALWAYS
C MAKE TAPE3=O.0 IF TAPE IS TO BE MADE.
TAPE3 = 3.

C CALL INPUT(30000, LIST)
IF (TAPE3) = 0.3, 3, 63
IF (FILE) = I, 10, 10, 20
CALL SKFILEl9.21

C

C PART 3. TAPE IS TO BE MADE SO MOVE EPHEMERIS LIST TO TMAKE AND
C TO TMAOE (FOR OUTPUT), CANCEL ANY ZERO OR DUPLICATE NAMES.
KOUNT = 1
DO 10 K=1, KOUNT
10 DO 13 J=1, KOUNT
13 CONTINUE
KOUNT = KOUNT - I
CONTINUE

C

C PART 4. FIND INPUT ERRORS.
7 IF (BEGIN = 2437202.5) 66, 9, 9
9 KM = 2

11 ERROR = 0.
WRITE TAPE 3, FILE
DO 21 J=1, KOUNT
KTAG(J) = 0.
21 CONTINUE
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), BEGIN, TEND.
WRITE OUTPUT TAPE 6, 10, TMAKE(J), BEGIN, TEND, PNAME(K),
EDATE(K), K=1, 20
15 FORMAT (2H TROUBLE ON TAPE 3 MAKE / 2X, A6, 10H T BEGIN = F10.1, 8H
1 T END = F10.1, J/21, X, A6, F20.13)
ERROR = 1.
GO TO 21

C
PART 40. CHECKS DATES AND STORES INDEX FOR MOON SO THAT EARTH
RAOII CAN BE CONVERTED TO A.U.
16 IF (IO-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) 22,22,22,22

PART 6. FIX UP A TAG (KTAGI) TO INDICATE WHETHER TO ENTER DATA DOUBLE OR
NOT. KHAMP WILL BE SHORTEST INTERVAL. KTAG WILL BE NON-ZERO IF
ANY DATA ENTERS MORE THAN ONCE FOR 10 ENTRIES OF THE MOST
FREQUENT DATA.
22 KHAMP = INTERVAL(0)
DO 23 J=1,KOUNT
K = KTAG(J)
KHPH = XMINDF(KHAMP, INTERVAL(K))
23 CONTINUE
KHPH = KHAMP
24 KTAG(J) = INTERVAL(KHPH) / KHAMP

PART 7. LOCATE FILE 2 ON TAPE 9.
25 READ TAPE 9, KFILE
26 IF (KFILE = 3) 27,28,29
27 IF (KFILE - 20) 28,29
28 BACKSPACE 9
29 CALL BSFILE(9)
GO TO 25
30 BY PASS A FILE.
31 CALL BSFILE(9)
32 READ TAPE 9, KTAPE, (DATUM(K), J-1,25)

PART 10. PUT AWAY NEEDED DATA. TEST NAME, TIME OF BEGIN AND END. DO NOT
WRITE TAPE 3 UNTIL BEGIN PREDATES THE END OF THE FITTED
INTERVAL. So REPLACES OLD DATA, 57 WRITES NEW DATA. THE NAMES
ARE ERASED FROM TMAKE AS SOON AS THE DATA POST DATES TEND. WHEN
ALL NAMES ARE DONE, RETURN TO INPUT 300 TO SEE IF ANOTHER
EPHEMERIS IS TO BE CONSTRUCTED.
46 DO 65 K=1,25,21
47 DO 67 J=1,KOUNT
48 IF (COMPARE(DATUM(K), TMAKE(J))) 37,39,37
49 CONTINUE
50 SYT = TBEGIN - DATUM(K+1) - TOATUM(K-1)
51 IF (SYT) 52,57,58
52 WRITE TAPE 3, (DATUM(KJ), J=1,21)
53 CONTINUE
54 IF (MOON) 56,54,56
55 DATUM(TKJ) = DATUM(TKJ) + ERTOAU
56 IF (SWT) 57,57,58
57 WRITE TAPE 3, (DATUM(KJ), J=1,21)
58 IF (TEND - DATUM(KJ)) IF (TEND - DATUM(KJ)) 59,59,65
59 TMAKE(J) = 0.
60 DO 60 K=1,KOUNT
61 IF (TMAKE(K)) 65,60,65
62 WRITE OUTPUT TAPE 6, 6,1, FILE,TBEGIN,TEND, KOUNT, TMAKE(K), LK=1,
63 CONTINUE
64 WRITE FORMAT OF EPHEMERIS COMPLETED. FILE=F3,6H, FROM FII,1,3M TO
1 FIO,4,4H FOR 12, 18H BODIES AS FOLLOWS / 12123.4611
FILE = FILE + 1.
END FILE 3
GO TO 2
65 WRITE TAPE 3, FILE
66 WRITE TAPE 5, FILE
67 WRITE TAPE 9, FILE
68 WRIND 3
WDIND 9
TAPES + 3.
69 DO 64 J=3,4000
70 CONTINUE
71 RETURN
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
REWOLO V
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
BOD BSFILE
BSFILE
SKD <=-4,J
SKD <=-5,J
XEC* "$TEC"
TSX "$REK",4
LXO BSFILE-2,J
CLA* 1,J
TSX "$ODK",4
CLA* "$RDS"
STA BSF
ANA A07000
STA BT1
STA BT2
LXO BSFILE-2,J
CAL 2,J
ANA =0777777700000
ERA=0000704000000
TNZ ONEARG
CLA* 2,J
IZE BACK
POX J,J
AXC ++1,J
XEC* "$1CG"
BTT1
BTTA ++
TRA ++J
BSF BSFA ++
XEC* "$RDS"
XEC* "$ISR"
AXC ++1,J
XEC* "$1CG"
BTT2
BTTA ++
TRA CHECK
TXL BSF,1,J
XEC* "$RDS"
BACK AXC ++1,J
XEC* "$1CG"
AXC ++1,J
XEC* "$1BC"
NOP
AXC ++1,J
XEC* "$TEI"
NOP
LXO BSFILE-4,J
LXO BSFILE-3,J
LXO BSFILE-2,J
TRA 3,J
CHECK TXL BACK,1,J
LXO BSFILE-2,J
CLA ERR+1
STD 0
CLA* 1,J
LDQ* 2,J
ERR
TSX 8,J
TXI BACK,0,J
PZE BSFILE-2,0,J,ERR
ONEARG CLA BSFILE-2
ADD =01000000
STD BSFILE-2
LXO CHECK,1
TRA BT1,J
A07000 OCT 7000
END
REM SKFILE(I,J) SKIPS TAPE I OVER J EOF MARKS.
REM
ENTRY  SKFILE
PIZ
PIE

