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

by Milton G. Kofsky 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

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $1.00
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 investi-
gated (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 perform-
ance 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 con-
tributors 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

\(g\) gravitational constant, 32.174 \(\text{ft/sec}^2\)

\(H'\) isentropic specific work based on total-pressure ratio, \(\text{ft-lb/}lb\)

2
Δh  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\(^{3/4}\)/(min)(sec\(^{1/2}\))
p  pressure, psia
Q  volume flow (based on exit conditions), cu ft/sec
Re  Reynolds number, \( w/\mu r_t \)
r  radius, ft
U  blade velocity, ft/sec
V  absolute gas velocity, ft/sec
V_j  \( \cdot \)ideal jet-speed corresponding to total- to static-pressure ratio across turbine, \( \left(\frac{2gJ}{\Delta h_{id}}\right)^{1/2} \), ft/sec
W  relative gas velocity, ft/sec
w  weight flow, lb/sec
α  absolute rotor exit gas flow angle measured from axial direction, deg
γ  ratio of specific heats
δ  ratio of inlet total pressure to U. S. standard sea-level pressure, \( p'_1/p^* \)
ε  function of \( γ \) 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, \( (V_{cr}, 1/V_{cr}^*)^2 \)
μ  gas viscosity, lb/(ft)(sec)
ν  blade-jet speed ratio (based on rotor inlet tip speed), \( U_t/V_j \)
Subscripts:
cr  condition corresponding to Mach number of unity
id  ideal
w  outer wall
II

1 station at turbine inlet
2 station at stator exit
3 station at turbine exit

Superscripts:
* absolute total state
* U. S. standard sea-level conditions (temperature equal to 518.67° R and pressure equal to 14.696 psia)

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, \( wc\sqrt{\frac{\theta_{\text{cr}}}{\delta}} \), lb/sec ............ 0.616
Equivalent specific work, \( \Delta h/\theta_{\text{cr}} \), Btu/lb ............... 11.9
Equivalent speed, \( N\sqrt{\frac{\theta}{\theta_{\text{cr}}}} \), rpm .................. 29 550
Equivalent total- to static-pressure ratio, \( p_1/p_3 \) ................ 1.540
Total to total efficiency, \( \eta_{\text{tot}} \) ............... 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 \( \text{Re} = \frac{w}{\mu r_t} \). 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. - 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 \( \dot{w} = \sqrt{\frac{\theta_{cr}}{\delta}} \) 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...
Figure 3. - Variation of weight flow with pressure ratio and stator throat area at equivalent design speed.

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. Comparison
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 effi-
ciency 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 contrib-
tuted 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 pres-
sure 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 rela-
tive 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\text{-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, \( ^\circ \)K

\( t \)  
time, sec

\( U \)  
gravitational potential

\( U_x, U_y, U_z \)  
x, y, z accelerations due to gravity, m/sec\(^2\)

\( V \)  
absolute velocity, m/sec

\( V' \)  
relative velocity, m/sec

\( v \)  
true anomaly, radians

\( X \)  
forces acting on object other than gravity, thrust, lift, drag, and perturbations due to perturbating bodies

\( x, y, z \)  
components of \( r \), m

\( \alpha \)  
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\(^2m_r\)

\( \rho \)  
atmospheric density, kg/m\(^3\)
\omega \quad \text{argument of pericenter, radians}

\Omega \quad \text{equatorial longitude of ascending node, radians}

Subscript:

0 \quad \text{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

$$\vec{V}' = \vec{V} - \vec{\omega} \times \vec{r}$$  \hspace{1cm} (B1)

In x,y,z component form,

$$V'_x = V_x + ay \quad \text{(B2a)}$$
$$V'_y = V_y - ax \quad \text{(B2b)}$$
$$V'_z = V_z \quad \text{(B2c)}$$

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

Thrust Resolution Along x,y,z Axes

The thrust direction is specified with respect to the relative velocity vector \( \vec{V}' \) by the angles \( \alpha \) 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 \( \vec{V}', \vec{A} \), and \( \vec{P} \) form an orthogonal set. Thus,

$$\vec{A} = \vec{r} \times \vec{V}' = \text{relative angular momentum per unit mass} \quad \text{(B3)}$$

$$\vec{P} = \vec{V}' \times \vec{A} \quad \text{(B4)}$$

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

$$\vec{F} \cdot \vec{V}' = FV' \cos \alpha \quad \text{(B5a)}$$
$$\vec{F} \cdot \vec{A} = FA \sin \alpha \sin \beta \quad \text{(B5b)}$$
$$\vec{F} \cdot \vec{P} = FP \sin \alpha \cos \beta \quad \text{(B5c)}$$

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

$$\vec{F} = \frac{F}{p^2} \left(V' \cos \alpha \vec{A} \times \vec{P} + A \sin \alpha \sin \beta \vec{P} \times \vec{V}' + P \sin \alpha \cos \beta \vec{P} \right) \quad \text{(B6)}$$
or, in \( x, y, z \) component form,

\[
F_x = \frac{F}{p^2} \left[ v' \cos \alpha (A_yP_z - A_zP_y) + A \sin \alpha \sin \beta (P_yV_z' - P_zV_y') + P \sin \alpha \cos \beta P_x \right]
\]

(B7a)

\[
F_y = \frac{F}{p^2} \left[ v' \cos \alpha (A_zP_x - A_xP_z) + A \sin \alpha \sin \beta (P_zV_x' - P_xV_z') + P \sin \alpha \cos \beta P_y \right]
\]

(B7b)

\[
F_z = \frac{F}{p^2} \left[ v' \cos \alpha (A_xP_y - A_yP_x) + A \sin \alpha \sin \beta (P_xV_y' - P_yV_x') + P \sin \alpha \cos \beta P_z \right]
\]

(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:

\[
\vec{L} \cdot \vec{v}' = 0 \tag{B9a}
\]

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

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

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

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

(B10)

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

\[
L_x = \frac{L}{p^2} \left[ A \sin \beta (P_yV_z' - P_zV_y') + P \cos \beta P_x \right] \tag{B11a}
\]

\[
L_y = \frac{L}{p^2} \left[ A \sin \beta (P_zV_x' - P_xV_z') + P \cos \beta P_y \right] \tag{B11b}
\]

\[
L_z = \frac{L}{p^2} \left[ A \sin \beta (P_xV_y' - P_yV_x') + P \cos \beta P_z \right] \tag{B11c}
\]
APPENDIX C

TRANSFORMATION EQUATIONS FROM ORBIT ELEMENTS TO RECTANGULAR COORDINATES

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

\[ x = r(\cos \Omega \cos u - \sin \Omega \sin u \cos i) \]  
\[ y = r(\sin \Omega \cos u + \cos \Omega \sin u \cos i) \]
\[ z = r(\sin u \sin i) \]

where

\[ r = \frac{p}{1 + e \cos v} \]

\[ u = \omega + v \]

and \( v \) can be obtained from

\[ \cos v = \frac{\cos E - e}{1 - e \cos E} \]

and

\[ M = E - e \sin E \]

The velocity components may be obtained by differentiating the position equations using the two-body relations \( \dot{u} = \dot{v} = \frac{\sqrt{\mu p}}{r^2} \) and \( \dot{r} = \sqrt{\frac{\mu}{p}} e \sin v \).
\[ \dot{x} = - \sqrt{\frac{\mu}{p}} (N \cos i \sin \Omega + Q \cos \Omega) \quad \text{(C3a)} \]

\[ \dot{y} = \sqrt{\frac{\mu}{p}} (N \cos i \cos \Omega - Q \sin \Omega) \quad \text{(C3b)} \]

\[ \dot{z} = \sqrt{\frac{\mu}{p}} (N \sin i) \quad \text{(C3c)} \]

where

\[ N = e \cos \omega + \cos u \quad \text{(C4a)} \]

\[ Q = e \sin \omega + \sin u \quad \text{(C4b)} \]
APPENDIX D

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

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

$$x_{l+1}^2 = x_2 - x_1 = \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (D1)$$

where

- $x$ = a dependent variable
- $x_1^2$ = increment in the dependent variable
- $h_2$ = increment in the independent variable $t$
- $k_1 = h_2 \dot{x}_2(t_1, x_1)$
- $k_2 = h_2 \dot{x}_2(t_1 + \frac{h_2}{2}, x_1 + \frac{k_1}{2})$
- $k_3 = h_2 \dot{x}_2(t_1 + \frac{h_2}{2}, x_1 + \frac{k_2}{2})$
- $k_4 = h_2 \dot{x}_2(t_1 + h_2, x_1 + k_3)$

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

$$\dot{x} = \dot{x}_0 \frac{(t - t_1)(t - t_2)}{h_1(h_1 + h_2)} - \dot{x}_1 \frac{(t - t_0)(t - t_2)}{h_1h_2} + \dot{x}_2 \frac{(t - t_0)(t - t_1)}{h_2(h_1 + h_2)} \quad (D2)$$

Integration of this equation from $t_1$ to $t_2$ yields

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

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

(D4)

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

$$\bar{\delta} = Ah^5$$

(D5a)

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

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

(D5b)

where

$$A' = \log A$$

(D6a)

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

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

(D6b)

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

$$A'_3 = A'_2 + \left( A'_2 - A'_1 \right) \frac{h_3}{h_2}$$

(D6c)

and on this basis $\delta_3$ would be predicted to be

$$\log \delta_3 = A'_3 + 5 \log h_3$$

(D7)

It is desired that $\delta_3$ should approximate $\bar{\delta}$, the reference error; therefore,

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

(D8)

Each dependent variable has an associated relative error and would lead to computation of a different step size for each variable; however, the maximum relative error of all variables may be selected for $\bar{\delta}$. 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

$$h_3 = \exp\left[\frac{1}{5} \left( \log \delta - A_2 \right) \right]$$  \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 $h_2 = h_1$,

$$X'_1 = \frac{h_1}{3} (x_0 + 4x_1 + x_2)$$  \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.
## APPENDIX E

### 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, M**2/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>ASYMPT</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
DELT(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
DIOFFJ 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(8), DEGREES

ELIPS (12,10) A(167) ELLIPSE DATA FOR PERTURBATING BODIES, READ FROM CARDS, 12 PIECES OF DATA PER BODY

EMONE B(28) ECCENTRICITY -1
END A(5) SEE TABLE II

EPAR B(26) SQUARE ROOT OF (ECCENTRICITY Squared -1)

EREF A(13) SEE TABLE II

ERLIMT A(14) SEE TABLE II

ERLOG B(17) NATURAL LOGARITHM OF EREF

ETOL A(30) SEE TABLE II

EXITA B(392) AEXIT(NSTAGE)/100, NEWTONS/MB

EXMODE B(27) ECCENTRICITY CALCULATED WHEN IMODE=3

FILE B(22) SEE TABLE II

FLOW1 (10) A(83) RATE OF PROPELLENT FLOW, KG/SEC

FLOW B(5) FLOW1(NSTAGE)

FORCE (3) B(66) X,Y,Z COMPONENTS OF THRUST ACCELERATION, M/SEC**2

GASFAC A(46) DEFINED IN SUBROUTINE AERO, SET IN STDATA

GEOH B(32) GEOPOTENTIAL, M

GK2M B(36) GRAVITATIONAL CONSTANT, MU, OF THE SYSTEM, M**3/SEC**2

GKM B(37) SQUARE ROOT OF GK2M

H2 B(15) VALUE OF DELT FOR PREVIOUS STEP

IBODY (8) B(177) DEFINED IN SUBROUTINE ORDER

ICC (10) A(153) SEE TABLE II

IDENT (10) A(123) INPUT IDENTIFICATION NUMBERS ASSOCIATED WITH EACH STAGE

IMODE A(1) SEE TABLE II

IND (3) A(60) SET OF INDICES, SET IN STDATA

INDERR B(51) NUMBER OF SETS OF ERROR DATA, SET IN ERROR2 FOR USE IN NBODY

28
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(8), DEGREES

LOOKX   A(8)    SEE TABLE II

LOOKSW  A(9)    SEE TABLE II

LSTAGE  A(38)   TOTAL NUMBER OF STAGES INTEGRATED BEFORE RETURNING TO THE MAIN PROGRAM

MBODYs  B(42)   NUMBER OF PERTURBATING BODIES (NBODYs−1)

MODOUT  A(20)   SEE TABLE II

NBODYs  B(41)   TOTAL NUMBER OF BODIES, EXCLUDING THE VEHICLE

NCASE   A(600)  SAVED VALUE OF NCASE

NCASE   C(1)    CASE NUMBER, RAISED ONCE EACH TIME CONTROL PASSES THROUGH THE MAIN PROGRAM

NEFMRs (8)  B(185)  DEFINED IN SUBROUTINE ORDER

NEQ      A(2)    NUMBER OF EQUATIONS TO BE INTEGRATED, SET TO 8 IN STDATA

NSAVE    C(4)    SEE TABLE II

NSTAGE   A(3)    THE INDEX INDICATING THE PARTICULAR STAGE CURRENTLY BEING INTEGRATED

NSTART   B(24)   INTERNAL CONTROL IN NBODY AND EQUATE

OBLATJ  A(26)   OBLATENESS COEFFICIENT OF SECOND HARMONIC

OBLATD  A(27)   OBLATENESS COEFFICIENT OF FOURTH HARMONIC

OBLATH  A(28)   OBLATENESS COEFFICIENT OF THIRD HARMONIC

OBLATN  A(40)   SEE TABLE II

OBLAT  (3)  B(75)   X,Y,Z COMPONENTS OF OBLATENESS ACCELERATION, M/SEC**2

OLDDEL   B(9)    VALUE OF DELT FOR PREVIOUS GOOD STEP

ORBELS  (6)  B(116)  ARRAY OF OUTPUT VARIABLES, EITHER RECTANGULAR OR ORBIT ELEMENTS

OUTPOT  B(399)  CAUSES ABSENCE OF OUTPUT WHEN NONZERO

P  (3)    B(84)    DEFINED IN EQ. (B4)
PAR (3)  B(60)  DEFINED BY EQUATIONS IN SUBROUTINE THRUST
PMAGN  B(50)  DEFINED IN EQUATION FORM BY SUBROUTINE THRUST
PNAME (30)  A(287)  PERMANENT LIST OF BODY NAMES MADE FROM PNAME 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(139)  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
REVOLUTION COUNTER, USED ONLY FOR OUTPUT

ROTATION RATE OF REFERENCE BODY WHEN ATMN=BNAME(1), RAD/SEC

INITIAL MASSES FOR AT MOST 10 STAGES, KG

ROTATION RATE OF A REFERENCE BODY, RAD/SEC

RADIUS SQUARED OF OBJECT TO ORIGIN, M**2

SEE TABLE II

SPECIFIC IMPULSES FOR AT MOST 10 STAGES, SEC

SIMP1(NSTAGE)

SINE OF ALPHA

SINE OF BETA

SINE OF TRU

SINE OF INCLINATION

SINE OF THE ARGUMENT OF LATITUDE

NUMBER OF EQUAL TIME UNITS UNTIL NEXT OUTPUT

SECONDS PER DAY, SET IN STDATA, SEC/DAY

GRAVITATIONAL CONSTANT OF THE SUN, AU**3/DAY**2

GRAVITATIONAL CONSTANT OF THE SUN, M**3/SEC**2

SEE TABLE II

SEE TABLE II

COUNT OF SUCCESSFUL INTEGRATION STEPS

COUNT OF UNSUCCESSFUL INTEGRATION STEPS (THOSE WHICH DO NOT PASS ERROR CONTROL TEST)

SEE TABLE II

TIME MEASURED RELATIVE TO THE JULIAN DATE OF TAKEOFF, DAYS

ARRAY OF INPUT PARAMETERS AND THEIR COMMON STORE LOCATIONS

SEE TABLE II

FLIGHT TIMES FOR AT MOST 10 STAGES, SEC
COEFFICIENTS FROM EPHEMERIDES TAPE TO BE USED IN DETERMINING POSITIONS AND POSSIBLY VELOCITIES OF PER-TURBATING BODIES, ONE SET FOR EACH OF 7 BODIES

ONE-HALF OF TIME SPACING BETWEEN TWO ADJACENT ENTRIES OF LIKE BODY NAME ON EPHEMERIDES TAPE, READ FROM TAPE FOR EACH BODY

TIME FOR SET OF EPHEMERIS DATA, READ FROM EPHEMERIDES TAPE, ONE FOR EACH BODY

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

ATMOSPHERIC TEMPERATURE TIMES THE RATIO OF MOLECULAR TO ACTUAL MOLECULAR WEIGHT, DEGREES KELVIN

SEE TABLE II

TRUE ANOMALY, RAD

SEE TABLE II

TIME TOLERANCE WITHIN WHICH PROBLEM TIME MINUS TMAX MUST LIE TO END STAGE

ECCENTRIC ANOMALY, RAD

VELOCITY OF OBJECT RELATIVE TO THE ORIGIN, M/SEC

X,Y,Z COMPONENTS OF THE RELATIVE VELOCITY, VQ, M/SEC

X,Y,Z COMPONENTS OF OBJECT VELOCITY RELATIVE TO ALL BODIES, M/SEC

INITIAL RELATIVE VELOCITY, USED WHEN IMODE=4, SKETCH (8), M/SEC

MACH NUMBER OF OBJECT

VELOCITY OF OBJECT RELATIVE TO ATMOSPHERE, M/SEC

SQUARE OF VQ, M**2/SEC**2

SQUARE OF V, M**2/SEC**2

X COMPONENT OF VELOCITY, M/SEC

Y COMPONENT OF VELOCITY, M/SEC

Z COMPONENT OF VELOCITY, M/SEC

WORKING SET OF INTEGRATION VARIABLES

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

LEWIS RESEARCH CENTER EPHEMERIS

General Description

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

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

Procedure Used to Fit Data

The process of computing the fitting equations is as follows:

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

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

\[
\dot{x} = (T_1 - T_{-1}) \left[ \frac{\Delta T_{-1} + \Delta T_{+1}}{2} - \frac{\Delta T_{+1} + \Delta T_{+3}}{12} + \frac{\Delta T_{-1} + \Delta T_{+1}}{60} \right] \left( \frac{\Delta v_{-1} + \Delta v_{+1}}{280} + \frac{\Delta v_{-1} + \Delta v_{+1}}{1260} \right)
\]

(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
\[ \dot{X} = \dot{X}_0 + 2C(T - T_0) + 3D(T - T_0)^2 + 4E(T - T_0)^3 + 5F(T - T_0)^4 \]
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>$\dot{x}$</td>
<td></td>
<td>$^a$All/day5</td>
</tr>
<tr>
<td>5</td>
<td>$\ddot{x}$</td>
<td></td>
<td>$^a$All/day4</td>
</tr>
<tr>
<td>6</td>
<td>$\dot{y}$</td>
<td></td>
<td>$^a$All/day3</td>
</tr>
<tr>
<td>7</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day2</td>
</tr>
<tr>
<td>8</td>
<td>$\dot{z}$</td>
<td></td>
<td>$^a$All/day</td>
</tr>
<tr>
<td>9</td>
<td>$\ddot{z}$</td>
<td></td>
<td>$^a$AU</td>
</tr>
<tr>
<td>10</td>
<td>$\ddot{x}$</td>
<td></td>
<td>$^a$All/day5</td>
</tr>
<tr>
<td>11</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day4</td>
</tr>
<tr>
<td>12</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day3</td>
</tr>
<tr>
<td>13</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day2</td>
</tr>
<tr>
<td>14</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day</td>
</tr>
<tr>
<td>15</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$AU</td>
</tr>
<tr>
<td>16</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day5</td>
</tr>
<tr>
<td>17</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day4</td>
</tr>
<tr>
<td>18</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day3</td>
</tr>
<tr>
<td>19</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day2</td>
</tr>
<tr>
<td>20</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$All/day</td>
</tr>
<tr>
<td>21</td>
<td>$\ddot{y}$</td>
<td></td>
<td>$^a$AU</td>
</tr>
</tbody>
</table>

$^a$Except for Moon data, which are in Earth radii and days.
(7) As soon as a set of coefficients was selected for an interval, addi-
tional 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 de-
scribed 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 in-
terval (i.e., only three points were fitted). Thus, a check of accuracy was not
available. The error in the attempt to fit the next greater interval (five
points) was not excessive, however, and it is judged that the accuracy obtained
from these fits is about equal to that held on the other bodies.

Merged Ephemeris Tape

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

The detail of this record is as follows:

- 1st word: FORTRAN compatible
- 2nd word: file number, fixed point in decrement
- 3rd word: planet name, code in BCD, first six characters
- 4th word: Julian date, floating point

Set 1

- etc., according to list in paragraph 6
- 21 words
- 23rd word: z
- 24th word: planet name, code in BCD, first six characters
- 25th word: Julian date, floating point

Set 2

- 44th word: z

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

Set 12 (last set)

- 234th word: planet name
- 235th word: Julian date, floating point
- 254th word: z
- End-of-record gap

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

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

INPUT-DATA REQUIREMENTS

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

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

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

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

Other cards: TAPE 3 = 0

TBEGIN = ephemeris beginning Julian date

TEND = ephemeris ending Julian date

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

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

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

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

As many similar sets as are needed may be appended.

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

(5) On the first execution after loading the routine, the common area is cleared whether an ephemeris tape is constructed or not. It is now necessary to load a table of variable names. Once loaded, this table will not be cleared again (except if the control variable TAPE 3 is set equal to zero). These names are for use on the input cards. If a different name is desirable for any
variable, it may be changed in the table and where it appears on the input card (ref. 7). The cards are:


(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

\[ \text{ROTATE} = \text{Earth rotation rate}, \ 7.29211585\times10^{-5} \ \text{radian/sec} \]

If integration in rectangular coordinates is desired set

\[ \text{IMODE} = 4 \]

or else if integration in orbit elements is desired set

\[ \text{IMODE} = -4 \]

Skip to instruction (9).

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

- **X** = x-component of position in \( x,y,z \) coordinate system, m
- **Y** = y-component of position in \( x,y,z \) coordinate system, m
Z = z-component of position in x,y,z coordinate system, m

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

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

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

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

Skip to instruction (9).

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

E = eccentricity

OMEGA = argument of pericenter, radians

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

INCL = orbit inclination to mean equator of 1950.0, radians

MA = mean anomaly, radians

P = semilatus rectum, m

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

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

DTOFFJ = Julian day number

TOFFT = fraction of day

TIME = time from previously set Julian date, sec

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

(10) To specify the origin and any perturbing bodies, list them as BODYCD = (list of body names in "ALF" format, see text example). The first body in the list is taken to be the reference body. The distances between the bodies in
this list must be computable from either ellipse data (instruction (11)) or ephemeris-tape data (instruction (2)). There may be no more than eight names in the list. Also, if the ephemeris tape is being used, the correct file must be found on it. For this purpose, set TFILE = desired ephemeris-tape file. The ephemeris files were numbered in sequence when written in instruction (2). If TFILE is not given, it will be set equal to 1.0 by the SUBROUTINE STDATA.

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

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

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

ELIPS = set 1, set 2, set 3, . . . , set n; n < 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:

\[ \text{TB} = \text{burning time for 1st stage, 2nd stage, etc., sec} \]
\[ \text{FLOW} = \text{propellant flow rate for 1st stage, 2nd stage, etc., kg/sec} \]
\[ \text{SIMP} = \text{vacuum specific impulse of 1st stage, 2nd stage, etc., sec} \]
\[ \text{AREA} = \text{aerodynamic reference area of 1st stage, 2nd stage, etc., m}^2 \]
\[ \text{AEXIT} = \text{engine exit area for 1st stage, 2nd stage, etc., m}^2 \]
\[ \text{RMASS} = \text{initial mass or jettison mass for 1st stage, 2nd stage, etc., kg} \]
\[ \text{DELT} = \text{initial integration step size for 1st stage, 2nd stage, etc., sec} \]
\[ \text{IDENT} = \text{input identification number 1st stage, 2nd stage, etc.} \]

\( \text{TB} \) must be loaded for as many stages as are to be flown. Others may be omitted if zero is appropriate. If \( \text{RMASS}(i) \) is not positive, the \( i \)th stage begins with the final mass of the previous stage reduced by the fixed amount \( \text{RMASS}(i) \). In the case of \( \text{DELT} \), zero will result in use of \( \text{TB}/100 \). \( \text{IDENT} \) 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 \( \text{AREA} \) in instruction (13):

\[ \text{ATMN} = \text{name of body that has atmosphere, in "ALF" format, (Earth)} \]
\[ \text{RATM} = \text{radius above which atmospheric forces are not to be considered, m} \]
\[ \text{ROTATE} = \text{atmospheric-rotation rate, radians/sec (7.29211585×10}^{-5} \text{ for Earth)} \]
\[ \text{BETA} = \text{angle } \beta, \text{ deg (see sketch (a))} \]
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:

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

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

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

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:

$$
\text{COEFN(J)} = \alpha \text{ coefficients}, C_L/\sin \alpha \text{ coefficients}, C_{D,0} \text{ coefficients}, \text{ etc.}
$$

$$
= t_1, a_{11}, a_{12}, \ldots, t_n, N_{M,1}, b_{11}, b_{12}, \ldots, N_{M,k}, \text{ etc.}
$$

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:

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

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

$$
\text{ICC(3)} = \text{fixed-point value of } J \text{ where } C_{D,0} \text{ coefficients begin}
$$

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:

\[ E\text{REF} = \text{error reference value; } E \text{ in appendix D} \]

\[ E\text{RLIMT} = \text{maximum value of } E \text{ that is acceptable on any particular step} \]

\[ E\text{REF} \] 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 \[ E\text{REF} = 1 \times 10^{-6}, \ ERL\text{IMT} = 3 \times 10^{-6}. \]

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

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

If \( \text{MODOUT} = 2 \), output occurs at equal time intervals of \( \delta\text{ELMAX} \) until \( t = T\text{MAX} \).

If \( \text{MODOUT} = 3 \), output occurs at equal time intervals of \( \delta\text{ELMAX} \) until \( t = T\text{MIN} \), then MODOUT is set equal to 4 by the program.

If \( \text{MODOUT} = 4 \), output occurs every \( n^{th} \) step \( (n = \text{STEPS}) \) until \( t = W\text{STEM} = \text{maximum step limit before problem is completed} \).

\( \text{STPMX} \) = maximum step limit before problem is completed

\( \delta\text{ELMAX} \) = time interval between outputs

\( \text{STEPS} \) = number of steps between outputs

\( T\text{MIN} \) = 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 \( \delta\text{ELMAX} \) is small. STDATA will put MODOUT = 4 and STEPS = 1.

Note that \( T\text{MAX} \) = time at start of a stage, plus the stage time, \( TB(N\text{STAGE}) \), 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

\[ \text{LOOKX} = \text{COMMON C location of arbitrary parameter} \]

\[ X\text{LOOK} = \text{value of C(LOOKX) where an interrupt is desired} \]
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 nth stage. This allows the flight to be flown again from the nth 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 nth stage, set

NSAVE = the number of the nth stage

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

RECALL = any nonzero number

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

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

CLEAR = any nonzero number

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

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

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

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

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

```
INPUT CHECK LIST

<table>
<thead>
<tr>
<th>Takeoff time</th>
<th>Position and velocity</th>
<th>Reference and perturbing bodies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(completely fill in one and only one block)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rectangular</td>
<td>Orbit elements</td>
<td>Spherical</td>
</tr>
<tr>
<td>X =</td>
<td>E =</td>
<td>LAT =</td>
</tr>
<tr>
<td>Y =</td>
<td>OMEGA =</td>
<td>LONG =</td>
</tr>
<tr>
<td>Z =</td>
<td>NODES =</td>
<td>AZI =</td>
</tr>
<tr>
<td>VX =</td>
<td>INCL =</td>
<td>ELEV =</td>
</tr>
<tr>
<td>VY =</td>
<td>MA =</td>
<td>ALT =</td>
</tr>
<tr>
<td>VZ =</td>
<td>P =</td>
<td>VEL =</td>
</tr>
<tr>
<td>IMODE = 2</td>
<td>IMODE = 1</td>
<td>TKICK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMODE = 4</td>
</tr>
<tr>
<td>Output control</td>
<td>Error control feature</td>
<td>Restart feature</td>
</tr>
<tr>
<td>TMIN =</td>
<td>EREF =</td>
<td>NSAVE =</td>
</tr>
<tr>
<td>MODOUT =</td>
<td>ERLIMIT =</td>
<td>RECALL =</td>
</tr>
<tr>
<td>STEPS =</td>
<td>CLEAR =</td>
<td></td>
</tr>
<tr>
<td>DELMAX =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEPMX =</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following standard data are loaded by SUBROUTINE STDATA:

- DTOFFJ = 2440 000.0
- MODOUT = 4
- EREF = 1x10^-6
- IMODE = 1
- BODYCD(1) = (ALF5)EARTH
- STEPS = 1.0
- ERLIMIT = 3x10^-6
- RMASS(1) = 1.0
- STEPMX = 100.0
- TFILE = 1.0
- LOOKSW = 711

At input 300, setting TAPE 3 = 0 is necessary to make an ephemeris tape.
```
APPENDIX H

PROGRAM LISTING

THIS MAIN PROGRAM IS THE SUPERSTRUCTURE ABOVE ALL SUBPROGRAMS.
SUBROUTINE TAPE CLEARS COMMON 1 THRU 4000 AND MAY CONSTRUCT AN
EPHEMERIS TAPE, ALSO, IT ALWAYS SETS TAPE = 0, SUBROUTINE STA
LOADS A STANDARD SET OF DATA. IF RECALL DOES NOT EQUAL ZERO.
A PREVIOUSLY SAVED SET OF DATA FROM STAGE3 IS MOVED TO THE INITIAL
DATA LOCATION. THE MAIN INPUT STATION IS STATEMENT B(INPUT).
WHERE THE VEHICLE DATA FOR ALL STAGES MAY BE LOADED. SUBROUTINE ORDER IS
CALLED TO ORDER THE LIST OF BODIES, DETERMINE THE GRAVITATIONAL CONSTANT.
ORIGIN ROTATION RATE, ATMOSPHERIC RADIUS, RELOCATE ELLIPTIC EPHEMERIS DATA
AND POSITION THE EPHEMERIS TAPE.

COMMON C
DIMENSION A(600), B(700), C(4000),
1 TB(10), W(1100)

EQUIVALENCE
11A ,C ( 1111), ( 1111), CLEAR,C ( 311),
210 ,C ( 1111), (STAGEI, L(311), NCASE, C ( 111),
31NCASEA ,C ( 6001), (STAGEA, L(311), IRECALL, C ( 511),
4TR ,A ( 631), (TAPE3, C ( 211), (TABLE, C ( 19111)

1 CALL INPUT (99,C,TABLE)
IF (TAPE3) 3 2 13
2 CALL TAPE
3 NCASE = NCASE + 1
4 WRITE OUTPUT TAPE 6, 12, NCASE
12 FORMAT(3(3H RECALLED INITIAl DATA FROM STAGE1,0H OF CASE14,1H)
8 CALL INPUT (11,C,TABLE)
IF (SENSE SWITCH 6) 13,14
13 WRITE OUTPUT TAPE 6, 15
15 FORMAT(3H EXIT VIA SENSE SMA)
CALL EXIT
14 IF (STAGE = 11) 11, 9, 11
9 DO 10 STAGE = 11, 10
10 CONTINUE
11 CALL ORDER
17 CALL STAGE
GO TO 1

SUBROUTINE AERO

SUBROUTINE AERO COMPUTES THE LIFT AND DRAG ACCELERATIONS. AS IN SUBROUT-
LINE THRUST, THESE VECTORS ARE REFERENCED TO THE RELATIVE WIND VELOCITY.
COEFFICIENTS OF LIFT, INDUCED DRAG, AND DRAG AT ZERO ANGLE OF ATTACK ARE
ASSUMED TO BE FUNCTIONS OF MISSION NUMBER AND ANGLE OF ATTACK. TABLES OF
CDI/CL**2, CL/SINALPHA, AND COD ARE ASSUMED AS FITTED QUADRATIC EQUAT-
IONS IN THE COSI ARRAY. GASFAC IS THE SPECIFIC HEAT RATIO * STANDARD
ACCELERATION OF GRAVITY / UNIVERSAL GAS CONSTANT. FOR EARTH, GASFAC=
20.064881 (METERS / SEC**2 / KELVIN DEGREE**1/2).