SKFILE SIZD =-3.1
SIZD =-3.2
SIZD =-3.3
TSX $RER),) CHECK LAST READ
TEFA #1
TEFB +1
LCX SKFILE,-1,4
CLA* 1,4
TSX $(105),4 LOAD TAPE ADDRESSES
LDX SKFILE,-1,4
CALL 2,4
IS THERE A SECOND ARGUMENT
ANA =07777777700000
ERA =00077700000000
TNZ ONEARG
NO SECOND ARGUMENT
GODO CLA* 2,4
PRINT SECOND ARGUMENT
TZE BUMP+1
DID SOME DUMMY WANT NO FILES
LOOP SUB =01000000
RDS REC* $RDS! READ THE TAPE
TNGA =
TCOR =
TEFA BUMP-W DID WE HIT
TEFB BUMP AN END OF FILE
TRA RDS+1
GO READ SOME MORE
BUMP TNZ LOOP
LCX SKFILE,-3.1
LCX SKFILE,-2,2
LDX SKFILE,-1,4
NCP
TRCA ++1
TRCB ++1
TRA 3,4
ONEARG (CLA SKFILE-1
ADD =01000000
SET UP XR4 FOR PROPER RETURN
STO SKFILE-1
PKD 0,0
SET UP FOR ONE FILE
TRA RDS
END