COMMON C
DIMENSION A(600), B(700), C(4000),
1 VATM(3), PL(3), XIFT(3), DRAG(3), PAR(3), X(100)

EQUIVALENCE
11A ,C ( 1111), ALPHA,A ( 491), (AREA ), B ( 611),
210 ,C ( 1111), ETA,A ( 501), (CD ), A ( 1651),
31CDI ,A ( 1693), CL ,A ( 1694), (COSI, B ) ( 491),
41COSI ,B ( 491), (ONSI, B ( 291), (DRAG ), B ( 697),
51GASFAC,A ( 461), LP , B ( 841), (PAR ), B ( 601),
61PMAGN ,B ( 501), (QV ), B ( 591), (B ) ( 1021),
71SINALPH,A ( 461), (SINBET ), B ( 471), (TM ), B ( 341),
81ONSI ,B ( 971), (VMACh ), B ( 391), (VQ ), B ( 1001),
91VQSQRD,B ( 1011), (X ), B ( 4011), (XIFT ), B ( 721)

Q = 0.5*SINALPI*VQSQRD
QV = QONSI*VQ*
VMACH=SQRT(VQSQRD/MTI)/GASFAC

COMPUTE THE X,Y,Z COMPONENTS OF LIFT.
IF (ALPHA) 4,1,2
1 CL = 0.0
GO TO 4
2 CL = QUAD(VMAXI,2)+SINALF
AA = QVAL+CL/PMAGN
AB = SINBET*QV
GO 3 K=1,3
3 XIFTIK = AA+AB*PAR(K)+COSI*P(K)
7 CDI=QUAD1(VMAXI)+31*LMC

COMPUTE THE X,Y,Z COMPONENTS OF DRAG.
4 CD = CDI*QMAXI,6)
AC = CDI*QV/LQ
DO 5 K=1,3
5 DRAGIK = AC*VATM(K)
6 RETURN
END
FUNCTION ARCTAN (Y, X)

THE FORTRAN I1 LIBRARY ATANF(+) OR - 2=TAN(THETA)) USES A SINGLE
ARGUMENT WITH ITS SIGN TO GIVE THETA IN THE FIRST (+2) 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-]*SIN(THETA)]/[+OR-]*
COS(THETA)]. )US THUS THE ARCTAN (Y,X) GIVES THETA IN ITS PROPER
QUADRANT FROM -180 DEGREES TO 180 DEGREES.

FUNCTION ARCTAN (Y, X)

IF (X) 2,1,2
1 ARCTAN=SIGNF(3.14159265,Y)
GO TO 4
2 ARCTAN=ATANF(Y/X)
IFIX 3,1,4
3 ARCTAN=ARCTAN+SIGNF(3.14159265,Y)
4 RETURN
EN0

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(1)
(4) VELOCITY, V(4)
(5) VELOCITY SQUARED, V(5)

COMMON C

DIMENSION A1(600), B1(700), C1(4000),
1 AMCI3), V(5), RBI3), INO3)

EQUIVALENCE
A A C
1 RBI A 111), (IND AN 1 601)
2 (R2B 1 931)

DO 1 J1=1,3
J2=IND(J1)
J3=IND(J2)
1 AMC(J3) = R o(J3) * V(J2) - R a(J2) * V(J1)
AMCI4) = AMCI3)**2*AMCI2)**2*AMCI3)**2
AMCI5) = SQRTFAMCI4)
V(5) = V(1)**2*V(2)**2*V(3)**2
V14) = SQRTFV(5)
RETURN

SUBROUTINE CONVT2

THIS ROUTINE CONVERTS RECTANGULAR COORDINATES INTO ORBIT ELEMENTS.

RECTANGULAR COORDINATES - POSITION COMPONENTS, X,Y, 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 A1(600), B1(700), C1(4000),
1 AMCI3), ORBELS(6), RBI3)

EQUIVALENCE
A A C
1 ORBELS(6) = AMSQRD/GKM
P = SQRTF((1)**2+RBI3)**2*RB(3)**2)
TRU = ARCTANAM/GK2M*RB(1)/VX+RB(2)/VY+RB(3)/VZ, ORBELS16)=R1
IF AMCI1) .2,1,2
1 ORBELS(3)=0.
GO TO 3
2 ORBELS(3)=ARC TANAMC(1)+AMCI2)
3 ORBELS(4)=ARC TANSQRTF(A MCI1)**2+AMCI2)**2, AMC(3)
SNODE = SINF( ORBELS(3))
CNODE = COSF( ORBELS(3))
AB = RB(1)*CNO D E+RBI2)*SNO D E
AA = RB(3)*SINF( ORBELS(4)) + COSF( ORBELS(4)) * (RB(1)*CNO D E- RB(3)*SNO D E)
ORBELS(2)=ARC TANAB, AA=TRU
ORBELS(5)=S QRTF (AMCI1) + ORBELS(6) + V SQR D/ GKM = 2.83)
EPONE = SQRT(F1, ORBELS(1))
E2Ml = 1- ORBELS(1)**2
EPAS = SQRTF (AMCI1) * E2Ml)
SINTRU = SINF( TRU)
COSTRU = COSF( TRU)
EPAS = SQRTF (AMCI1) * ORBELS(1)**2 * SINTRU/11 + COSTRU11 + ORBELS(11) * COSTRU11 * EPAR
4 IF (E2Ml) .5,6,6
5 ORBELS(5) = LGF ( EPONE1 * EPAS) (EPONE1 * EPAS) + ETHETA
GO TO 7
6 ORBELS(15)+2,1,ARC T AN EPAS, EPONE1, ETHETA
7 RETURN

SUBROUTINE CONVT2
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 C
DIMENSION A(6000), B(7000), C(400000),
I RELERR(10), XPRIM(2000), XINC(100)
EQUIVALENCE (A1, B1, C1, D1, E1, F1, G1, H1, I1, J1, K1, L1, M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1, W1, X1, Y1, Z1)
EQUIVALENCE (RELERR(1, A1, B1, C1), XPRIM(1), C1, XINC(1, B1, C1))
EQUIVALENCE (RELERR, A1, B1, C1)
E2 = 0.
RELERR(2) = RELERR(2)/XINC(21)/XPRIM(21)
IF (IMODE-1) = 1, 2
DO 10 J=1, 4
RELERR(J) = RELERR(J)/XINC(J)/6.61853
GO TO 3
C
C COMPUTE THE NORMALIZED INTEGRATION ERRORS FOR THE ORBIT ELEMENTS.
1 RELERR(3) = RELERR(3)/XINC(3)/1.1/10.
RELERR(4) = RELERR(4)/XINC(4)/10.
DO 10 J=1, 4
RELERR(J) = RELERR(J)/XINC(J)/6.61853
GO TO 3
C
C COMPUTE THE NORMALIZED INTEGRATION ERRORS IN RECTANGULAR VARIABLES.
2 V1 = V1/100.
DO 20 J=1, 3
RELERR(J) = RELERR(J)/V1
RELERR(J) = RELERR(J)/XINC(J)
DO 20 J=1, 3
RELERR(J) = RELERR(J)/XINC(J)
C
C SELECT MAXIMUM ERROR, COMPUTE ERROR COEFFICIENT, POSSIBLY SAVE ERROR DATA.
3 DO 5 J=1, 4
IF (ABS(RELERR(J))-E2) > 5, 4
4 K=J
E2 = ABS(RELERR(J))
5 CONTINUE
C
SUBROUTINE EQUATE
THIS SUBROUTINE IS CALLED FROM NBODY TO EVALUATE THE DERIVATIVES OF THE
VARIABLES OF INTEGRATION. EITHER RECTANGULAR COORDINATES OR ORBIT ELE-
MENTS MAY BE USED AS THE VARIABLES OF INTEGRATION, BUT IN THE CASE OF THE
LATTER, THE CORRESPONDING RECTANGULAR COORDINATES MUST FIRST BE FOUND.
THIS IS DONE AT THE BEGINNING THRU THE USE OF KEPLERS EQUATION. THE
PERTURBATING ACCELERATIONS ARE FOUND BY CALLING VARIOUS OTHER SUBROUTINES
AND THEIR SUM RESOLVED ALONG THE X,Y,Z AXIS. FINALLY, THE DERIVATIVES
ARE CALCULATED. IN THE CASE OF ORBIT ELEMENTS, THE X,Y,Z PERTURBATING
ACCELERATION COMPONENTS MUST FIRST BE RESOLVED INTO CIRCUMFERENTIAL,RADIAL
AND NORMAL COMPONENTS. THIS ROUTINE ALSO CHANGES THE INTEGRATION VARI-
ABLES FROM ORBIT ELEMENTS TO RECTANGULAR VARIABLES IF THE ECCENTRICITY
APPROACHES UNITY. THE X,XPRIM, AND XDOT ARRAYS ARE AS FOLLOWS.
COMMON C
DIMENSION A(6000), B(7000), C(400000),
1 XPRIM(1000, 21), X(1000),
2 R(3), NEFRMS(8),
3 XPRIM(1000, 21), FORCE(3),
4 DRAG(3), OBLAT(3),
5 XDOT(1000)
COMPUTE SINE AND COSINE OF TRUE ANOMALY.
PART A, E=1
3 IF (EOMNE) 10,4,5
4 SINTU = 0.
COSTU = 1.
GO TO 14
C
PART a, E IS GREATER THAN 1
5 DD 7 J = 1, 10
DELH = XI7-U-XI3!+SINFU)
ECOSU = X13*ECOSFU)
DELU = DELM/(L-0-ECSU+0.01*ECOSU+3))
U = L DELU
6 IF (ABSFIUDELH) = CONSTU 9,9,7
7 CONTINUE
ASYMP = 1.0
IF (MODUSI) = 23, 8
8 CALL EPHMRb
GO TO 23
9 COSU = ECOSFU)
DEM = L-XI3I+ECOSU
COSTU = (ECOSU-XI3I)*OEM1
SINTU = EPAR-SINFIU)/OEM1
GO TO 14
C
PART C, E IS LESS THAN 1
10 DD 12 J = 1,19
DELH = XI7-U-XI3I+SINFU)
ECOSU = X13*ECOSFU)
DELU = DELM/(L-0-ECSU+0.01*ECOSU+3))
U = L*DELU
11 IF (ABSFIUDELH) = CONSTU 13,13,12
12 CONTINUE
WRITE OUTPUT TAPE 6.55, V, DELU
CALL EEP
13 COSU = COSFU)
DELH = L-XI3I+ECOSU
COSTU = (ECOSU-XI3I)/OEM1
SINTU = EPAR-SINFIU)/OEM1
14 PDR = L-XI3I*OEM1
GO TO 14
C
COMPUTE POSITION AND VELOCITY FROM ORBIT ELEMENTS AND TRUE ANOMALY.
ALSO, CLEAR THE PERTURBING ACCELERATIONS.
15 SOMEA = SINFU)
COMEA = COSFU)
SNDUE = SINFU)
CNDE = COSFU)
SINCL = SINFU)
COSCL = COSFU)
SINV = SINTU+COSMEASUR+OSMEAGAMEA
COSY=COSY+COSMEASUR+OSMEAGAMEA
AMAS=AMAS+SINMEASUR+SINMEAGAMEA
COSY=COSY+COSMEASUR+OSMEAGAMEA
AMAS=AMAS+SINMEASUR+SINMEAGAMEA
L1 = L-XI3I+SUMBAGASINV
F1 = L-XI3I+SUMAGA+CDV
AS = E1+CNDUE+P1*SNUG+CINCL
42Q = E1+CNOUE+CINCL*E1*JMODU
R = X1B)*PVFR
RSORD = +)
SINV = SINV+CINCL
RBl) = E1*AN
RB12 = E1C)
RBl3) = E1*SINV
VXI 1= VECIRL+AS
VX12 = VECIRL+AS
VX13 = VECIRL+P1*VINCL
GO TO 18
TEST FOR PRESENCE OF PERTURBING BODIES:
IF (IMODY) 20,21,20
20 CALL EPHMRS
21 IF (IXBSPF/IMODE)=1) 26,22,26

TEST FOR CHANGE FROM ORBIT ELEMENTS TO TEMPORARY RECTANGULAR COORDINATES IF E IS TOO NEAR TO UNITY:
22 IF (ETDCL-ABS(FIEMONE)) 26,23,23
23 IF (IMODE) 54,24,24
24 IMODE=-3
25 IF (INSTRT) 25,54,25
26 TEST = XJ7
27 CALL TESTR

TEST FOR OBLATENESS PERTURBATION COMPUTATION:
26 IF (FBLATN/=NAME)30,29,30
29 CALL OBLATE

TEST FOR PRESENCE OF THRUST:
30 XDOTI(2) = FLOW
IF (R-RATMO) 31,31,32
31 CALL ICAO
GO TO 33
32 PRESS=0.
33 IF (PUSMO) 37,30,37
36 ASSIGN 40 TO NOONE
GO TO 38
37 CALL THRUST
ASSIGN 41 TO NOONE
GO TO 39
38 IF (PRESS) 39,42,39
39 GO TO NOONE, 40,41
40 CALL THRUST
41 CALL AERO

SUM COMPONENTS OF THE PERTURBING ACCELERATION:
42 DO 43 J=1,3
43 COMPAl(J)=-QXIAl(J)+OBLATI(J)++FORCEI(J)+DRAGI(J)
44 GO TO (47,45,45),IMODE

COMPUTE DERIVATIVES FOR THE RECTANGULAR VARIABLES OF INTEGRATION:
45 AA = GZKM/RSQRO
GO 46 K=1,3
46 XDOTI(K+5) = XK(K+1)
47 XDOTI(K+2) = COMPAl(AA*K(K+5))
GO TO 54

COMPUTE THE DERIVATIVES OF THE ORBIT ELEMENTS. (AFTER RESOLVING PERTURBING ACCELERATION INTO CIRCUMFERENTIAL, RADIAL, NORMAL COMPONENTS)
47 CIRCUM=COMPAl(3)+COSV+SINCL-COMPAl(I)+B1-COMPAl(2)+OI
RADIAL=COMPAl(A)+COMPAl(2)+C1+COMPAl(3)+SINY
NORMAL=COMPAl(I)+SINDE+SIGNL-COMPAl(2)+CNGDE+SIGNL-COMPAl(3)+SINCL
ZVRPPI = 1./POVR + 1.
RNUA = G2M/POVR
XDOTI(B) = 2.*R/VCIRCL*CIRCUM
IF (X33) 48,48,48
48 CSQRD = CIRCUM*CIRCUM
RASQRO = RAU(I)+RASRO
DEMl = (4.*CSQRO+RASQRO)/VCIRCL

TEST FOR IN-PLANE PERTURBATION:
49 IF (DEMl) 50,50,50
50 XDOTI(3) = 0.
51 XDOTI(4) = 0.
52 XDOTI(7) = 0.
GO TO 50
53 XDOTI(3) = SQRTI(4) + CSQRO+RASQRO)/VCIRCL
54 XDOTI(4) = SQRTI(4) + CSQRO+RASRDl/DEMl+RADIAL
55 XDOTI(7) = ZN-VDVZRO+46.*CSQRO+RASQRO/DEMI+RADIAL
GO TO 50
56 XDOTI(3) = (SINTRU+RADIAL+PDVR-ROVA)/X33+CIRCUM/VCIRCL
57 XDOTI(4) = (SINTRU+RADIAL+PDVR-ROVA)/X33+CIRCUM/VCIRCL
58 XDOTI(7) = ZN-VDVZRO+46.*CSQRO+RASQRO/DEMI+RADIAL
GO TO 50
59 IF (SINCL) 54,52,51
60 XDOTI(5) = SINL/SINCL+JORMAL/VCIRCL/POVR
GO TO 53
61 XDOTI(5) = 0.
62 XDOTI(6) = CUSV+JORMAL/POVR/VCIRCL
63 RETURN
64 FORMAT(1HOKEPLENS EQUATION CONVERGENCE FAILURE, U=15.8,7H DELU= 1G15.8,7H)
END
SUBROUTINE EPHMRS

SUBROUTINE EPHMRS 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 (ITRSFER=11) 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 00 19 LOOP ENCOMPASSES ALMOST THE ENTIRE EPHMRS SUBROUTINE AND, IN EFFECT, ELIPSE TOO.

COMMON C

DIMENSION A(600), B(700), C(4000),
QX(3), (SODY(18), EPHMRS(7), XP(3,8), RB(3,8), R (8), TIM(7),
NEPHMR(1), TDATA(6,3,7), TDEL(7), BMAS(5), VEPHM(3,8), DATA21)
,
C = , TDA1(6,7)

EQUIVALENCE
11A ... C (111), (AU ... C (29)), I8 ... C (111)),
21BMAS ... C (137)), (TOFF,JA (23)), (EPHMR, B (139)),
31BODY ... C (177)), (MODYS,B (42)), (EPHMR,B (165)),
40QX ... A ... B (781,18), A (102), (RB ... B (193)),
51SQRK ... A (351), (SQRK A (44)), (TAYL ... B (201)),
61DATA ... A (205), (TDEL ... B (170)), (TIM ... B (193)),
71TMR & ... B (81), (VEFM ... B (241)), (XP ... B (217)),

EQUIVALENCE (IBF,FI8), (DATA, TDATA)

C PART 2. SET INDEXES, FIND POSITION IF ELLIPSE IS USED (NEPHMR = 20 OR UP).
DO 19 JB=1,MODYS
IB=JB41
IF (MODYS(JB)=00 20) 2=2,1
1 CALL ELIPSE(JB)
IF (ITRSFER) 12,12,17

C PART 3. TAPE EPHEMERIS IS TO BE USED. FIND DIFFERENCE (DT) BETWEEN CURRENT PROBLEM TIME AND MIDPOINT TIME OF CURRENTLY STORED TAPE DATA. THEN SEE IF CURRENT DATA IS OKAY. DT = TABL - TIM FOR WHICH CURRENT DATA IS GOOD.
2 DT = TABL - TIM(JB) - TOFF(JB)

C PART 4A. CURRENT DATA NOT OKAY. READ IN NEXT DATA SET. IF DT IS -,
BACK UP THE TAPE 2 RECORDS BEFORE READING.
3 IF (DT) 4,5,5
4 BACKSPACE 3
5 READ TAPE 3, (DATA(1), =1,1)21

C PART 4B. IF THIS DATA IS FOR A BODY IN THE BNAME LIST, STORE IT.
IF (IF NOT STORED, WE MIGHT HAVE TO RETURN FOR IT.) IF ELLIPSE DATA IS PROVIDED FOR THE BODY FOUND, BY-PASS THE TAPE DATA AND READ IN NEXT SET.
2 IF (DATA(I)=EPMAS(J)=(-DATA(I)=EPHMR(J))) 7,6,7
6 IF (EPHMR(J)=20) 8,8,3
7 CONTINUE
GO TO 3

C PART 4C. MOVE THE DATA INTO PLACE AND THEN GO BACK AND SEE IF IT IS OKAY.
8 TIM(J) = DATA(1) TDATA(J) = DATA(3) TDATA(J) = DATA(J+3)
9 CONTINUE
GO TO 2

C PART 5. CURRENT DATA IS OKAY. GET POSITION FROM THE POLYNOMIAL
P = A + BX + CX**2 + DX**3 + EX**4 + FX**5.
10 DO 11 K=1,3
XP(I,JB) = TDATA(I,K,JB)
11 CONTINUE
IF (ITRSFER) 12,12,15

C PART 6. COMPUTE DISTANCE FROM REFERENCE AND FROM ROCKET .
12 DO 13 K=1,3
XP(I,JB1) = XP(K,JB1) + XP(K,JB1) + SGNFAU(FI8)
13 R(BK,JB1) = R(BK,JB1) - XP(K,JB1)
C PART 7. COMPUTE PERTURBING ACCELERATIONS (QX). 4194304=2**22 IS REMOVED TO PREVENT OVERFLOW. 2048=2**11 AND 8589934592=2**33 RESTORE THE SCALE.
PRSQRO = (R61,JB1)**2 + R2(JB1)**2 + R61(3,JB1)**2 + 4194304.
RRELL = SQRTFPRSQRO
RSQRO = (XP1,JB1)**2 + XP2(JB1)**2 + XP3(JB1)**2 + 4194304.
RSPRO = FIX(RSQR, SQR, SQRO)
PSCUBE = RSPRO + RRELL
RJ(1) = RRELL/2048.
14 QX(I) = SQRK * BMAS(1,B) * ((XP(I,JB1)+ACUBE) + R(B1,JB1)+PSCUBE)+
I 8589934592 * QX(I)
GO TO 19

C PART 8. COMPUTE VELOCITY FROM V = B + 2CK + 3DX**2 + 4EX**3 + 5FX**4
AND FROM REFERENCE BODY VELOCITY (VEFM(JB1)).
15 DO 16 K=1,3
VPBK(JB1) = XP1(JB1)+DPBATATK(JB1)+FLATAFI-KT+DJ)
16 VEPKM(JB1) = (VEPK(JB1)+DT+TDA1(TK(JB1)+FLATAFI-KT+DJ)
17 DO 18 K=1,3
18 VEPK(JB1) = VEPKM(JB1)+VEPK(JB1)+SIGNEDU(SPD,F18)
GO TO 12
19 CONTINUE
RETURN
END
SUBROUTINE EXTRA
C
C THIS ROUTINE IS EXECUTED BETWEEN FLIGHTS AND MAY THEREFORE BE EXPANDED TO
C DO ADDITIONAL COMPUTATION BETWEEN SUCCESSIVE FLIGHTS.
C
COMMON C
C
DIMENSION A16001, B17001, C140001
C
EQUIVALENCE (1A, C ( 111), B, C (1111), (QMAX, B ( 44)),
2 (SIGNAL, B ( 31))
C
SIGNAL = 0.,
QMAX = 0.,
RETURN
END

SUBROUTINE EXTRAS
C
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 (JBI)
C
C THIS SUBROUTINE IS CALLED FROM EPHMRS TO COMPUTE THE POSITION OF A BODY
C USING APPROXIMATE ELLIPTIC DATA. THE VELOCITY IS ALSO COMPUTED IF THE
C ORIGIN IS BEING TRANSLATED (TRSFER=1.0). THE ELLIPSE DATA IS READ FROM
C INPUT CARDS AND ORGANIZED IN SUBROUTINE ORDER. TPO IS TIME SINCE PERIHELION
C PASSAGE, ZM IS MEAN ANOMALY, U IS ECCENTRIC ANOMALY.
C
COMMON C

DIMENSION A16001, B17001, C140001,
1 XP1(3, 1), YEPM(3, 0), TDATA(121)
C
EQUIVALENCE (1A, C ( 111), B, C (1111), (CONS, A ( 31)),
2 (TOFFJ, A ( 23)), (TABLT, B ( 201), (TDATA, B ( 265)),
3 (TRSFER, B ( 91), (VEPM, B ( 241), (XP, B ( 217))
C
K = 1B+4(JBI-2)+1
TPO = (TOFF-2-4+K*5)+TBLT-TDATA(K+6))
ZM = 6.28318533/TDATA(K+7)
ZM = ZM+MOD(TPO, TDATA(K+7))
C
GET THE SINE (SINT) AND THE COSINE (COSTR) OF THE TRUE ANOMALY
BY ITERATING KEPLERS EQUATION. THEN COMPUTE X, Y, Z (XPI).
U = 2M+TDATA(K+11)-SINT/2.9+TDATA(K+13)*SINT/2.9
DO I=1,10
DEL = 2M-U-TDATA(K+21)+SINT/I
DELU = DELM(1-I-TDATA(K+11)+COSTR(I))
U = U+DELU
IF (DELM+DEL) = CONS U ZM, ZM, ZM
1 CONTINUE
2
COSU = COSTR(I),
DENOM = 1-TDATA(K+11)+COSU
COSTR = (COSU-TDATA(K+11))/DENOM
PHI = TDATA(K+11)+COSTR
SINT = SQRT(1-PHI**2),
SINF = SINT/2.9+TDATA(K+13)
IF (SINF/SINT) = CONS U ZM, ZM, ZM
C
COMPUTE THE VELOCITIES FOR THE TRANSFER OF ORIGIN.
3 EX = TDATA(K+11)+COSTR*2.9+SINF
F = TDATA(K+11), TDATA(K+11)*2.9+SINF
CFACT = 2M+TDATA(K+11)+COSU
AK = EX+ATDATA(K+2)+F*DATA(K+9)+DATA(K+13)
BK = FX+DATA(K+13)+TDATA(K+13)-EX+DATA(K+9)
VEPM(1, JBI) = -AX*CFACT
VEPM(2, JBI) = BX*CFACT
VEPM(3, JBI) = FX*CFACT+TDATA(K+10)
4 RETURN
END
REM SUBROUTINE EXADD (A,B,C)
REM THIS ROUTINE WILL ADD IN DOUBLE PRECISION A QUANTITY C TO THE
REM DOUBLE 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 TEMPI COMMON 1 TEMP2 COMMON 1

ENTRY 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 ATOMSPHERE (ICAD TO 20 KM.). A SHORT FAD
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
R IS DISTANCE TO CENTER OF EARTH IN METERS.
C ALT IS VEHICLE ALTITUDE ABOVE EARTH IN METERS.
C TABLE H IS METERS OF ALTITUDE FROM THE EARTHS SURFACE AND IS
C THE ARGUMENT OF ATMOSPHERE PROPERTY TABLE.
C ALM IS THE MEAN SLOPE OF THE TABLE H VS. TM CURVE AT TABLE H.
C TMR IS TM AT TABLE H.
C REF P IS THE PRESSURE IN MILLIBARS AT TABLE H.
C TM IS THE TEMPERATURE TIMES 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

DIMENSION A(1600), B(1600), C(4000),
1 TABLEH(23), TM(23), REFP(23), ALM(23), RB(3)
C
EQUIVALENCE
31 PRESS [3111] , [31111] , [31111] , [31111],
41 REFP [4111] , [41111] , [41111],
51 RESQRO [5111] , [51111], [51111], [51111],
61 EQUVALENCE TABLEH(24), TM(24), TABLEH(47), ALM(27), TABLEH(70), REFP
C
IF (OBLATN) 102,101,102
101 ALT = R - RE
GO TO 103
102 ALT = R - 6356783.2B/SQRTF(.9933065783+.DD66934216B51R131/R131**2)
103 IF (ALT<90000.) 105,104,104
104 HEIGH = ALT
GO TO 106
105 HEIGHT = ALT/1.0+ALT/6356766.1
106 K=K
C
FIND THE
HEIGHT IN A TABLE OF BASE DATA. DATA ARE
ARRANGED IN DECREASING ALT WITH 21 REGIONS. ABOVE THAT, PRESSURE AND
DENSITY ARE SET = 0. TEMPERATURE IS SET TO 3000.
1 IF (K<22) 2,6
2 IF (HEIGHT-TABLEH(K+11) 5,3,3
3 K = K+1
GO TO 1
4 K = K-1
5 IF (K) 7,7,6
6 HINC = HEIGHT - TABLEH(K
7 K = 1
8 IF (ALM(K)) 9,100,9
C
CONTROL COMES HERE FOR NONISOTHERMAL LAYERS

IF (ATL-90000.1) \nIFI10,10,108 \n107 PRESS = REFPK(1)(TMRK(1)/TM)**0.341631947/ALM(1) \nGO TO 10

108 IF (K-KC) 109,110,109

109 KC = K

C1 = RE+TABLEH(K) \nC2 = TMRK(K)/ALM(K) \nC3 = 1.0/(C1-C2)

C4 = -0.341631947*RESTM+D3*ALM(K)

110 PRESS = REFPK(1)*EXP(-0.341631947*HINC/C2+C1/1.0*RESTM)

GO TO 10

100 IF (K-KC) 110,112,112

111 TM = TMRK(K)

PRESS = REFPI+EXP(-0.341631947*HINC/TMRK(1))

GO TO 10

112 PRESS = 0.0

DENS = 0.0

TM = 3000.

RETURN

C CONTROL COMES HERE FOR ISOTHERMAL LAYERS

REM THIS IS THE FAP PROGRAM WHICH LOADS ICAO DATA INTO MACHINE.

1100 IF (K-KC) 111,112,112

111 TM = TMRK(K)

PRESS = REFPI+EXP(-0.341631947*HINC/TMRK(1))

GO TO 10

C CONTROL COMES HERE FOR EXTREME ALTITUDES

1200 IF (K-KC) 111,112,112

112 PRESS = 0.0

DENS = 0.0

TM = 3000.

RETURN

END

SUBROUTINE NBODY

NBODY COMPUTES THE TRAJECTORY IN EITHER ORBIT ELEMENTS OR RECTANGULAR COORDINATES USING THE RUNGE-KUTTA TECHNIQUE. A LOWER ORDER INTEGRATION TECHNIQUE IS ALSO PERFORMED TO FACILITATE AUTOMATIC STEP SIZE CONTROL.

THE \( X, XPRIM, \dot{X}, XPRIMB, XDOT, XDOTB, XWHOLE, XWHOLEB, \) EARTH SPHERICAL - CHANGE TO RECTANGULAR ORBIT ELEMENTS - CHANGE TO TEMPORARY RECTANGULAR EARTH SPHERICAL - CHANGE TO ORBIT ELEMENTS - CHANGE TO TEMPORARY RECTANGULAR ORBIT ELEMENTS - CHANGE TO TEMPORARY RECTANGULAR EARTH SPHERICAL - CHANGE TO ORBIT ELEMENTS

COMMON C

DIMENSION A(1000), B(1000), C(1000), X(1000), Y(1000), Z(1000), V(1000), W(1000), U(1000), T(1000), N(1000), M(1000), L(1000)

1 XPRIM (100,2), XPRIMB (100,2), XDOTM (100,2), XDOTM (100,2)
2 X (100), X (100), XDOT (100), XDOT (100)
3 XDOT (100), XDOT (100), XDOT (100), XDOT (100)
4 AMC (3), AMC (3), AMC (3), AMC (3)
5 XWHOLE (6), VX (3), VX (3), VX (3)
PART 1. SET UP THE STARTING SEQUENCE FOR ERROR CONTROL AND DELAY CHECKING
THE ERROR UNTIL TWO STEPS ARE COMPLETED. THE ASSIGNED GO TO'S NSTART AND