COUNT 1200
REM INPUT ROUTINE USING ARITHMETIC STATEMENTS. CF NASA TN D-1092
00020
LIB INPUT.6
00030
ENTRY INPUT
00040
REM THIS IS SUBROUTINE INPUT. ITS CALLING SEQUENCE
00050
REM CONTAINS THREE ARGUMENTS--AN IDENTIFICATION NUMBER,
00060
REM CODE NUMBER, THE FIRST LOCATION RELATIVE TO WHICH
00070
REM ALL DATA IS TO BE LOADED, AND THE FIRST LOCATION
00080
REM OF A TABLE TO BE USED BY THE ROUTINE.
00090
REM
00100
REM
00110
REM INCLUDED IN THIS ASSEMBLY ARE SUBROUTINES
00120
REM 1 INPUT
00130
REM 2 CHARTR
00140
REM 3 CLEAR
00150
REM 4 LOOPAR
00160
REM 5 ERROR
00170
REM 6 LOCK
00180
REM 7 NAME
00190
REM 8 NUMBER
00200
REM 9 STORE
00210
REM 10 TABLE
00220
REM 11 TEST
00230
REM 12 ACCUM, FIX, FLT, BINARY
00240
REM 13 PRINT
00250
REM 14 READ
00260
REM
00270
REM INTAPE PIZ 0,7 LEWIS INPUT TAPE HOT STD.
00280
OUTAPE PIZ 0,6 FORTRAN STANDARD OUTPUT TAPE
00290
INDX PIZ 0,6 STORAGE FOR IRA
00300
PIE IRB
00310
PIE IRC
00320
BG1 1,INPUT
00330
INPUT SIZD INOX,1 SAVE INDEX REGISTER A.
00340
SIZD INOX,2 SAVE INDEX REGISTER B.
00350
SIZD INOX+2,4 SAVE INDEX REGISTER C.
00360
NIZT 1,4 IF THE IDENTIFICATION NUMBER IS 2
00370
TRA 4,4 RETURN TO THE CALLING PROGRAM.
00380
CLA =1353
00390
ADD 2,4 2,4 IS THE BASE LOCATION.
00400
STA SET
00410
STA LCL1
00420
STA LCL4
00430
CLA TSKBS OPEN BACKSPACE GATE
00440
STO* 3(LINK) CALL CHAIN WILL BACKSPACE
00450
LOCA 1,4
00460
STA NREG1 1,4 IS THE IDENTIFICATION NUMBER.
00470
ANT 36,1 INITIALIZE 36
00480
STZ 1,1,1 LOCATIONS
00490
TIX -1,1 TO ZERO.
00490
STD ILOC1 MAKE NON-TERAD.
00500
CLA 3,4 3,4 IS THE LOCATION OF THE TABLE.
00510
STA LOCFC PREPARE
00520
STA NREG1-1
00530
ADD =1353 THE
00540
STA LOGPA ARGUMENT STORAGE.
00550
STA LOCKL
00560
TSX CLEAR,4 CLEAR THE VAR REGION.
LOGAM TSX ACCUM,4 ACCUMULATE RESULTS IN ACC.
TSX CLRALL,4
LXA WORD,4 * WORD IN ACC FOR LOGAR
PAA 0,4 NOT WORD IN ACC FOR LOGAR
TAL LOGAR,+,58 NOT COMMA
TAH LOGAR,+,59 NOT COMMA
LKD JK1,2 COMMA
CLA ACC
STZ ACC
LOC ACC
LQ ILOC1 IS THIS VARIABLE FIXED POINT.
TOP LQ1 NEGATIVE IS FIXED POINT.
TSX Fix,4
LOC1 STD **,2 STORE THE NUMBER RELATIVE TO BASE.
LOGAN TGU **,1,2,1 RAISE STORING INDEX BY ONE.
LOGANZ SOD JK1,2 SAVE IT.
LKD OPER,1 OPERATORS LEFT OVER.
TSL **,3,1,0 ANY OPERATORS LEFT OVER.
ERRL TSL ERROR,4
BCI 1,10011 ANY DATA LEFT OVER.
CLA ACC
REM CALL THIS THE SWITCH HOUSE.
LOGAD TSX CLEAR,4
LOGAP TSX CHMKTR,4
LOGAG TSX COMPAR,4
BCI 1,00000
TRA **,2,2,2
TRA LOGAR $D, $T, OR OPERATORS.
TRA LOGAL ALPHABETIC
TRA LOHAL NUMERIC
TRA LOCAT COMMA
TRA LOGC DECIMAL
LOGAR LKD OPER,1 ANY OPERATORS LEFT OVER.
ERRL 1,0 HIGH MEANS ALREADY HAS OPERATOR.
SUB 00000000 SPLIT OFF & FROM OTHERS
TPL LQ1
REM WHAT KIND OF OPERATOR IS THIS.
TSX COMPAR,4
BCI 1,50000
TRA ERRL,5,3,5 REMOVE THE JUNK.
TRA LOGAR,4,5 COMMA
SRO OPER,2,4 SAVE REST, WILL BRANCH IN SUB ACCU
TRA LOGAP AFTER BOTH OPERANDS HAVE BEEN FOUND
REM
REM COMES HERE IF THE OCT OR ALF MODE.
LOCAT TSX CHMKTR,4
TSX COMPAR,4
BCI 1,50000
TRA **,5,2
TRA ERL JUNK
TRA LOGAR A CHARACTER
TRA LOGAU Q CHARACTER
REM
REM COMES HERE IF EMPTY PARENTHESIS WERE FOUND.
TSX CHMKTR,4
TSX COMPAR,4
TOP **,7,0
TRA ERRL,5,3,5 INSERT COMMA IF NEEDED.
CLA ILOC1
STD ILOC PREPARE TO GET VALUE OF
LIQ ILOC
LKD JK1,2 CURRENT LEFT SIDE.
TRA SET
REM COMES HERE IF OCTAL MODE.
LOCAU TSX CHMKTR,4
SUB 00000000
TNA LOGAU
TRA LOGC,4,7,1 SIGN
LOCAL LQO VAR
RUL 3 REPLACE TOP 3 BITS
LRG 3 BY NEXT OCTAL CHARACTER
RUL 3 PUT IN BOTTOM OF AQ
STQ VAR
REM COMES HERE WHEN J IS FOUND.
LOCAW TSX CHMKTR,4
TOP **,2
TRA ERRL,5,3,5
TRA LOGC,4,7,1 CHARACTER TO AQ
LCX LOGAL,4 OCTAL DIGITS
LKD ERRJ,4,5,5 SPLIT ITS
CLA VAR
TRA LOGC
REM COMES HERE IF THE OCT OR ALF MODE.
LOCAY TSX BINARY,4
REM
REM COMES HERE IF ALF MODE.
LOCAZ TSX CHMKTR,4
TSX COMPAR,4
BCI 1,00000
TRA **,5,2,2
TRA ERRL JUNK
TRA LOGC3 ALPHABETIC
TRA LOGO3 NUMERIC
REM COMES HERE WHEN J IS FOUND.
LOCBA LXA VAR.1,1 I SIGN 02540
TXN ERRK.1,0 ALF COUNT WAS ZERO. 02550
SKD ALF.1 02560
TSK CLEAR.1 02570
TSX CHCTR.1,4 PULL THROUGH CHARACTERS AND STORE 02580
SUB =O17 FILE FLAG, NEVER NEG. 02590
TZE ERK COUNT WENT PAST E O JOB. 02600
TSK STONE.1,4 ROOM ONE AT A TIME. 02610
TIX ++=1,1 GO BACK TILL NCHAR = 1 02620
LXD J1 02630
LXD MSHIFT.4 02640
CAL BLANK 02650
LGR +1,4 02660
DRS VAR.1,1 FILL IN PARTIAL WORD WITH BLANKS. 02670