NEQ = NEQ
1 DO 2 J=1,NEQ
 2 X(J) = XPRIM(J,1)
NSTART = 0
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
IF (IMODE = 4,5,5)
4 CALL TESTTR
GO TO 1
5 CALL TESTTR
IF (ITRSFER) 1,205,1
205 ASSIGN 21 TO NSTART
C STATEMENTS 7 TO 9 INITIALIZE NREV1 AND NREV2 FOR USE IN PART TA.
C PART 50. MAX ERROR TEST--STARTING ONLY--CHECK THE MAX ERROR AND
C EITHER ENTER RUNNING MODE OR REPEAT START WITH SMALLER STEP.
 23 GO TO J+2,NEQ
 24 XINC(j) = XINC(j)+OLDINC(j)+3.-IXDOTPM(j,1)*XDOTPM(j,2)+4.  
                    1*XDOTD(j)))*DEL)
240 CALL ERRORZ
 25 IF (ERRORZ.LT.EARLMT) 26,26,56
26 ASSIGN 27 TO NSTART
ASSIGN 11 TO IBEGIN
A1 = A2
GO TO 31
C
C PART 6. RUNNING PHASE PROGRAM.
C PART 6A. CHECK THE INTEGRATION BY INTEGRATING OVER THE LAST
C RUNGE-KUTTA STEP BUT USE DOTS FOR LAST TWO INTERVALS, OLDEL
C AND DELT RESPECTIVELY. STATEMENT 28 IS THE LOWER INTEGRATION
C RING RUNGE-KUTTA INCREMENTS. ERRORZ COMPUTES THE MAXIMUM RELATIVE
C ERROR AND STATEMENT 29 TESTS THIS ERROR AGAINST THE
C LIMIT VALUE.
 27 RATIO = DELT/OLDEL
 28 XINCIJI = XINCIJI+OLNOINCJJI-I+XOOTPMJlJ.1+XOOTPMj2)+4.  
                    1*XOOTD(j)))*DEL
DD 28 J+2,NEQ
 29 XINC(j) = AGCDF1*XDOTPM(j,1)+AGDF2*XDOTPM(j,2)+6.*XINC(j)  
                    1*AGOEF*DOT(j)
290 CALL ERRORZ
 29 IF (ERRORZ.LT.EARLMT) 30,30,57
C
C PART 7A. LAST POINT OKAY. COUNT THE REVOLUTIONS PAST THE X-AXIS.
C A STEP GREATER THAN 1/2 REV. MAY FAIL TO ADD IN.
 30 HZ = DELT
 31 IF (HZ.GT.MAXFQ,QMAX) 32,32,34
 32 GO TO NREV1 (37,33)
33 ASSIGN 37 TO NREV1
ASSIGN 35 TO NREV2
GO TO 37
34 GO TO NREV2 (37,35)
35 ASSIGN 33 TO NREV1
ASSIGN 37 TO NREV2
36 REVS = REVS + 1.
37 IF (XABSFLI~OOEl-L-1) 42,38,42
C
C PART 7B. IN ORBIT ELEMENTS. ADJUST ARGUMENT OF PERICENTER AND MEAN ANOMALY
C TO = OR = PI TO MAINTAIN ACCURACY IN SIN-COS ROUTINES.
 38 IF (EMONE).EQ.90.,92,42
 39 DD 41 J+4,7,3
 40 ADJ2 = INTFIX(XPRIM(J,2)/6.28318532+SIGNDE(5,XPRIM(J,2))
IF (ADJ2EQ 40,92,40
 40 ADJ3 = -ADJ2+0.28179
400 CALL EXAOEI(XPRIM(J,1),XPRIM(J,2),ADJ3)
ADJ3 = ADJ2+0.019353072
401 CALL EXAOEI(XPRIM(J,1),XPRIM(J,2),ADJ3)
41 CONTINUE
C
C PART 7C. ADVANCE THE REMAINING PARAMETERS, FIND NEW STEP SIZE.
C AND TEST FOR AN ORIGIN TRANSLATION.
 42 DD 43 K+1,3
 43 XWHOLE(K) = XK(K)
 44 DD 44 J+1,NEQ
XDOTPM(J,1) = XDOTPM(j,2)
XDOTPM(J,2) = XDOTJ1
XPRIM(J,1) = XPRIM(J,2)
XPRIM(J,2) = XPRIM(J,1)
KINC(J) = 0.
44 CONTINUE
OLDEL = DELT
45 CALL STEP
IF (EONEEQ 67,450,67
450 IF (NSTART) 451,1,451
451 IF (MOSYSI) 46,47,46
46 CALL TESTIR
IF (TRANSF) 1,47,1
47 IF (XABSFIKMOD= 63) 11,48,11
C
C PART 7D. IF IN TEMPORARY RECTANGULAR COORDINATES, TEST FOR RETURN
C TO ORBIT ELEMENTS. FIRST, E IS FOUND. IF TIME HAS NOT ADVANCED
C SUFFICIENTLY, INTEGRATION CONTINUES IN RECTANGULAR VARIABLES (STATE. 48).
C STATEMENT 49 DETERMINES IF KEPLERS EQUATION CAUSED IMODE = 3. IF NOT,
C AN E CLOSE TO 1 CHECK IS MADE IN STATEMENT 50. IF IT DID, RECTANGULAR
C VARIABLES WILL BE USED IF THE LIMIT IS TOO SMALL (STATEMENT 52). OR
C IF E IS 5 OR GREATER (STATEMENT 53) OR IF THE PATH LIES CLOSE TO AN
C ASYMPYT (STATEMENT 55).
 48 CALL CONV1(VX,AMC)
EINODE = SQRT(1.+AMSQR/GK2+IVSQR/GK2-2./RJ)
EMONE = EMODE-1.
IF (XPRIMI1-TEST)+DEL') 11,11,49
49 IF (ASYMPT) 51,50,51
50 IF (ETOLABS(EFION)) 55,11,11
51 IF(EFION) 55,55,52
52 IF(CONSTU1,1-7) 11,53,53
53 IF (EMODE 95) 54,51,11
54 CALL CONV2
IF (ABS(TRAN)=2.2/SQRT(F(EXTIME))) 55,55,51
55 ASYMPT = 0.0
IMODE = 2
555 CALL TESTIR
GO TO 1
C
C PART 8. COMES HERE WHEN ERROR TEST FAILED-SOOTH 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 I TO IBEGIN
57 DO 58 J=1,NEQ
XPRM(J,J1) = XPRM(J,J2)
J00T(J1) = J00T(J2)
XINC(J1) = 0.
58 CONTINUE
STEPNO=STEPNO+1.
H2 = DELT.
DELTS=SIGNFLH2*LOG-A2/5.1.DELT~
CONTINUE
I11 = INCLINATION OF THE ORBIT.
I12 = SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
I13 = ARGUMtNT OF PERIHELION.
I14 = LONGITUDE OF ASCENDING NODE.
I15 = PERIOD OF THE ELLIPSE IN MEAN SOLAR DAYS.
I16 = PERIGE PASSAGE JULIAN DAY.
I17 = PERIGE PASSAGE FRACTION OF DAY.
I18 = ORBIT.
I19 = AMOUNT OF PERIHELION.
I20 = INCLINATION OF THE ORBIT.
I21 = NAME OF BODY IN BCD, ONLY & CHARACTERS.
I22 = NAME OF REFERENCE BODY IN BCD, SAME RESTRICTION.
I23 = MASS OF THE BODY IN SOLAR MASSES.
I24 = DISTANCE INSIDE OF WHICH COORDINATES WILL BE TRANSFORMED TO THIS BODY.
I25 = SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
I26 = ECCENTRICITY OF THE ORBIT.
I27 = PERIGE PASSAGE JOLAN DAY.
I28 = PERIGE PASSAGE FRACTION OF DAY.
I29 = PERIOD OF THE ELLIPSE IN MEAN SOLAR DAYS.
A BLOCK IS ARRANGED AS FOLLOWS:
(1) = NAME OF BODY IN BCD, ONLY & CHARACTERS.
(2) = NAME OF REFERENCE BODY IN BCD, SAME RESTRICTION.
(3) = MASS OF THE BODY IN SUN MASSES.
(4) = DISTANCE INSIDE OF WHICH COORDINATES WILL BE TRANSFORMED TO THIS BODY.
(5) = SEMILATUS RECTUM IN ASTRONOMICAL UNITS.
(6) = ECCENTRICITY OF THE ORBIT.
(7) = LONGITUDE OF ASCENDING NODE.
(8) = ARGUMENT OF PERIHELION.
(9) = INCLINATION OF THE ORBIT.
(10) = PERIGE PASSAGE JULIAN DAY.
(11) = PERIGE PASSAGE FRACTION OF DAY.
(12) = PERIOD OF THE ELLIPSE IN MEAN SOLAR DAYS.
AMASS = MASS OF EACH BODY, SUN MASSES. ORDER OF PNAME.
BNAME = THE ORDERED LIST OF BCD BODY NAMES. CAN BE USED IN OUTPUT.COMMON.
BODYCD = THE ORIGINAL BCD NAMES READ FROM CARDS.
BODY = THE LIST OF BCD BODY NAMES WITH THE REFERENCE BODY AT TOP.
IBODY = ARRAY OF SUBSCRIPTS, WHEN A DISTANCE IS FOUND FROM EPHEMERIS, IT MAY BE ADDED (OR SUBTRACTED) FROM THE BODY POSITION GIVEN BY XPRM(IBODY). IT TAKES THE POSITION OF THE PRESENT BODY. COMMON.
KZERO = COUNT OF ZERO REFERENCES. THERE MUST BE ONE AND ONLY ONE ZERO.
MANE = ARRAY OF SUBSCRIPTS, GIVES LOCATION OF BODY IN PNAME LIST.
NBODY = COMUTED INTERNALY. TOTAL NUMBER OF BODIES.
NBODY = NUMBER OF EPHemerides, BNAME(11).
NAME = ARRAY OF SUBSCRIPTS, GIVES DLL LOCATION OF NAME IN BODY.
NEFRS = ARRAY OF SUBSCRIPTS. GIVES LOCATION OF BODY IN PNAME LIST.
IN TERMS OF THE EPFRS LIST. STORED IN COMMON.
NFREFER = ARRAY OF SUBSCRIPTS, LOCATES THE REFERENCE BODY IN BODY.
ORDER OF THE ARRAY CORRESPONDS TO BODY, NOT IN COMMON.
PNNAME = A PERMANENT LIST OF BCD BODY NAMES. 1 WORD EACH 16 CHARACTERS MAX. USED TO IDENTIFY MASS, REFERENCE NAMES, ETC. THE LIST IS A MAXIMUM OF 30 NAMES. PRECISION TAPE NAMES ARE FROM 1 TO 20.
ELLIPSE NAMES ARE FROM 21 TO 30.
References are the references of the distances given in ephemerides (tapes or ellipse). CORRESPONDS TO PNAME LIST.
COMMON C
DIMENSION A(400), B(700), C(4000)
C THIS SECTION SEES WHAT ELLIPSE DATA WAS READ FROM CARDS AND PUTS THE
C NAMES IN PLACE SO THAT DATA WILL BE USED IF NEEDED. ELLIPSE DATA HAS
C PRIORITY OVER TAPE DATA BECAUSE LAST DATA IN LIST IS THAT ACTUALLY USED.
C
C FUNCTION COMPAREFA.B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
C
C EQUIVALENCE (NAME(1),NBDYS+1)
C
C DO 2 K=1,12,12
C IF (ELIPSIK) 3,2,1
C KOUNT = (K-1)/12+2
C NAME(KOUNT) = ELIPSIK
C REFER(KOUNT) = ELIPS(K+1)
C AMASS(KOUNT) = ELIPS(K+2)
C RCRT(KOUNT) = ELIPS(K+3)
C 2 CONTINUE
C
C PART 0. THROW AWAY BLANKS AND DUPLICATES IN BNAME LIST.
C ALSO COUNT THE BODIES.
C IF (TRANSFER) 4,3,4
C 3 BNAME(I) = GODYC(I)
C 4 DO 5 K=1,12
C 5 NAME(K) = BODYC(I)
C L = 1
C BODY(I) = 0.
C DO 8 I=1,9
C BODY(I) = 0.
C DO 6 K=1,12
C IF (COMPARE (BNAME(I)), BODY(I-K+1)) 6,7,6
C 6 CONTINUE
C BODY(I) = NAME(I)
C L = L+1
C 7 BNAME(I) = 0.
C 8 CONTINUE
C NBDYS = L-1
C NBDYS = NBDYS+1
C
C PART 1. FIND THE REFERENCE BODY FOR EACH BODY IN THE LIST OF BODIES
C READ FROM CARDS. CLEAR REFER AND BNAME.
C DO 13 KL=1,NBDYS
C REFER(KL) = 0
C NAME(KL) = 0.
C DO 12 K=1,30
C IF (COMPARE (BODY(K)), NAME(KP)) 12,9,12
C 9 REFER(KP) = K
C DO 11 K=1,8
C IF (COMPARE (REFER(KP), BODY(KP))) 11,10,11
C 10 REFER(KP) = KR
C 11 CONTINUE
C 12 CONTINUE
C 13 CONTINUE
C
C PART 2. COUNTS 0 REFERENCES AND SAVES TEMPORARY SET OF INDEXS.
C 14 IF (NBDYS) 24,24,15
C 15 KIROS = 0
C MISPEL = 0
C DO 20 K = 1,NBDYS
C NAME(KP) = REFER(KP)
C 16 IF (REFER(KP)) 18,17,18
C 17 MISPEL = MISPEL + 1
C 18 IF (REFER(KP)) 20,19,20
C 19 KIROS = KIROS + 1
C 20 CONTINUE
C 21 IF (KIROS = 1) 24,22,24
C 22 IF (MISPEL) 24,23,24
C 23 IF (NBDYS) 28,28,24
C
C PART 3. REPORTS ERRORS IN BODY LIST.
C 24 WRITE OUTPUT TAPE A.2S, NBDYS.MISPEL.KIROS,(BODY(K),K=1,NBDYS)
C WRITE OUTPUT TAPE A.2S, (REFER(KP),K=1,NBDYS)
C WRITE OUTPUT TAPE 6.2T, (K,NAME(KP),REFER(KP),K=1,30)
C 25 FORMAT (12G4.0, BODY LIST (NBDYS =12,13), MISSPEL =12,13,14)
C 26 FORMAT (12H, REFER =16,70)
C 27 FORMAT (15(3,15H) K3X%HODY4YA,5H%POSITORY5X,5%I3,2X,A6,2X,A6,5X)
C CALL EXIT
C
C PART 4. TRACES OUT „REFERENCE TO BODY„ RELATIONSHIPS
C 28 KA = 2
C KN = 1
C NAME(KP) = NAME(KP)
C NAMER(KP) = 0
C KN = NAME(KP)
C KA = KA + 1
C 29 IF (REFER(KP)) 24,31,30
C 30 NAME(KP) = REFER(KP)
C NAMER(KP) = 0
C KN = NAME(KP)
C KA = KA + 1
C 31 PART 1. PART 2. PART 3. PART 4.
C PART 5. TRACES OUT **BODY TO REFERENCE**  RELATIONSHIP
31 DO 34 K = 1,NBOODYS
32 IF (INNREFR(K) = NAME(K)) 34,33,34
33 NAME(K) = K
34 KK = KK + 1
34 CONTINUE
C
C PART 6. INVERTS NAME TO MANE,STORES BNAME, BMMASS, BRCRIT, AND A
TEMPORARY NEFMRs.
DO 35 K = 1,NBOODYS
   N = NAME(K)
   MANE(K) = N
   NEF(K) = NEFM(K)
   BNAME(K) = NAME(K)
   BMMASS(K) = AMASS(K)
   BRCRIT(K) = CRIT(K)
   NEFMR(K) = NEF
35 CONTINUE
C
C PART 7. FINDS NNREFR REFERENCE FOR BNAME LIST, ALSO TEMP. IBODY
DO 36 K = 1,NBOODYS
   N = NAME(K)
   NRFR = NREFER(N1)
   NNREFR(K) = MANE(NRFR)
36 CONTINUE
C
C PART 8. FINDS IBODY FOR BACKWARD REFERENCE.
DO 37 K = 1,NBOODYS
   NRFR = NNREFR(K)
   IBODY(K) = -K
37 CONTINUE
C
C IBODY LIST IS COMPLETE.
C
C PART 9. WRITES OUT EPHEMERIS LIST TO BE USED IN STORING DATA AND
MAKES FINAL NEFMRS LIST.
40 KK = 1
DO 43 K = 1,NBOODYS
   EFMR(K) = BNAME(K)
   NEFMR(K) = NEFMRS(K)
   KK = KK + 1
43 CONTINUE
NEFMR(NBOODYS) = 0
C
C PART 10. SAVES ELLIPSE DATA
FILE = 0.
IF (FILE) 40,410,41
40 TIM(1) = 2400000.5
FILE = 10.
41 CONTINUE
C
C PART 11. COMPUTE GRAVITATIONAL CONSTANTS. 1.9866 E+30 = KILOGRAMS/SUN MASS
IF ORIGIN BODY HAS AN ATMOSPHERE, SET ROTATION RATE AND ATMOSPHERE RADIUS.
C POSITION THE EPHEMERIDES TAPE AT THE BEGINNING OF THE CORRECT EPHEMERIS
C BY MATCHING THE EPHEMERIS NUMBER READ FROM TAPE (FILE) WITH THE DESIRED
C EPHEMERIS NUMBER (TFILE).
C
400 RESQRO = RESQUK1**2
GKZM = RESQRO/(BMASS(K)+XPRM12)/K1.9866 E301
GKY = SQRTF(GKZM)
REVOL = 0.
RATMOS = 0.
IF (ATMN-BNAME(K)) 51,51,51
51 REVOL = REVOL + XPRM12/4.9151
52 CALL BSFILE13)
53 REA0 TAPE 31 FILE
54 CALL SKFILE131
IF (FILE = TFILEI 54,55,55
GO TO 53
55 BACKSPACE 3
BACKSPACE 3
GO TO 52
C
C PART 12. WRITES THE BNAME LIST ON TAPE 6.
56 IF (OUTPUT TAPE 6) 57,51,51
57 WRITE (OUTPUT TAPE 6,BNAME(K),BNAME(K),NBOODYS)
58 RETURN
END
SUBROUTINE ODLATE
THIS SUBROUTINE COMPUTES THE OBLATENESS ACCELERATIONS (OBLAT) DUE TO AN
AXIALLY SYMMETRIC EARTH. THE 2ND, 3RD, AND 4TH SPHERICAL HARMONIC COEFF-
ICIENTS ARE OBLATJ, OBLATI, AND OBLATD RESPECTIVELY.
COMMON C
DIMENSION A(600), B(700), C(4000),
1 RB(3), OBLAT(3)
EQUIVALENCE
1RA,C ( 1111),CB ( 99111),16K2M ,B ( 3611),
2OBLATJ,A ( 2611),OBLATL,A ( 2711),OBLATH,A ( 2811),
3OBLAT *B ( 7511),R,B ( 1021),RB *B ( 19311),
4RE ,A ( 2511),RSQRD ,B ( 4911),RESQRD ,B ( 711)
AA = RB/31/R
AB = AA*AA
IF IABSFAA) -1.E-6] 1.1.2
AA=O.,
AB=O.
AC = RESQRD/RSQRD
AD = GK2M/RSQRD/AC
AE = OBLATJ*AA
AF = OELATH*AR/R
AG = OBLATI*AC
AH = AE*AC+AG*1.6/1.7+AF*1.4/1.5+AA+AG*1.6/1.7
OBLATJ = AH*RBl31
OBLATJ2 = AH*RE/21
OBLATJ3 = (AH-2.4/1.3+AG*1.6/1.7)*RB(3)-AF*1.4/1.5+R
RETURN
END

SUBROUTINE OUTPUT
ENTS AND RECTANGULAR COORDINATES ARE OUTPUTTED. IF THE OBJECT IS NOT WITH
IN AN ATMOSPHERE IPRESS=O.I. ONE LINE OF DATA IS DELETED. LIKEWISE, ONLY THOSE PERTURBING BOOIES PRESENT HAVE THEIR DISTANCES OUTPUTTED.
COMMON C
DIMENSION A(600), B(700), C(4000),
1 R (81), URBELS (61), VATM (31),
2 BNAME (81), RB(3,81), DIRCOS(3,81),
3 XPRIM (2001), RAMC (51)
EQUIVALENCE
1RA,C ( 1111),ALPHA ,A ( 4911),ALT ,A ( 4111),
2AMC ,B ( 8711),AM ,B ( 9011),AREA ,A ( 6311),
3NAME ,B ( 1211),C ( 1111),ICO ,A ( 1611),
4ICL ,A ( 1611),COSALF ,B ( 4811),COSTRU ,B ( 5311),
5STEPOFFJ,A ( 2311),H2 ,B ( 1511),IMODE ,A ( 1111),
6NBODYS ,B ( 4211),NBODYS ,B ( 4111),ORBELS ,B ( 1161),
7PRESS ,B ( 3311),P ,B ( 8111),PSI ,B ( 3011),
8PSIR ,B ( 39911),PSIR ,B ( 10011),G ,B ( 5911),
9RAMC ,B ( 30311),RB ,B ( 19311),RESQ ,A ( 4811)
EQUIVALENCE
1IR ,A ( 10211),SINL2 ,B ( 4611),SINTRU ,B ( 5211),
2STEPGDA ,A ( 4111),STEPND ,A ( 4211),ITABL ,B ( 2011),
3TRU ,B ( 4011),VATM ,B ( 9711),VG ,B ( 10011),
41V ,B ( 9511),VX ,B ( 9211),VY ,B ( 9311),
51VZ ,B ( 9611),XPRIM (C ( 7111),OUTPUT ,B ( 39911)
DAYJ = IDOFF+2.451+TABLE
ALPHA = ALPHA+7.25977951
REV = REV+ARCOS(RE21)-REV21/6.28318532 + .5
16 CALL CONVT(VX,AMC)
IMODE=IMODE
GO TO (2111), IMODE
1 CODE=+RECTAN
18 CALL CONVT 2
GO TO 4
2 DO 3 K=1,6
3 ORBELS(1) = XPRIM(K+21)
CODE=+SH revolution
TRU=ARCOS(SINTRU,COSTRU)
7 PSI = ATANF(VR111)+VR(21)+VR(31)+V2(21)+AM)/7.25977951
IF (OUTPUT) 19,6,19
6 WRITE OUTPUT TAPE 6 ,1,STEPGDA,STEPND,ORBELS(1),ORBELS(2),VR(21),
NAME(11),CODE(1),NAME(2),XPRIM(11),ORBELS(6),TRU,VR,RB111),KPRIM21),DAYJ21
2RBELS(15),ORBELS(31),VR,RB121),REV,ALPHA(11),PSI(11),URBELS(14),V2,RB131),H2
1 IF (WITHIN AN ATMOSPHERE) COMPUTE DRAG, LIFT, G, ETC., AND PRINT EXTRA LINE.
19 IF (PRESS) 7.14,7
5 ZIFT = G*AREA
G = (PUSH+DRAG+COSALF2+XIFT+SINL2)*XPRIM(21)/8.90655
17 CALL CONVT1VATM,RAMC)
PSIR = ATANF(RU111)+VATM111)+VR(21)+VR(31)+VATM(31)+RAMC(411)+
5.2597795
IF (OUTPUT) 7.14,7
14 WRITE OUTPUT TAPE 6,12,ALT,PSIR,DRAG,VG,G,PSH
IF PERTURBATING BODIES ARE PRESENT, PRINT THEIR DISTANCES AND PRINT THEM.

IF

END

FUNCTION QUAD

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

WHERE A1,B1,C1 ARE THE COEFFICIENTS TO BE USED FOR X BETWEEN X1 AND X2, ETC.

IC IDENTIFIES WHICH DEPENDENT VARIABLE, QUAD, IS BEING SOUGHT.

COMMON C

DIMENSION A(600), B(700), C(4000), 1, COEFN(1901), ICC(5)


I=ICC(IC)

IF(X-COEFN(I)) 2,3,3

I=1

GO TO 1

IF(X-COEFN(I+4)) 5,5,4

5 QUAD = COEFN(I+1)*X+COEFN(I+2)*X*X+COEFN(I+3)

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-SPHERICAL TO RECTANGULAR OR ROTOR ELEMENTS TAKES PLACE IN TUUES.

COMMON C

DIMENSION A(6001), B(7001), C(40001), 1XPRIM1201,XPRIMB201,T6(101),FLOWL1101,AEXIT1101, SIMP1101, ZAREA1101,DSTL1101,IDENT1101,TABLE1200, RMSASS1101,O6000)

EQUIVALENCE 1A .C, 111), (AEXIT, B (33), (AEXIT, A (103),

2AREA1 .A (133), (AREA, B (63), E .C (111),

3DEL1 .A (133), (DEL, B (11), E .C (111),

4DEL .A (43), (DELMAX, A (191), IDNDB .B (391),

5EREF .A (11), (ERLOG, B (17), (EXITA .B (392),

6FLOW .B (51), (FLOWA .B (83), (IDENT .A (123),

7IMODE .A (17), (LSTAGE, A (387), (MODOUTA .A (203),

8INCASE .C (11), (INCASE, A (6003), (SAVE, .C (41),

9INSTAGE .A (33), (PUSHG .B (3913), (RMSASS .A (73),

EQUIVALENCE 15SIMP .A (933), (SIMP, B (23), (TB .A (63),

2ITABLE .C (1911), (TKECK .A (153), (TMAX .B (43),

3ITOL .A (45), (XRPRM2 .C (911), (XRPRM .C (71),

4RETURN, B (4001), (OUTPUT, B (399))
PART 0. SAVE INITIAL DATA IF DESIRED. LOAD STAGE DATA INTO WORKING
STORAGE, ALLOW ADDITIONAL STAGE INPUT.

IF (TOL) 100.99+100
99 DEL = DELMX-TOL
100 IF (NSAVE-NSTAGE) 103,101,103
101 NCASES = NCASE
102 GO TO 2
103 IF (OUTPUT) 103,97+103
97 WRITE OUTPUT TAPE 6,98,NSTAGE,NCASE
98 FORMAT(12H SAVED INITIAL DATA FOR STAGE12,OH OF CASE14,1H.)

104 NSTAGE = NSTAGE + TMAX
105 IF (TMAX) 106,107+107
106 IF (DEL(DELMAX)) 108,109+109
107 DEL = MODF(DEL,DELMAX)
108 IF (DEL) 114,115+115
114 DEL = MINF(DEL,DEL)
115 IF (MODF(MODE)=4) 1,110+110
110 CALL TUEES
116 RETURN

END SUBROUTINE STEP
C SUBROUTINE STEP TESTS FOR THE END OF THE PROBLEM, COMPUTES STEP SIZE, AND
C CONTROLS QUANTITY OF OUTPUT DATA. END OF PROBLEM OCCURS IF TIME = TMAX,
C STEPDGATE=STPPO=STEPH, OR CILLOOK .1 XLOOK. THE LAST OPTION ALLOWS STOP-
C PING ON A DEPENDENT VARIABLE. THE TEST FOR STOPPING AT XLOOK IS NOT MADE
C UNTIL CILLOOK .1 IS GREATER THAN SYLUOK. CONTROL ON QUANTITY OF OUTPUT
C IS
C
C COMMON C
C
C COMMON X4000), BITODJ, (14000),
1 XPRIM2400), DELT (10)
C
C EQUIVALENCE
11A ,C ( 111,141),A ( 101),EAZ,A ( 111),
11B ,C (1111,1),DLMAX,A ( 101),DEL, A ( 431),
3 DLEM ,B ( 111),DOME ,B ( 391),EZ, A ( 191),
4 DLEM, A ( 51),DELLOG, A ( 171),EZ, A ( 151),
552NLOOK, A ( 599),DLOGSW, A ( 91),DLOOK, A ( 81),
4 MODOUT ,A ( 201),NSTAGE, A ( 37),DEL-T, A ( 131),
7 DLE, A ( 581),DLOGS, A ( 311),DLE, A ( 101),
BISTEPD, A ( 411),DSTEMP, A ( 161),DSTEPOM, A ( 421),
9 DSTEPS, A ( 171),DLOGD, A ( 101),DSTABLE, A ( 191),
C EQUIVALENCE
11MAX, A ( 411),DMIN, A ( 181),TMAX, A ( 461),
2 DLOOK, A ( 121),XPRIM, C ( 111),TLOOK, A ( 111),
3 DSTART, A ( 241),SSWITCH, A ( 161),OUTPUTB, A ( 399),
CHECK(1B,1) = ABSF(A-B) - ABSF(A+C)
C C PART 1. TEST FOR END OF THE PROBLEM (MAXIMUM PROBLEM TIME OR MAXIMUM
C NUMBER OF STEPS).
C STEPDGATE = STEP0 = 0.
C OUT = OUTPUT
C IF (ABSF(A>B)) 1,113+113
1 DOME = 1.0
112 CALL OUTPUT
C IF (OUTP0T) 26,111,126
111 WRITE OUTPUT TAPE 6,12,NSTAGE
2 FORMAT(12H STAGE12,1H COMPLETED.//) GO TO 26
3 IF (STEPSGATE=STEPNO=STEPH) 1,114+114
4 CALL OUTPUT
C WRITE OUTPUT TAPE 6,5,STEPH
5 FORMAT(22H HOSTEPGATE=STEPNO=STEPH=6,.)
6 CALL EXIT
PART 2. COMPUTE STEP SIZE (DELT) AND CONTROL OUTPUT.

N=1
A3 = (A2-A1)\times RATIO+R2
A2 = \exp(-0.3)\times 0.5
\text{IF} (ABS(A1) > 0.026) \text{ABS} (SWITCH) 0.8, 0.6
Delt = \text{SIGN}(\exp(A4), \text{DELT})
\text{IF} (\text{DELT}/H2-3.) 10,10.9
\text{DEL} = 3.\times H2
\text{MODOUT} = \text{MODOUT}
\text{GO TO} (11,12,13,21,\text{MODOUT}) 21,12,12
\text{MODOUT} = 2
\text{DEL} = \text{MIN} - \text{XPRIM}(1)
\text{GO TO} 16
\text{IF} (\text{DELT} = \text{XPRIM}(1) - \text{MIN}) 15,15,14
\text{MODOUT} = 4
\text{GO TO} 21
\text{DEL} = \text{DEL} - H2
\text{16 SPACES} = \text{INIT}(\text{DELT}/\text{DELT}1+\text{SIGN}1.9, \text{DELT}/\text{DELT}1)
17 \text{IF}(\text{SPACES}) 20, 18, 20
18 \text{CALL OUTPUT}
N=2
\text{DEL} = \text{DELMAX}
\text{IF} (\text{ABS}(\text{DELT}) - \text{ABS}(\text{DELT})) 19,16,16
\text{DELT} = \text{SIGN}(\text{DELT}, \text{DELT})
\text{GO TO} 16
\text{DELT} = \text{DEL}/\text{SPACES}
\text{GO TO} 23
21 \text{IF}(\text{MODISTEPRD}, 23) 23,22,23
22 \text{CALL OUTPUT}
N=2

PART 3. SEARCH FOR CLLOOKX = XLOOK UNLESS LOOKX=0.

23 \text{IF}(\text{LOOK X}) 27,42,27
27 \text{LOOK X} = \text{LOOK X}
\text{LOOK SW} = \text{LOOK SW}
\text{OUTPUT} = 1.
\text{GO TO} (44,45,46,47,48,49)
44 \text{CALL OUTPUT}
45 \text{IF}(\text{SWITCH}) 32, 28, 33
28 \text{IF}(\text{SW LOOK} - C(\text{LOOK SW}) 29,29,42
29 \text{XTOL} = \text{XTOL}+\text{ABS}(\text{LOOK})
\text{IF}(\text{XTOL}1) 30,30,31
30 \text{XTOL} = \text{XTOL}
31 \text{SWITCH} = -1.
\text{GO TO} 41
32 \text{SWITCH} = 1.
\text{ASSIGN 43 TO MODE}
\text{OVER} = 0.
F = D.
33 \text{SLOPE} = (C(\text{LOOKX})-\text{DLX}X)/H2
\text{GO TO MODE, (43,35)}
43 \text{IF}(\text{SLOPE} \times (\text{LOOK X}) - X \text{LOOK}) 350,41,41
350 \text{ASSIGN 35 TO MODE}
35 \text{IF}(\text{ABS}(C(\text{LOOK X})-X \text{LOOK}) - \text{XTOL1}) 36,36,37
36 T=1.
36 \text{IF}(\text{OUT}) 63,46,63
46 \text{OUTPUT} = 0.
\text{CALL OUTPUT}
47 \text{IF}(\text{T}) 61,41,61
47 \text{IF}(\text{OUT}) 62,51,62
50 \text{WRITE OUTPUT TAPE 6,64, \text{LOOK X}, C(\text{LOOK X}), H2, LOOK X, SLOPE}
48 \text{FORMAT}(\text{SLOPE}(14,41) = 141.5.8, 31.8\text{ CONVERGENCE TROUBLE})
\text{DEL} =
\text{IG15.8,14H SLOPE OF C14,13HI VS. TIME = G15.8//}
\text{GO TO 62}
47 \text{IF}(\text{OUT}) 62,50,62
50 \text{WRITE OUTPUT TAPE 6,48,41, LOOK X, C(LOOK X)}
48 \text{FORMAT}(\text{SLOPE}(14,84) = 141.5.8, 81.8//)
60 \text{LOOK X} = 0
62 \text{XTOL} = 0.
\text{SIGNAL} = 1.
\text{SWITCH} = 0.
\text{DONE} = \text{END}
\text{NSTAGE} = 0
\text{NSTAGE} = \text{NSTAGE}
\text{DEL} = \text{DEL} (\text{NSTAGE})
49 \text{CALL INPUT(LOOK X, \text{TABLE})}
\text{IF}(\text{DONE}) 100,42,110
110 \text{IF}(\text{OUT}) 26,11,12,26
37 \text{SIGN} = \text{CHECK}(\text{LOOK X}, \text{LOOK X}, C(\text{LOOK X})
\text{IF}(\text{SIGN}) 40,40,38
40 \text{OVER} = 1.
\text{GO TO} 400
38 \text{IF}(\text{COVER}) 400,401,400
401 \text{XGUESS} = \text{C(LOOK X)} \times \text{SLOPE} \times \text{DELT}
\text{IF}(\text{CHECK}(\text{LOOK X}, \text{LOOK X}, \text{XGUESS})) 402,41,41
402 \text{F} = \text{F13}
\text{IF}(\text{F} = 2.) 0.0, 400,403
403 \text{SLOPE} = \text{SLOPE} / F
\text{IF}(\text{SLOPE}) 404,40,404
404 \text{DELT} = \text{SIGN}(\text{ABS}(\text{LOOK X}, \text{LOOK X})) / \text{SLOPE}, \text{SIGN} \times H2)
41 \text{DLX} = \text{C(LOOK X)}
42 \text{IF}(\text{ABS}(\text{MAX} - \text{XPRIM}(1) - \text{ABS}\text{S})(\text{DELT})) 25,26,26
25 \text{DELT} = \text{MAX} - \text{XPRIM}(1)
\text{GO TO} (26,27,24,26,26, \text{MODOUT})
26 \text{DEL} = \text{DEL} - \text{DELT}
26 \text{OUTPUT} = \text{OUT}
\text{RETURN}
\text{END}
SUBROUTINE STOATA

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

C COMMON C

C DIMENSION A(6000), B(7000), C(40000),
1 PHA(112), AMASS(301), XPRIM(2001),
2 CDEPN(170), IEC(4),
3 AK (3), XDOT(100), IND(3),
4 REFER(122), RCRT(301), AM(4),
5 RMASS(10)

C EQUIVALENCE
1 IAI, IC(111), (AI, 511), (AMASS, A(347)),
2 IAI, IA(291), JAM, IA(551), IB, IC(11111)),
3 IBDYDA, IA(143), (CDEPN, A(4071), (CONSTU, A (321)),
4 IBDYMA, IA(317), (IOFF, I, IECRT, A(197)),
5 IERLMT, IA(143), (ETOL, IA(301), (GASFA, A(461)),
6 IEC, IA(1531), (IMODE, IA(11), (NO, IA(601)),
7 IILKDOK, IA(91), (JUDOUT, IA(201), (FREQ, IA(21)),
8 INSTAGE, IA(31), (OBLAT, IA(271), (OBAT, IA, 281)),
9 IOBLAT, IA(207), (PNAME, IA, 2071), (CRIT, IA(377)),
10 IEC, IA(1531), (IMODE, IA(11), (NO, IA(601)),
11 IREFF, IA(117), (FNAME, IA(271), (FNAME, IA(501)),
12 HPRM, IA(117), (RMASS, A(731))

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

C THE FOLLOWING NH STATEMENTS LOAD THE BODY NAMES INTO THE MACHINE.
PHA(1) = HSUN
PHA(2) = HMERCU
PHA(3) = SHVENUS
PHA(4) = SHEARTH
PHA(5) = 8HZMARS
PHA(6) = 8HZJUPITE
PHA(7) = 8HZSATURN
PHA(8) = 8HZURANUS
PHA(9) = 8HZNEPTUN
PHA(10) = HPLUTO
PHA(11) = HMOON
PHA(12) = HHEARTH