LOCBB ART 1,1 IAC TO 1 02680
LXD J1K 02690
CLA J PREPARE TO STORE ALF WOS 02700
STD LOLID 02710
LOCBC SKD J1K.1 02720
CLA VAR.1,4 02730
LOC4 STD ++,2 02740
TXI ++=1,1 J = J + 1 02750
LOCC1 TXI LOLID.2,1 *J*K=J+1 02770
LOCCD STZ ALE 02780
TSX CLEAR.4 LOOK AT NEXT CHARACTER. 02790
TOP MINUS FOR NEW CARD 02800
TSX TEST.1,4 PUT IN COMMA IF NEEDED. 02820
SUB =HUGDOO, GU RAISE AND STORE JK1. 02830
TZE LOLID GO RAISE AND STORE JK1. 02840
REM 02850
REM THESE ARE ERROR CALLS 02860
ERRB TXS ERROR.4 02870
BCI 1,01J 02880
ERRD TXS ERROR.4 02890
BCI 1,01J 02900
ERRJ TXS ERROR.4 02910
BCI 1,01J 02920
ERRK TXS ERROR.4 02930
BCI 1,01J 02940
ERRM TXS ERROR.4 02950
BCI 1,01M 02960
ERRU TXS ERROR.4 02970
BCI 1,01J 02980
REM 02990
REM 41 COMES HERE AFTER $0 03000
REM 03010
LOCBG CLA =U76100000000 NOP 03020
STD $1(LINK) CLOSE BACKSPACE GATE 03030
TRA LOLID RETURN 03040
REM PURPOSE OF SEND CARU IS TO PROTECT FORIEGN DATA FROM 03050
REM BACKSPACE WHEN CHAIN IS CALLED. 03060
REM 03070
REM ENU OF THE MAIN SEGMENT 03080
REM THIS ROUTINE TOS BACKSPACE THE INPUT TAPE WHEN A 03090
REM CALL CHAIN IS GOING TO SPILL THE BUFFER. 03090
REM THIS ROUTINE IS EXECUTED FROM CHAIN VIA THE ONE 03100
REM NAME SUBROUTINE (LINK) WHICH CONTAINS EITHER TSX OR NOP 03110
REM 03120
TSXBS TXS LOLID.4 TO BE STORED AT (LINK) 03130
LOCBS SKX ALE 03140
TXA ++,4 SAVE INDEX 4 03150
CLA INTAPE INPUT TAPE NUMBER 03160
CALL $1050 SELECT INPUT TAPE 03170
XEC $108R BACKSPACE IT 03180
AXT ++,6 RETURN TO THE CHAIN ROUTINE 03190
TRA 1,4 03200
EJECT 03210
REM THIS IS SUBROUTINE CHCTR. IT STORES SUCCESSIVE 03220
REM CHARACTERS FROM THE CARD AT LOCATION WORD, READS 03230
REM SUCCESSIVE CARDS INTO THE ARRAY RECORD, AND PRINTS 03240
REM $ TYPE CARDS, THE FIRST CHARACTER FROM A NEW CARD 03250
REM IS STORED IN WORD WITH A MINUS SIGN. 03260
REM 03270
REM 03280
CHCTR SKD TEMP-10,2 03290
SKD TEMP-17,6 03300
LXD I$2 CARO COL COUNT, SAM COUNT 03310
THX **3,2,83 TOURS ONE TO READ. 03320
THX **2,83 TUG EARLY TO READ. 03330
XEC READ. GATE MAY BE CLOSED 03340
LQD Q HAS UNUSED CHARACTERS FROM BEFORE 03350
CLA S1M ZERO OR $ GOES TO TAG 03360
LOC2A AL5 $ SHIFl LEFT 1 CHARACTER 03370
SLM G3 CLEARS OR PRELOADS TAG 03380
LOC2B LXD ALF.4 NONZERO MEANS ALF MODE. 03390
LOC2C TXH LOLID.2,49 SAW COUNT GIVES COL.81 = 49. 03310
LOC2D LOLID.2,48 WAS COL.80 PROCESSED. 03320
LOC2E PXD 0,0 CLEAR ACCUMULATOR. 03330
LQD 6 SHIFT NEXT CHARACTER INTO ACC. 03340
LTD LOLID.2,14 COUNT DOWN BY 1 03350
LTD RECORDS,3,2 LOAD NEXT WORD 03360
TRI ++,1,89 JUMP BACK COUNTER. 03370
LOC2E TXH LOLID.4,0 RETURN IF ALF MODE. 03380
PAK 0,1 MOVE CHR. INTO INDEX 1 03390
THX LOLID.1,40 TAK MEANS GOOD CHARACTER. 03400
THX LOLID.1,47 TAK IF BLANK 03410
THX LOLID.1,48 TAK IF GOOD CHARACTER. 03420
TXL LOLID.1,42 TAK IF GOOD CHARACTER. 03430
ZET TAG HERE ON $ 03440
TRA PRINT HERE ON $ GO PRINT 03450
TRA LOLID $ GOES TO TAG. 03460
REM 03500
LOCDO CLA 2,4 USE SECOND ARGUMENT IN THE CALLING
PDX 0,1 SEQUENCE DECREMENT AS THE TEST
TXN LOCDC,1,1024 FOR ALPHABETIC-NUMERIC SPLIT.
SKD LOLDP,12 BECOMES INCREMEN
LX A WORD,1 CHARACTER TO IRA
TX L LOLDP,1,9 NUMERIC
TAH LOLDP,1,57 SPECIAL 0 ZONE, X
THX LOLDP,1,49 ALPHABETIC 0 ZONE, N
TIX **1,1,132 KNOCK OFF 11 ZONE EXCEPT ->
TIX **1,1,16 KNOCK OFF 12 ZONE EXCEPT +
REM - AND - SIGNS WILL BE 111110, WILL BE 111110.
THX LOLDP,1,9 SPECIAL
LOCDFI TXI LOLDP,2,1** ADJUST IRB FOR ALPHABETIC
LOCDFI TXI LOLDP,2,** ADJUST IRB FOR SPLIT
REM END OF THE SAP SUBROUTINE COMP.
EJECT
REM THIS IS SUBROUTINE ERROR. IT IS CALLED IF AN
REM ERROR WAS DETECTED ON ANY OF THE INPUT CARDS.
REM
ERROR SMA **4,4 SAVE SOURCE
XEC+ SITE3 WAIT FOR QUIET BUFFERS
ZEC+ **4 CLA 1,4 GET PRINT ARGUMENT
STU OUTBUF=5,4 ART 1,1
CAS 8
TRA **3 S THROUGH V
TXI **1,1,1 R
MESSA PKD BLANK+4,1 A THROUGH N
ANA =707
ARS 16
ACL MESA
STA 102.
ART 4,4
IOZ. LDO **4 STU OUTBUF=5,4
TIX =**1,4,1 ART 14,4
TDO RECORD+2,4 STU OUTBUF=49,4
TIX =*2,4,1 ART 14,4
TSX PRIN4+ XEC+ SITE3 WAIT FOR QUIET BUFFER
ART =1,2,4 CLA BLANK
STU OUTBUF=19,2
TIX =**1,2,1 ART 14,4
LDQ =*3,3,10 PICK UP *
LDO 1,2 SAVE COUNT
TXL =**2,7,1 BACK UP IF OVER 71
TRI =**3,3,2,69
RGL 6 ROTATE ACCORDING TO CHR PART.
TIX =**1,2,16 COUNT CHARACTER PART.
STO OUTBUF=19,2 STORE ACCORDING TO RESIDUAL