C FILL OUT SUN REFERENCE LIST. INITIALIZE Mass ARRAY.
10 2 K = 1, 10
RMASJL(K) = 1.
2 REFER(K+1) = PHASE(1)
REFER(12) = PHASE(11)

C FILL OUT EARTH REFERENCE LIST.
REFER(1) = PHASE(4)
REFER(4) = SHERO
REFER(11) = PHASE(4)

65
C LOAD THE REMAINING STANDARD DATA.
AK(1) = 0.5
AK(2) = 0.5
AK(3) = 1.0
AMASS(1) = 1.0
AMASS(2) = 1.0/120000.0
AMASS(3) = 1.0/40845.0
AMASS(4) = 1.0/32931.3
AMASS(5) = 1.0/308600.0
AMASS(6) = 1.0/1047.39
AMASS(7) = 1.0/3500.0
AMASS(8) = 1.0/28869.0
AMASS(9) = 1.0/18889.0
AMASS(10) = 1.0/400000.0
AMASS(11) = AMASS(4)/81.335
AMASS(12) = AMASS(4)/81.335
AU = 1.49599
AW(1) = 1.0/6.
AW(2) = AW(1) + AW(1)
AW(3) = 1.0 - (AW(2) + AW(1) + AW(4))
BBEFC = PNAME(4)
COEFO = PNAME(4)
COEFN(1) = -1E20
COEFN(19) = 1E20
CONST = 1.0 
ECOOL = 1.0 
ETOL = 0.01
DTDFJ = 244.44
EBEF = 1E-6
ERLMT = 3E-6
GASFAC = 20.064881
ICC(1) = 185
ICC(2) = 185
ICC(3) = 185
ICC(4) = 185
IMODE = 1
IND(1) = 1
IND(2) = 3
IND(3) = 1
LOOKSW = 711
MDDUT = 4
NEG-8
NSTAGE = 1
OBLATJ = 1.42345 E-3
URLATJ = -5.75 E-6
URLATD = 7.075 E-6
RRC1(1) = 1.0 E+20
RRC1(2) = 1.0 E+8
RRC1(3) = 6.14 E+8
RRC1(4) = 9.25 E+8
RRC1(5) = 5.78 E+8
RRC1(6) = 4.81 E+10
RRC1(7) = 5.46 E+10
RRC1(8) = 5.17 E+10
RRC1(9) = 8.31 E+10
RRC1(10) = 3.81 E+10
RRC1(11) = 1.60 E+10
RE = 6378165.
SFDP = 86400.0
SQRK = 2.959122083 E-4
STEPMX= 100.0
STEPS = 1.
TFILE = 1.
XOPT(1) = 1.0
XPRIM(2) = AMASS(11)
XTOL = 5E-8
WRITE OUTPUT TAPE 6,3
3 FORMAT (15H0STANDARD DATA.)
RETURN
END
SUBROUTINE TESTTR
C SUBROUTINE TESTTR MAY BE CALLED FOR ONE OF TWO REASONS, (1) TO TEST FOR AND
C POSSIBLY TRANSLATE THE ORIGIN WHEN IMOE IS =+1 OR (2) TO CHANGE THE
C VARIABLES OF INTEGRATION WHEN IMOE IS =-. A TRANSLATION OF THE ORIGIN
C OCCURS WHEN THE OBJECT MOVES INTO A SPHERE OF INFLUENCE WHICH IS SMALLER
C THAN ANY OTHERS IT MAY ALSO BE IN-. WHEN THIS HAPPENS, THE NAME OF THE NEW
C ORIGIN IS MOVTD 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),
! APRIM1(100,2),XPRIM1(100,2),XWHL0E(16),YPRM1(13,8),VXI3),
! ORBELS(0,0),BMASS(0),BNAME(0),RBI3(0),RBCRIT(0),R(N)
C
EQUIVALENCE
11A C ( 111), (AMC ,B ( 87)), (ASYMPT,A ( 71)),
21B C ( 112)), (OBMASS ,B ( 137)), (NAME ,B ( 122)),
3(NCHAMP ,B ( 299)), (DELT ,B ( 31)), (OKAF ,B ( 363)),
4(IMODE ,A ( 11)), (NBODY ,B ( 41)), (ORBELS ,B ( 116)),
5(ORBCRT ,B ( 145)), (RQ ,B ( 193)), (REVS ,A ( 408)),
6(FO ,B ( 102)), (SORDK ,B ( 35)), (TABE ,C ( 911)),
7(TIMAX ,B ( 4)), (TRSFER ,B ( 9)), (TRU ,B ( 401)),
8(RITEST ,A ( 54)), (VFRM ,B ( 2413)), (VX ,B ( 923)),
9(XPRIM ,C ( 711)), (XPRIMB ,B ( 911)), (XWHL0E ,B ( 110))
EQUIVALENCE
10(OUTPUT ,B ( 399))
C
IMOE = IMOE
IF (IMOE) 12,12,1
C
C IF IMOE IS =+, TEST FOR TRANSLATION OF THE ORIGIN.
1 CHAMP = 1.E+30
DO 4 JB=1,NBODYS
IF (R(JB)-RBCRIT(JB)) 2,4,4
2 IF (CHAMP-RBCRIT(JB)) 4,4,3
3 CHAMP = RBCRIT(JB)
NCHAMP = JB
4 CONTINUE
IF (NCHAMP=1) 26,26,5
5 TRANSF = 1.0
8 BTEMP = BNAME(1)
BNAME(1) = NAME(NCHAMP)
BNAME(NCHAMP) = BTEMP
TEST = 0.
REVS = 0.
IF (OUTPUT) 6,9,6
9 WRITE OUTPUT TAPE 6,10,BNAME(NCHAMP),BNAME(1)
10 FORMAT (28HORIGIN IS TRANSLATING FROM A6,A4M TO A6)
6 CALL EPHERNS
DO II K=1,3
VX(K) = VX(K)-YEPHM(K,NCHAMP)
RB(K) = RB(K,NCHAMP)
XPRIM(K+2,1) = VX(K)
XPRIM(K+5,1) = RB(K)
XRIM(K+2,1) = 0.
XPRIM(K+5,1) = 0.
11XWHL0E(K+3) = RB(K)
GO TO 20
C
C IF IMOE IS --, CHANGE THE VARIABLES OF INTEGRATION.
12 DO II K=1,3
XPRIM(K+2,1) = XWHL0E(K)
XPRIM(K+5,1) = XWHL0E(K+3)
XPRIM(K+2,1) = 0.
XPRIM(K+5,1) = 0.
VX(K) = XWHL0E(K)
13 RB(K) = XWHL0E(K+3)
GO TO 16,14,15,IMOE
14 CODE = SHOWRT
IMODE = 1
GO TO 18
15 IMOE = 3
GO TO 17
16 IMODE = 2
17 CODE = 6 RECTAN
18 NCHAMP = 1
9 WRITE OUT TAPE 20,7,20
7 WRITE OUT TAPE 6,19,7,CODE
19 FORMAT (33HINTEGRATION MODE IS CHANGING TO A6)
20 GO TO (21,26,26),IMODE
21 CALL CONTV(3,A,KM,
22 GO TO 21,26,26),IMODE
23 CALL CONVT2(XPRIM,AMC,
24 GO TO 21,26,26),IMODE
25 CALL CONV12
C IF ORIGIN TRANSLATION CAUSES PATH TO LIE NEAR AN ASYMPTOTE, CHANGE
C INTEGRATION VARIABLES TO RECTANGULAR IF THEY ARE ORBIT ELEMENTS.
C IF (ORBELS1=1) 24,24,22
22 IF (ABS(TRU)-2.3/QRXFR(ORBELS1)) 24,24,23
23 ASYMPT = 1.0
GO TO 15
24 DO 25 J=1,6
25 XPRIM(J+2,1) = ORBELS1(J)
26 IF (TRSFER) 27,28,27
27 CALL INPUT (101,3,26)
29 CALL ORDER
28 RETURN
END
SUBROUTINE THRUST
C
THIS ROUTINE COMPUTES X, Y, Z THRUST ACCELERATIONS. THE THRUST VECTOR IS
C
ASSUMED COINCIDENT WITH THE LONGITUDINAL AXIS OF THE VEHICLE, WHICH IS
C
ORIENTED TO THE RELATIVE WIND VELOCITY BY THE ANGLE OF ATTACK (ALPHA) AND
C
THE ROLL ANGLE (BETA). ALPHA IS ASSUMED TO BE A QUADRATIC FUNCTION OF TIME
C
WHEREAS BETA IS ASSUMED TO BE CONSTANT.
C
REVOLV IS THE EARTH'S Rotation RATE IN RADIANS/SEC (7.29211585E-5) AND THE
C
FACTOR 8589934592. ~ 2**33 IS REMOVED TO PREVENT OVERFLOW.
C
COMMON C
C
DIMENSION A(600), B(700), C(4000),
1 FORCE(3), PARA(1), VATM(3), P(3), IND(3), RMC(5), RBE(3), X(100)
C
C
EQUIVALENCE
1A ,C ( 11),IEXIT ,B ( 31),I ALPHA ,A ( 49),
2E ,C ( 1111), BETA ,A ( 501), ICOSALPHA, B ( 481),
3ICOSBETA,B ( 491), IEXIT ,B ( 3921), IFLOW, B ( 59),
4IFORCE ,B ( 661), IND ,A ( 601), IPAR, B ( 601),
5ISEMPN, B ( 501), IPRESS, B ( 331), IP, B ( 81),
6IMPG, B ( 3911), IUSH, A ( 161), IRMAGC, B ( 3931),
7IATMPR,B ( 231), IRO ,B ( 1931), IREVOLV ,B ( 211),
8IA,B ( 1021), IRESURG, B ( 451), ISIMP, B ( 21),
9ISINALE, B ( 461), ISENBIT, B ( 471), IATM ,B ( 971),
C
EQUIVALENCE
1IV3 ,B ( 1001), IVEG, B ( 1011), IXY ,B ( 921),
2IVY ,B ( 931), IVZ, B ( 941), IX ,B ( 401),

SINBET = SINF(BETA/57,295795)
COSBET = COSF(BETA/57,295795)
VATM1 1 = VATM1 + REVOLVX*RBC1
VATM1 2 = VATM1 3 = VATM3
C
3 CALL CONVT1(VATM,RANC)
4 ALPHA = QUADXI1/57.2957795
SINALF = SINF(ALPHA)
2 FORCE = PMAG*SINALF*COSBET*PARA
3 RETURN
END

SUBROUTINE TUBES
C
THIS ROUTINE COMPUTES THE RECTANGULAR POSITION AND VELOCITY COMPONENTS
C
WITH RESPECT TO THE EARTH MEAN EQUINOX AND EQUATOR OF 1950.0 FROM THE
C
LATITUDE, LONGITUDE, AZIMUTH, ELEVATION, ALTITUDE, TOTAL VELOCITY, AND
C
TIME. ALSO, WHEN TRICK DOES NOT EQUAL ZERO, A NON-DRAG VERTICAL STEP OF
C
SIZE TRICK IS MADE IN CLOSED FORM (STATEMENTS 2 TO 4). THE INTEGRATION
C
WILL THEN BEGIN AT TIME EQUAL TO TIME+TRICK WITH THE ORIENTATION SPECIFIED
C
BY THE ABOVE FOUR ANGLES AND THE COMPUTED VALUES OF ALTITUDE AND VELOCITY.
C
FOR THE CLOSED FORM APPROXIMATION, A CONSTANT FLOW RATE (FLOW), VACUUM
C
SPECIFIC IMPULSE (SIMP) AND ENGINE EXIT AREA (EXIT) ARE ASSUMED KNOWN.
C
THE ATMOSPHERIC PRESSURE IS TAKEN TO BE THE SEA LEVEL VALUE.
C
COMMON C
C
DIMENSION A(600), B(700), C(4000),
1 SINA(4), COSA(4), ANGLEB(4), XPRIM(200)
C
EQUIVALENCE
1A ,C ( 11), IEXIT ,B ( 31), IALT ,A ( 41),
2AE1 ,A ( 351), R0 ,C ( 1111), IDTOFF, A ( 231),
3IELAY ,A ( 361), IFLOW, B ( 51), IKRM2, B ( 361),
4LAT ,A ( 331), ILONG ,A ( 361), IUBLAT, A ( 261),
5IBLATN, A ( 401), IRE ,A ( 251), IRESURG, B ( 711),
6ISTORE ,A ( 391), ISIMP, B ( 21), ISEQ, A ( 441),
7ITFROST ,A ( 41), ISTEPNO, A ( 421), ITRICK ,A ( 261),
8IOTOFF , A ( 241), IVEL, A ( 371), IEXPRIM, C ( 711),
9IOUTPUT ,B ( 391)
C
EQUIVALENCE
1QLAT, LALT, ( QLONG, LONG),

68
ALTI = 0.
VEL1 = VEL
DELI = 0.
DEL = 0.
ASSIGN 1 TO NGO
DAYS = DTOPFJ = 2432582.5
GREEN = MODU((100.0*75526+985674364)DAYS+2.9015E-13DAYS+2
1.729115859+41*(SUFP+SUXPRMII11)=57.297795,360.1)
SINA1 = SINFQLAT/57.297795
IF (TBLATN1 = 102,101,102
101 RADIUS = RE + ALT
GO TO 8
102 RADIUS = 6.956783+28/SQRTF1.9933065830.00699341685*SINA11+2)ALT
GO TO 8
1 XPRIM(6) = COSA12)*COSA41)+RADIUS
XPRIM(7) = SINA12)*COSA41)+RADIUS
XPRIM(8) = SINA11)+RADIUS
RMASS = XPRIM(2)
XPRIM(2) = XPRIM(2)-FLOW+TKICK
IF (OUTPOT) I2,11,13
11 WRITE OUTPUT TAPE 6,3,STEPG,STEPNO,LAT,LONG,ALT,ALT
XPRIM(11),VEL,RMASS,XPRIM(J)=6,8,
3 FORMATT=60STEMPS=5,1H +4D,1H LAT=1IA15.8,7H LONG=q15.8,6M ALT
11.5,8,7H ELEV=1G5,8,6H ALT=1G5,8,6H TIME=1G5,8,6H VEL=1G5,9,6
7H RMASS=1G5,8,4;2H=1G5,8,5X,2HT=1G5,8,4X,2ZT=1G5,8
12 IF (TKICK) I3,50,2
2 XPRIM(1) = XPRIM(1)+TKICK
B1 = LOG/FIRMASS/XPRIM(2)
SIMPSL = SIMP-AEXIT/LOW+10332.75
VELI = VEL+1MP5*9.996598+1*TKICK
ALT1 = TKICK+VEL1=0,9996598!*SIMPSL+1-B1*XPRI1(2)1
1 (RMASS-XPRIM(2)))
1 RADIUS = RADIUS + ALT
GREEN = GREEN + 7.292115855*TKICK*57.297795
ASSIGN 5 TO NGO
GO TO 8
5 XPRIM(6) = COSA12)*COSA41)+RADIUS
XPRIM(7) = SINA12)*COSA41)+RADIUS
XPRIM(8) = SINA11)+RADIUS
GO IF (TBLATN1 = 6.7,6
6 DEL = ATANFIC2-1.1/(C3-1.1*SINA11)/COSA41)*57.297795-QLAT
7 DEL2 = RADIUS/COSA1)+COSA11)*ROTATE+ROTATE*57.297795
DEL = DEL2 + DEL2
ASSIGN 10 TO NGO
ANGLE(1) = QLAT + DEL
ANGLE(1) = QLONG + GREEN
ANGL(3) = AZI
ANGLE(4) = ELEV
DO 9 I=1,4
SINA1 = SINFANGLEG111157.297795
9 COSA1 = COSFANGLE(3157.297795
C1 = 9.9*RESURD/RADIUS+RADIUS+OBLAT
C2 = C1*SINA11)/COSA41)*(-6)
C3 = C1+SINA11)+COSA11)-7,2
G = 1*2INFANGLE/RADIUS
GO TO NGO (1.1,5,10)
10 COS1 = COSA11)+SINA1)+COSA1)*COSA1)+SINA1)
COS2 = COSA11)+SINA1)
XPRIM(3) = VEL1+COS1+COS2)-COS2*SINA1)+(XPRIM(7)+ROTATE
XPRIM(4) = VEL1+COS1+SINA1)+SINA1)+(XPRIM(6)+ROTATE
XPRIM(5) = VEL1+SINA1)+SINA1)+COSA1)*COSA1)+COSA1)
RETURN
END