TIX =**1,1 ART 14,4 PRINT THE *
XEC+ SITE3 WAIT FOR THE * TO BE PRINTED
LXO ERSH+4,0 PICK UP ERROR SWITCH.
TXL LOLDP=1,4,0 NUN ZERO MEANS TRY NEXT SET
ART 120,4 BYPASS MARK
SKD BLANK+4,4 MARK BYPASSED CARDS
LX A NREGI+4 NONZERO IF THIS $DATA CARD.
TSX **1,4,0 RX
LXO READ+1,4 CRASH READ GATE IF $DATA CARD.
LOCED TST CMCHR+4,4 SKIP TO NEXT $DATA AND TRY THAT SET.
TST LODEM OTHER
SUB =HODOSO TRM NOT A $DATA CARD
TST T NREGI+1 PUTS - SIGN IN TABLE(@)
LXO BLANK+7,4
SKD BLANK+4,4
LDO INO+=2,4
TRA LODEM
LOECD ADD =5 TEST FOR END FILE FLAG
IZE SGNOUT END FILE.. GET OFF
TRA LODEM OTHER
REM ERROR MESSAGES. FIRST WORD ALSO USED AS A BLANK.
BLANK BCI 4, REDUNDANCY CHECK
BCI 4, ILLEGAL CHARACTER
BCI 4, NO MANDISSA BEFORE E.
BCI 4, NO ENTRY IN TABLE
BCI 4, TYPE MISING OR WRONG
BCI 4, EXPRN OUT OF RANGE
REM END OF THE SAP SUBROUTINE ERROR.
EJECT
REM THIS IS SUBROUTINE LOOK, IT SEARCHES THE TABLE
REM FOR THE NAME STORED AT LOCATION VAR. IF FOUND,
REM THE ACC IS NON-ZERO AT THE RETURN.
REM
LOOK SKD TEMP=12,4 SAVE INDEX REGISTER C.
CLA J SUBROUTINE.
STD LODEM OTHER
ART 1,6 JK = 2 IN INDEX B
AKT 1,1 JI = 1 IN INDEX A
LOCFA CAL **2,2 CALL TABLE12.
TZE LOLDP NO ENTRY THIS VARIABLE
STD LOLDP DECREMENT HAS NEXT
ACL =037777000000
ANA =037777770000 ENTRY LOC. SAVE DEC
SUB 1,2=1 ONLY, CHECK ENTRY LENGTH.
TNZ LOCFO IF NOT THE SAME, LOOK AT NEXT ENTR
PDX 0,2 JM = JK IN INDEX C.
POX 0,4 JM = JK IN INDEX C.
LOCFB CLA
VAR+1.1
SIE IF VAR AND THIS
05540
LOCFC
CAS
**+1," ENTRY AGREE
05550
TRA
**+2," IF SO, CHECK REST OF NAME
05560
TRI
**+2+1," RAISE JM BY ONE.
05570
LOCFD
TXI
LOCF=1-2," IF NOT SO, GO TO NEXT ENTRY.
05580
**+1+1," RAISE J1 BY ONE.
05590
LOCFE
TXL
LOLFB=1-2," FINISHED IF J1 IS GREATER THAN J.
05600
TXR
05610
LOCFF CLA
LUCFA
STD
ILUC
SAVE COMMON INDEX AT ILOC.
05620
LOCFG
XD
TEMP-12.4
PREPARE TO RETURN.
05630
TRA
1.4
RETURN TO THE CALLING PROGRAM.
05650
REM
05660
REM END OF THE SAP SUBROUTINE LOOK.
05680
EJECT
05690
REM THIS IS SUBROUTINE NAME. IT IS USED TO
05700
REM MEMORY LOCATIONS BY REFERRING TO THE TABLE.
05720
REM
05730
REM
05740
NAME
SXD
TEMP-20.4
SAVE INDEX C.
05750
REM GET THE REST OF THE VARIABLE NAME. STOP AT ANY
05760
REM NON ALPHANUMERIC CHARACTER.
05770
LOCGB
TSX
STORE-4
05780
LOCGC
TSX
CHRCTR-4
05790
TOP
**+2," MINUS FOR NEW CARD
05800
TOP
TEST+4 " COMMA MAY BE NEEDED.
05810
YNZ
**+3," LOOK FOR ZERO. IF ZERO, MAKE IT
05820
ACL
=NOODDDDD
A LETTER D
05830
STD
WORD
05840
LOCDE
TSX
COMPAR-4
05850
BCI
1.-10000
05860
TRA
**+2.2.1
05870
TRA
LOLCF
JUNK OR OPERATORS
05880
TRA
LOLCB
NUMERIC OR ALPHABETIC
05890
TRA
LOLGC
I SIGN
05900
STZ
ILUC
* SIGN
05910
REM GO TO THE TABLE LOOKUP ROUTINE IF AN = SIGN
05920
REM OR AN OPERATOR WAS FOUND.
05930
LOCDF
TSX
LOOK+4 " FIND THE NAME IN TABLE.
05940
TZE
ERAT
NAME WAS FOUND IN TABLE IF NON-ZER
05950
TRA
ILUC
2
05960
TRA
LOLGL
05970
REM
05980
REM GO TO THE TABLE VARIABLE LOOKUP ROUTINE IF A
05990
REM I SIGN WAS FOUND.
06000
LOCGG
TSX
LUCHA
06010
YNZ
LOLGC
06020
ERRT
TSX
ERROR-4
06030
BCI
1.0(0)
06040
REM CONVSTANT AND ERROR CALL.
06050
LOCGH
TSX
BINARY-4
06060
REM GET THE NUMERICS FOR THE INDEX TO THE VARIABLE.
06070
LOCGI
TSX
CHRCTR-4
06080
TAL
LUGW+1.9
NUMERIC
06090
TAL
ERKC+1.27
JUNK
06100
TAL
ERKC+1.29
JUNK
06110
TSX
CHRCTR+4 " I SIGN. GET NEXT CHARACTER.
06120
TOP
**+4," MINUS FOR NEW CARD
06130
TOP
TEST+4 " COMMA MAY BE NEEDED.
06140
TEX
COMPAR+4
06150
BCI
1.+20000
06160
TRA
**+4.2.1
06170
TRA
LOLGC
OPERATORS
06180
TRA
ERROR
ALPHABETIC AND NUMERIC
06190
STZ
ILUC
* SIGN
06200
REM
06210
LOCGL
CLA
VAR
COMPUTE STORING INDEX.
06220
ACL
ILUC
06230
PAX
0.2
STORE ADDRESS AT DECREMENT WITHOUT
06240
TXI
**1.2-1," I SIGN.
06250
LOCGL
SKD
JN+2
ACCUMULATOR OVERFLOW.
06260
CLA
ILUC
06270
TLX
TEMP-20.4
RESTORE INDEX C.
06280
TRA
1.4
RETURN TO CALLING PROGRAM.
06290
REM
06300
ERRC
TSX
ERROR-4
06310
BCI
1.0(0)
06320
REM
06330
REM END OF THE SAP SUBROUTINE NAME.
06340
EJECT
06350
REM THIS IS SUBROUTINE NUMBER. IT IS USED TO
06360
REM ASSEMBLE NUMERIC DATA FROM CARDS. ALL VALUES ARE
06370
REM TREATED AS FLOATING POINT NUMBERS IN THIS ROUTINE.
06380
REM
06390
NUMBER
SKD
TEMP-23.4
SAVE INDEX C.
06400
SKD
KNT2.4 INITIALIZE
06410
STZ
KNT3
THE SUBROUTINE
06420
STZ
KNT1
BRANCH PARAMETERS.
06430
STZ
KNT4
06440
STZ
TEMP
06450
TRA
LOLGB
06460
REM
06470
LUCHA
TSX
CHRCTR+4
06480
TOP
**+2," MINUS MEANS FROM NEW CARD
06490
TSX
TEST+4 " RETURN TO THE CALLING PROGRAM.
06500
LOCHW TXS COMPAR, 4
BEX 1.8E0000
TAA **E2.2X
TRA LOCHX JUNK OR AN OPERATOR
TEX ERRF ALPHABETIC
TRA LOCHC NUMERIC
TRA LDLNE E
CLA KN12 DECIMAL POINT.
TNZ **=1 ZERO MEANS THIS IS THE SECOND POIN
TEX ERROR, 4
BCI 1x0(N)
STE KN12
STZ NEPX
TRA LOCHA
LOCHC CLA NEPX COUNT THE NUMBER OF DIGITS BEHIND
ADD =1035 THE IF THERE IS ONE
STO NEPX.
LOCHD LAA KN11,1 DO NOT ACCUMULATE PAST 10
TXH LOCHD2,1,10
LOCHDI TXS BINARY, 4
ZTX LOCHA DO NOT COUNT LEADING ZEROS.
ZTX LOCHA
LOCHD2 TXS **11,1 COUNT TOTAL NO. OF DIGITS
STO
TRA LOCHA
REM COMES HERE WHEN THE EXPONENT FIELD IS
LOCHC CLA KN1 ENCOUNTERED.
TNZ LOCHH THERE MUST BE AT LEAST ONE DIGIT
TEX ERRK, 4 BEFORE THE E OF AN E FORMAT NUMBER
BCI 1x-00D
STO
TRA **-2
TRA LOCHC-2 OTHERS
TRA ERRF ALPHABETIC
TRA LOCHJ NUMERIC
TRA ERRF DECIMAL
TRA LOCHG MINUS
TRA LOCHH PLUS
STO
REM CONVERT THE EXPONENT TO BINARY.
LOCHJ CLA TEMP
ALS 2
ADD TEMP
ALS 1
ACL WORD
STO TEMP
SXX KN13, 2 RECORD FACT FOR SECOND SIGN.
TRA LOCHH
REM COMES HERE WHEN AN OPERATOR WAS FOUND.
CLA KN13 TEST FOR THE PRESENCE OF EXPONENT.
TZE ERRF ZERO MEANS NO EXPONENT CAME.
LOCHC CLA KN12
TZE **2
STZ NEPX
CLA KN11 SEE IF MORE THAN TEN NUMBERS HAVE
SUB +10835 BEEN CONVERTED
TPL ** 2 IF SO, USE THE DIFFERENCE IN THE
PAS 0,0 COMPUTATION OF THE EXPONENT.
SUB NEPX
ADD TEMP
STO NEPX
REM MANTISSA IN VAR AND THE EXPONENT IS IN NEKP.
CLA VAR
TZE LOCHG SHORT CUT IF ZERO.
LDC &233000000000 CHARACTERISTIC FOR LOW BITS
LGR 8 LOW B BITS TO MQ
RQL 8
RQS 0 BRING SIGN
STA VAR
ORA &233000000000 CHARACTERISTIC FOR HIGH BITS
FAD VAR
FAN
STO
VAR
CLA NEKP THE EXPONENT
CAL 1/2
LDC 1x1 PUT L IN MQ
LOCHJ LBT EXPONENT IN ACCU
TRA LOCHM FOUND NO BIT.
TAE ERRV, 2,6 EXPONENT EXCEEDS 63
STO VAR-2 THIS FORMS 10 **NEXP
FNP TAD+1,2 SAVE IN MQ
XCA VAR-2
CLA VAR-2
LOCHM AAS 1
TZE LOCHM 10 **NEXP FINISHED.
TRI LOCHM2, 2,1
LOCHH TRI LOCHO MULTIPLY IF PLUS.
FNP VAR
FNN
TRA LOCHO
LOCHG STQ VAR-2 DIVIDE IF EXPONENT IS MINUS.
PDP VAR-2
XCA
LOCHQ LXD TEMP-23, 4 RESTORE INDEX CODE
TRA 1,4 RETURN TO CALLING PROGRAM.
REM THESE ARE THE ERROR CALLS FOR SUB NUMER.
ERRF TIX ERROR, 4 07580
RECF TIX ERROR, 4 07590
ERRV TIX ERROR, 4 07600
BCI 1, 0(1) 07610
BCI 1, 0(1) 07620
BCI 1, 0(1) 07630
REM 07640
REM THIS IS THE FLOATING PT. TABLE USED IN DEO 07650
DEC 
1E+32.1E+16.1E+8.1E+4.1E+2 07660
CONVERSION TABLE
DEC 
1E+1
REM
07670
REM END OF THE SAP SUBROUTINE NUMBER.
07680
EJECT
REM
07690
REM
07700
REM THIS IS SUBROUTINE STORE. IT STORES CHARACTERS
REM AT THE ARRAY VAR. 07710
REM
07720
STORE
SXD 0.13D 07730
SXD 0.14D 07740
REM
07750
LXD J, 1 07760
PUT J INTO INDEX REGISTER A.
LXD MSHIFT, 2 07770
LOAD INDEX B WITH MSHIFT.
TIX LSHIFT, 2 07780
ADVANCE MSHIFT.
REM
07790
XJX **+1, 1, 1 07800
SAVE INDEX A.
REM
07810
STZ VAR, 1 07820
CLEAR NEXT CELL
LOCJ CLA WORD 07830
LEAVE SIGN BEHIND.
LOQ +0 07840
REM
07850
STM TEMP, 7 07860
PLACE THE LEFT.
CAL TEMP, 7 07870
REM
07880
ORS VAR+1, 1 07890
STORE THE CHARACTER AT VAR.
SXD MSHIFT, 2 07890
SAVE MSHIFT.
REM
07900
LXD J, 1 07910
SAVE J.
LXD TEMP, 7 07920
RESTORE INDEX A.
REM
07930
TRA 1, 4 07940
RETURN TO CALLING PROGRAM.
REM
07950
REM END OF THE SAP SUBROUTINE STORE.
REM
07960
RETM 07970
REM CONSTRUCT A TABLE OF NAMES TO BE USED ON CARDS
REM AND THEIR MEMORY LOCATIONS RELATIVE TO ANG 2 OF
REM THE CALLING SEQUENCE.
REM
07980
TABLE
SXD 0.15D 07990
SAVE INDEX C.
STM TEMP 08000
REM
08010
LOCKA TIX CMCTR, 4 08020
STM TEMP 08030
REM
08040
TSX CMCTR, 4 08050
TSX COMPR, 4 08060
BCI 1, 0(0000) 08070
TRA **+2, 2, 2 08080
TRA LOCKD+1 08090
TRA LOCKA 08100
TRA LOCKC 08110
TRA LOCKD 08120
REM COMES HERE TO CONVERT THE ADDRESS TO OLTAL FOR
LOCKA CLA TEMP 08130
THE TABLE.
REM
08140
ALS 2 08150
ADD TEMP 08160
REM
08170
ALS 1 08180
AIC WORD 08190
REM
08200
STD TEMP 08210
ADDS TO MAGNITUDE
REM
08220
LOCKD TIX CMCTR, 4 08230
TSX COMPR, 4 08240
BCI 1, -7000 08250
TRA **+7, 2, 2 08260
TRA ERRA 08270
TRA ERRA 08280
TRA LOCKC 08290
TRA LOCKD 08300
TRA LOCKF 08310
TRA LOCKG 08320
REM
08330
REM COMES HERE TO GET NUMERICS.
REM
08340
REM COMES HERE TO GET NUMERICS.
REM
08350
REM COMES HERE TO GET NUMERICS.
REM
08360
REM COMES HERE TO GET NUMERICS.
REM
08370
REM COMES HERE TO GET NUMERICS.
REM
08380
REM COMES HERE TO GET NUMERICS.
REM
08390
REM COMES HERE TO GET NUMERICS.
REM
08400
REM COMES HERE TO GET NUMERICS.
REM
08410
REM COMES HERE TO GET NUMERICS.
REM
08420
REM COMES HERE TO GET NUMERICS.
REM
08430
REM COMES HERE TO GET NUMERICS.
REM
08440
REM COMES HERE TO GET NUMERICS.
REM
08450
REM COMES HERE TO GET NUMERICS.
REM
08460
REM COMES HERE TO GET NUMERICS.
REM
08470
REM COMES HERE TO GET NUMERICS.
REM
08480
REM COMES HERE TO GET NUMERICS.
REM
08490
REM COMES HERE TO GET NUMERICS.
REM
08500
REM COMES HERE TO GET NUMERICS.
REM
08510
REM COMES HERE TO GET NUMERICS.
REM
08520
REM COMES HERE TO STORE CHARACTER.
REM
08530
REM COMES HERE TO STORE CHARACTER.
REM
08540
REM COMES HERE TO STORE CHARACTER.
REM
08550
REM COMES HERE TO STORE CHARACTER.
REM
08560
REM COMES HERE TO STORE CHARACTER.
REM
08570
REM COMES HERE TO STORE CHARACTER.
REM
08580
REM COMES HERE TO STORE CHARACTER.
LOCK
TSX
LOCK
LAX
J,1
TEX
***1,1
PXD
0,1
ACL
TEMP
LOCRL
SLW
***2
STO
***1
TXI
***1,2**
LOCK
CAL
VAR+1,1
LOCK
XEC
LOCAL
SWE
***2,1
TRANSFER WHEN DONE
TXI
**1
LOCK+\=1
LOCK+\=2
GO BACK TO FINISH
ARS
3%
KEEP ZONE OF 1ST VAR CH.
TZE
ERRG
HAS NUMERIC, OR J=0
REM
8670
LOCK+\=3
REEXAMINE THE CUT OFF CHARACTER.
8676
LOCRP
LXD
Bz
TRH
LOCKB+2,1
COMMA
8677
LOCK
LXD
TEMP-15,X / CHARACTER
8780
TRA
1,4
RETURN.
8790
REM
REM
COMES HERE TO REPLACE KEY
8810
LOCK
ANA
#077777000000
*1 IN DECREMENT
8820
AEL
TEMP LOCATION AND SIGN
8830
LOCKS
LXDL
SLW IN TABLE
8840
TRA
LOCKP
8850
LOCRT
CAL
TEMP
IS / LEGAL
8860
TZE
LOCKJ
YES NO, NUMERIC WAITING
8870
REM
REM THESE ARE THE ERROR CALLS.
8900
ERRA
TSX
ERROR=0
8910
ERAG
TSX
ERROR=4
8920
BCL
1,0,143
8930
REM
REM END OF THE SAP SUBROUTINE TABLE
8950
EJECT
REM
REM THIS IS A SUBROUTINE TEST. IT LOOKS AHEAD TO CLASSIFY
8970
REM A NEW CARD. ACOMMA WILL BE PUT INTO THE CURRENT
8990
REM CHARACTER POSITION ONLY IF EITHER (1) THE NEXT
9000
REM CARD BEGINS WITH A $ SIGN FOLLOWED BY SOME OTHER
9010
REM CHARACTER OR (2) THE NEXT CARD BEGINS WITH AN
9020
REM ALPHABETIC AND AN + SIGN IS FOUND AND IT PRECEDES
9030
REM ALL + $ AND + CHARACTERS ON THAT CARD.
9040
REM
9050
REM
9060
TEST
SKD
TEMP-12,4
SAVE INDEX FOR RETURN.
9070
SUB
?000000 TEST FOR A $ SIGN.
9080
TPL
LOLL
POSITIVE MEANS $ SIGN.
9090
XEC
READ.
SAFE TO REFILL BUFFER
9100
TXI
LXL0+1,16
NUMBERS AND SPECIAL
9110
TX
***1,1,33
FIX 50 SLASH IS SPECIAL
9120
TX
***1,16
M0D OUT ZONE
9130
TXH
LXL0+1,9
SPECIALS
9140
REM
REM ALPHABETIC COME THRU.
9150
LOCRL
AXT
15,1
SCAN THE CARD.
9160
LEQ
RECORD+3,1
9170
TXI
***1,1,09
FOR CHARACTER COUNT
9180
TAL
LXL0+1,70
DONE IF WHOLE CARD SCANNED
9190
LOCRL
PXD
0,0
OK TO SEARCH 84 COLUMNS
9200
LOL
6
PAX
0,0
ZONE TO IRB
9210
ANA
-Z017
KEEP DIGIT
9220
SUB
-Z013
DIGIT PART OF + 1 CHAR.
9230
TZE
LOLL
CHECK ZONE
9240
TIX
LXL0+1,14
TRY NEXT CHARACTER
9250
TRA
LXL0+1
9260
LOCLF
TXH
LXL0+2,15
, $ NEED NO COMMA
9270
LOCLA
AXT
84,1
84 IS CARD COL 1
9280
SKD 1,1
RESET CHMCTR TO BEGIN CARD
9290
STO
Q
?000000
9300
CLA
RECORD+12
9310
STO
Q
?000000
9320
CLA
SUBSTITUTE A COMMA.
9330
STO
WH0D
9340
LOCLF
LXD
TEMP-12,4
CLA
WH0D
IN AC FOR SR NAME, TABLE
9350
TRA
1,4
RETURN TO THE CALLING PROGRAM.
9360
REM
REM END FOR TABLE SUB STATEMENTS
9370
REM
REM TEST FOR SUBROUTINE TABLE.
9380
TEST
SKD
TEMP-12,4
9390
AXT
84,4
IF NEXT CARD HAS VALID
9400
LOQ
RECORD+12
LEFT PART OF SUBSTATEMENT
9410
LOCRL
PXD
0,0
9420
LOL
6
PAX
0,0
9430
TXH
LXL0+2,48
0 ZONES EXCEPT BLANK
9440
TXH
LXL0+2,47
BLANK.
9450
TXH
LXL0+2,27
11 ZONES AND 1
9460
TXH
LXL0+2,26
12 ZONES AND 1
9470
TXH
LXL0+2,24
12 ZONES AND 0
9480
TXH
LXL0+2,22
12 ZONES AND 0
9490
TXH
LXL0+2,20
12 ZONES AND 0
9500
LOCRL
TIX
LXL0+4,14
NUMERIC AND BLANK
9510
LOQ
RECORD+3,4
9520
TXH
LXL0+4,1
9530
TXI
LXL0+7,10
9540
REM
REM END OF THE SAP SUBROUTINE TEST.
9550
EJECT
REM
REM THE FOLLOWING FOUR SUBROUTINES ARE USED TO
9560
REM CONVERT DECIMAL, DIGITS TO BINARY IN VAR.
9570
REM FIX FLOATING POINT NUMBERS, FLOAT FIXED POINT
9580
REM NUMBERS, AND FORM ARITHMETIC RESULTS IN THE
9590
REM PSEUDO ACCUMULATOR (ACCI) FOR EACH OPERATION
9600
REM ON A CARD.
9610
REM
BIN CLA VAR
ALS 2 DIGITS IN BINARY IN VAR.
ADD VAR
ALS 1
ACL WMD
STD VAR
TRA 1.4
REM

FLT CLA TEMP
CONVERT TO FLOATING POINT THE
LRS 16 CONTENTS OF THE STORAGE CALLED
ORA +0233000000000 TEMP.
FAO -0233000000000 TEMP.
STD 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

ACCUM LDX OPER,2 BRANCH FOR OPERATOR
STI OPER PREPARE FOR NEXT OPERATOR.
CLA TEMP
TRA +0.2
TRA LOCMB /
CAS 0 MINUS
FAD ACC PLUS
ACCUM STD ACC NONE
TRA 1.4
REM

LOCMB CLA ACC DIVIDE.
FDP TEMP
STD ACC
TRA 1.4
REM

LOCMB LDU ACC MULTIPY.
FNP TEMP
TRA ACCUM
REM

REM END OF THE SAP SUBROUTINES ACCUM, FIX, FLOAT.
EJECT
REM

REM SUBROUTINE PRINX DRAINS PRINT BUFFER TO LOGICAL TAPE
REM IN DECREMENT OF OUTAPE GIVEN IN DECREMENT OF OUTAPE

PRINX SKA PRINX,4 LOGICAL OUTPUT TAPE NUMBER
CLA OUTAPE
CALL (I05) 10130
AXC IODC,4
XEC = (IRES) 10140
XEC = (RCHL) 10150
PRA 0.4 10160
STA = (TRES) 10170
CLA TSATK 10180
STD = (TES) 10190
PRINX AXT +14 10200
TRA 1.4 10210
REM

I0DC IORT OUTBUF,19 EXECUTED FROM (ITES)
REOR READ,1.4 REOR. GATE INITIALLY OPEN
READ. TSX READ,1.4 READ. GATE INITIALLY OPEN
CLA =D076100000000 CLOSE READ GATE
STD +2 10280
SKA AXT,4 CLOSE OUT BUFFER
CLA =D076100000000 SAY BUFFER IS QUIET
STD = (TES) 10420
AXT 5.4 PRESET REDUNDANCY
SKA RTT,4 COUNT
REM

TSX = RTT,4
XEC = (TRES) 10440
XEC = (RCHL) 10450
XEC = (TRES) 10460
XEC = (RCHL) 10470
AXC RRT,4 10480
AXC RTT,4 10480
AXC RRT,4 10480
XEC = (TEF) 10550
TRA AXT RETURN 10510
RRT AXT +0.4 INTERROGATE COUNT 10530
TIX SKA +0.1 GIVE ANOTHER TRY 10590
AXT 1.4 CARD SURE BAD 10550
CLA IN0UP+1.4 SAVE IMAGE 10560
STD RECORD+2.4 10570
TIX +2.4+1 MAKE ERROR ROUTINE LOSIE 10580
AXT 8.4 IN INPUT BUFFER 10590
SXD 1.4 SKA ERROR,4 10600
AXT 1.4 SKA RTT,4 10610
AXC +0.4+1 SAVE COUNT 10620
AXC +0.4+1 TURN OFF EOF IND 10630
AXC = (RMS) 10640
AXC IODC,4 BACKSPACE. 10640
AXC = (RMS) RREAD 10670
AXC RTT,4 10680
TRA TSX+T

83
"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546