SUBROUTINE TAPE
SUBROUTINE TAPE USES THE MASTER MERGED EPHEMERIDES TAPE (TAPE 3 AT LEWIS) WHICH CONTAINS ONLY
THAT DATA NEEDED AT EXECUTION TIME. THIS MINIMIZES TAPE HANDLING DURING
EXECUTION. 2 EPHEMERIS FILES ARE ON TAPE 9, FIRST FILE HAS DATA AND IS
IDENTIFIED BY THE SECOND WORD OF EACH 254 WORD RECORD. FIRST WORD IS THE
DUMMY FORKAN COMPATIBLE WORD, SECOND WORD=2. THE SECOND FILE IS ONLY 2
WORDS LONG, FIRST WORD IS FORKAN COMPATIBLE, SECOND WORD=3.
MASTER FILE 1 -- PLANETS (EXCEPT MERCURY AND EARTH), SUN, MUN, AND
EACH EPHEMERIS COMPILATION REQUIRES A SET OF INPUT 300 DATA, THE FIRST.piece
OF DATA WRITTEN ON A FILE IS THE FILE IDENTIFICATION NUMBER: FILE.
EACH
FILE IS NUMBERED CONSECUTIVELY STARTING WITH FILE=1. SINCE MUN, DATA IS IN
TERMS OF EARTH RADIO, THE CONVERSION OF MUN DATA TO A.U. IS MADE BEFORE
WRITING ON TAPE 3. THE COMMON USED IN SUBROUTINE TAPE IS LOCAL AND ALL
HUT TAPES IS CLEARED BY A FINAL CLEARING LOOP.
FUNCTION COMPARFA(A,B) IS EQUIVALENT TO (A-B) BUT WILL NOT OVERFLOW.
NORMAL INPUT - ELIST, TDESC, TEND, TAPE3
C
C
ELIST- THE BCD LIST OF EPHEMERIDES DATA NAMES TO BE PLACED ON
TAPE 3. THE NAMES ARE READ FROM CARDS, AND IS USED TO
MAKE THE TMAKE LIST. ELIST IS NOT CHANGED IN STORAGE UNTIL
THE FINAL CLEAR FOR THIS SUBROUTINE.
TMKE- THE LIST OF EPHEMERIDES NAMES WITH Duplicates Dropped AND
ZERO SPACES CLOSED IN. AS THE EPHEMERIDES ARE FINISHED THE
NAMES ARE ERASED FROM THIS LIST.
TMAKE- LIKE TMAKE BUT IS HELD FOR OUTPUT.
TDESC- THE BEGINNING DATE EXPRESSED AS A JULIAN DAY.
TEND- ENDING DATE EXPRESSED AS A JULIAN DAY.
INTVAL- THE APPROX. NUMBER OF DAYS COVERED BY ONE SET OF COEFF. IT
IS USE TO DECIDE WHICH DATA ARE TO BE ENTERED UDLATE. THE
DOUBLE ENTRIES PERMIT FASTER OPERATION IF REVERSE OF
INTEGRATION IS REQUIRED FOR ANY REASON.
EDATE- JULIAN ENDING DATE FOR THE MASTER EPHEMERIS.
ERTOA- EARTH RADI IN A.U.
C
C
COMMON C
DIMENSION
1 C (700), TMAKE (12), LIST (30),
2 EDAT (12), INTVAL (30), KTAG (12),
3 ELIST (11), TMAKE (12), INTVA (2),
4 PNAME (30), DATUM (252), DATUM (21,12)

C EQUVALENCE
1 TAPE3 (C (21)), TMAOE (C (31)),
1 KTAG (C (41)),
1 FILE (C (16)),
1 ELIST (C (17)), TMAOE (C (29)),
1 TEND (C (30)), PNAME (C (31)),
1 KMAKE (C (81)), TDATE (C (89)),
1 INTVAL (C (81)), KTAG (C (81)),
1 FILE (C (189))

C B COMPARF: Bl = (A+B) + (1-A*B)
REWIND 3
DO 1 J=1,4000
1 (K) = 0.0

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) = JSUN
PNAME(2) = GMERCURY
PNAME(3) = SVENUS
PNAME(4) = GMARS
PNAME(5) = GOJUPITER
PNAME(6) = GHSATURN
PNAME(7) = GHURANUS
PNAME(8) = GSATURN
PNAME(9) = SHPLUTO
PNAME(10) = GAUDE
PNAME(11) = GHERTH

C PART 2. SET UP JULIAN DATES ENDING EACH EPHEMERIS.
EDATE(1) = 2451872.5
EDATE(3) = 2451848.5
EDATE(4) = 2451020.5
EDATE(6) = 2473520.5
EDATE(8) = 2473520.5
EDATE(9) = 2473520.5
EDATE(10) = 2449613.5
EDATE(11) = 2451848.5

INTVA = 30000
INTVAL(1) = 8
INTVAL(2) = 5
INTVAL(3) = 15
INTVAL(4) = 44
INTVAL(5) = 330
INTVAL(6) = 825
INTVAL(7) = 1211
INTVAL(8) = 1172
INTVAL(9) = 1101
INTVAL(10) = 2
INTVAL(11) = 15
FILE = 1.
ERTDAU = 4.26546512 E-5
MOON = 0
LI = 1

C PART 2B. CALL INPUT AND SEE IF TAPE IS TO BE MADE. INPUT MUST ALWAYS
C MAKE TAPE3=0.0 IF TAPE IS TO BE MADE.
TAPES = 1.
8 CALL INPUT (300,C,LIST)
IF (TAPES) 63,63
3 IF (FILE=1) J0,10,20
10 CALL SKFILE (9,2)

C PART 3. TAPE IS TO BE MADE SO MOVE EPHEMERIS LIST TO TMAKE AND
C TO TMAKE (FOR OUTPUT), CANCEL ANY ZERO OR DUPLICATE NAMES.
20 KOUNT = I
DO 6 K=1,KOUNT
TMAKE(K) = 0.
6 TMAKE(K) = 0.
4 DO 5 J=1,KOUNT
IF (COMPARF (ELISTIK), TMAKE(J-1)) 5,6,5
5 CONTINUE
TMAKE(KOUNT) = ELISTIK
TMAKE(KOUNT) = ELISTIK
KOUNT = KOUNT+1
6 CONTINUE
KOUNT = KOUNT - 1

C PART 4. FIND INPUT ERRORS.
7 IF (BEGIN=2437202.5) 66,9,9
9 KOUT = 2
11 ERROR = 0.
WRITE TAPE 3,FILE
DO 21 J=1,KOUNT
KTAG(J) = 0.
21 KTAG(J) = 0.
12 DO 13 K=1,20
IF (COMPARF (PNAME(K), TMAKE(J))) 13,16,13
13 CONTINUE
C PART 5. PRINTER THE MISPALETTED NAMES AND OTHER ERRORS.
14 PRINT IS.
TMAKE(J), TBEGIN, TEND
WRITE OUTPUT TAPE 6.
10 TMAKE(J), TBEGIN, TEND, PNAME(K),
I,EDAT(AK), K=1,20
15 FORMAT (23H TROUBLE ON TAPE 3 MAKE /
2X,6,10H T BEGIN= F10.1,8H
1 T END= F10.1,210)
ERROR = 1.
GO TO 21

70
PART 40. CHECKS DATES AND STORES INDEX FOR MOON SO THAT EARTH RAOII CAN BE CONVERTED TO A.U.

16 IF (10-KI) 18,17,18
17 MOD = J
18 KTAG(J) = K
19 IF (EDATE(KI- TEND)) 14,21,21
21 CONTINUE
ASSIGN 36 TO NS1
IF (ERRG3) 42,22,68

C PART 6. FIX UP A TAG KTAG 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() DO 23 J=1,KOUNT K = KTAG(J) K0 = XMNOF(KHAMP, INTERVAL(K))
23 CONTINUE
KHAMP = KHAMP + 10
DO 24 J=1,KOUNT K = KTAG(J) K0 = INTERVAL(K) / KHAMP

C PART 7. LOCATE FILE 2 ON TAPE 9.
25 READ TAPE 9, KFILE
26 IF (KFILE - 1) 31,32,29
27 IF (KFILE - 31) 20,21,28
28 BACKSPACE 9
BACKSPACE 9
CALL BSFILE(9)
GO TO 25
BY PASS A FILE.
GO TO 25
29 CALL BSFILE(9)
PART 8. THIS IS CURRENT FILE ON TAPE 9, READ DATA. THERE CAN BE UP TO 10 SETS OF DATA PER RECORD. A RECORD SET OF DATA IS 21 WORDS.
31 BACKSPACE 9
32 READ TAPE 9, KTAPE, (DATUM(KJ), J=1,21)
33 CONTINUE
34 READ TAPE 9. KTAPE, (DATUM(KJ), J=1,21)
35 CONTINUE
36 READ TAPE 9. KTAPE, (DATUM(KJ), J=1,21)
37 CONTINUE
38 LI = J
BACKSPACE 9
BACKSPACE 9
GO TO 32
39 IF (TDATE(KJ) - TDATE(KI+1) - TBEGIN) 40,41,40
40 DO 41 KJ=1,21
41 DATUM(KJ,AK) = DATUM(KJ,AK)*ERTOAU
42 CONTINUE
43 IF (MODN) 44,45,44
44 DATUM(KJ,MODN) = DATUM(KJ,MODN)*ERTOAU
45 CONTINUE
46 GO TO 25

C PART 9. IS THIS A SATISFACTORY STARTING POINT? QUESTION MARK.
THE 1ST SET OF DATA FOR EACH PLANET MUST PRE DATE TBEGIN.
PART 9 IS EXECUTED ONLY ONCE.
47 DO 48 J=1,KOUNT
48 K = KTAG(J)
49 IF (TDATE(KJ) - TDATE(KI+1) - TBEGIN) 50,51,50
50 CONTINUE
51 SYT = TBEGIN - TDATE(KJ+1) - TDATE(KI)
52 DO 53 KJ=1,21
53 DATUM(KJ,AK) = TDATE(KJ-1)
54 DO 55 KJ=1,21
55 DATUM(KJ,AK) = DATUM(KJ,AK)*ERTOAU
56 IF (SYT) 57,58,57
57 WRITE TAPE 3. (DATUM(KJ,AK), KJ=1,21)
58 IF (TEND - TDATE(KJ,AK)) 59,60,59
59 TMAKE(J) = 0.
60 DO 61 KK=1,KOUNT
61 IF (TMAKE(KK)) 62,63,62
62 WRITE OUTPUT TAPE 6, 61, FILE, TBEGIN, TEND, KOUNT, TMAKE(KK), 1, KK-1, KOUNT)
63 WRITE TAPE 3. (DATUM(KJ,AK), KJ=1,21)
64 IF (TEND - TDATE(KJ,AK)) 65,66,65
65 TMAKE(J) = 0.
66 DO 67 KJ=1,21
67 DATUM(KJ,AK) = DATUM(KJ,AK)*ERTOAU
68 CONTINUE
69 WRITE OUTPUT TAPE 6, 61, FILE, TBEGIN, TEND, KOUNT, TMAKE(KK), 1, KK-1, KOUNT)
70 CONTINUE
61 FORMAT(20HDEPHDERISERIS COMPLETED. FILE=FS,6H, FROM FILE,3,3H TO 1 FILE), 4H FOR 12, 18h BODIES AS FOLLOWS / 12t25,66/ FILE = FILE + 1.
END FILE 3
GO TO 2
62 WRITE TAPE 3, FILE
63 REWIND 3
64 WRITE TAPE 9
65 TAPES = 3
66 DO 64 J=3,4000
67 U(J) = 0.
68 CONTINUE
70
CONTINUE
GO TO 32
PRINT 67, TBEGIN
WRITE OUTPUT TAPE 6, TBEGIN
FORMAT(33H TBEGIN PREDATES 2437202.5, IT IS F10.1)
CONTINUE
REWINO 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
PZE
BCD 1BSFILE
BSFILE SXO ↔4,1
SXO ↔4,2
SXO ↔4,4
XEC* $1EG1
TSX $1(RER1)
LXD BSFILE-2,4
CLA* 1,4
TXL BSFILE-2,4
ANAF A07000
STA BTI1
STA BTI2
LXD BSFILE-2,4
CAL 2,4
ANAF =077777777770000
ERA =0007400000000
TNZ ONEARG
CLA* 2,4
PZE BACK
PX 1,1
AXC ↔1,4
XEC* $1(TG0)
BT1 BT1 BT1 **
TRA ↔1
BSF BSFA ↔2,2
XEC* $1(RDS)
XEC* $1(FSR)
AXC ↔1,4
XEC* $1(TG0)
BT2 BT2 BT2 **
TRA CHECK
TXL BSF,1,1
XEC* $1(RDS)
BACK AXC ↔1,4
XEC* $1(TG0)
AXC ↔1,4
XEC* $1(TRC)
NOP
AXC ↔1,4
XEC* $1(FEF)
NOP
LXD BSFILE-4,1
LXD BSFILE-3,2
LXD BSFILE-2,4
TRA 3,4
CHECK TXL BACK,1,1
LXD BSFILE-2,4
CLA ERR*1
STO 0
CLA* 1,4
LDQ* 2,4
ERR TXL 8,4
TXI BACK,0,14
PZE BSFILE-2,0,ERR
ONEARG CLA BSFILE-2
ADD =01000000
STO BSFILE-2
LXD CHECK,1
TRA BTI2-2
A07000 OCT 7000
END
ENTRY SKFILE
PZE
PZE

SKFILE SXD = -3.1
SXO = -3.2
SXO = -3.3
TSX $1(ER)1,4 CHECK LAST READ
TEFA ++1
TEFB ++1
LXD SKFILE-1,4
CLA* 1,4 PICK UP THE TPE NUMBER
TSX $1(ER)1,4 SET UP THE TAPE ADDRESSES
LXD SKFILE-1,4 LOAD IT AGAIN—MAN
CAL 2,4 IS THERE A SECOND ARGUMENT
ANA =0777777700000
ERA =0000000000000
TNZ ONEARG NO SECOND ARGUMENT
GOGO CLA* 2,4 PICK UP THE SECOND ARGUMENT
TZE BUMP+1 DID SOME DUMMY WANT NO FILES
LOOP SUB =01000000
RDS XEC* $10DS1 READ THE TAPE
TCOA
TCOR
TEFA BUMP DID WE HIT
TEFB BUMP AN END OF FILE
TRA RDS GO READ SOME MORE
BUMP TNZ LOOP
LXD SKFILE-3,1
LXD SKFILE-2,2
LXD SKFILE-1,4
NDP TRCA ++1 TURN OFF TAPE CHECK
TRCB ++1
TRA 3,4 ONEARG CLA SKFILE-1
ADD +01000000 SET UP XR4 FOR PROPER RETURN
STO SKFILE-1
PKO 0.0 SET UP FOR ONE FILE
TRA RDS

COUNT 1200 00020
REM INPUT ROUTINE USING ARITHMETIC STATEMENTS. CF NASA TN D-1092 00030
LBE INPUT 6 00040
ENTRY INPUT 00050
REM THIS IS SUBROUTINE INPUT. ITS CALLING SEQUENCE 00060
REM CONTAINS THREE ARGUMENTS—AN IDENTIFICATION 00070
REM CODE NUMBER, THE FIRST LOCATION RELATIVE TO WHICH 00080
REM ALL DATA IS TO BE LOADED, AND THE FIRST LOCATION 00090
REM OF A TABLE TO BE USED BY THE ROUTINE. 00100
REM 00110
REM 00120
REM INCLUDED IN THIS ASSEMBLY ARE SUBROUTINES 00130
REM 1 INPUT 00140
REM 2 CHARTR 00150
REM 3 CLEAR 00160
REM 4 COMPAR 00170
REM 5 ERROR 00180
REM 6 LOCK 00190
REM 7 NAME 00200
REM 8 NUMBER 00210
REM 9 STORE 00220
REM 10 TABLE 00230
REM 11 TEST 00240
REM 12 ACCUM, FIX, FLT, BINARY 00250
REM 13 PRINX 00260
REM 14 READ 00270
REM 00280
INTAPE PZE 0.7 LEWIS INPUT TAPE NOT STD. 00290
OUTAPE PZE 0.8 FORTRAN STANDARD OUTPUT TAPE 00300
INDEX PZE 0.6 STORAGE FOR IRA 00310
PZE 0.7 IRB 00320
PZE 0.8 IRC 00330
BCI 1+INPUT 00340
INPUT SXD INOX,1 SAVE INDEX REGISTER A 00350
SXO INOX+1,2 SAVE INDEX REGISTER B 00360
SXO INOX+3,3 SAVE INDEX REGISTER C 00370
NZT* 1.4 IF THE IDENTIFICATION NUMBER IS Z 00380
TRA 4,4 RETURN TO THE CALLING PROGRAM. 00390
CLA '1035 00400
ADD 2,4 2,4 IS THE BASE LOCATION.
STA SET 00410
STA LOC1 00420
STA LOC4 00430
CLA TSXBS OPEN BACKSPACE GATE 00440
STD* $(LINK) CALL CHAIN WILL BACKSPACE 00450
LOCA CLA* 1.4 1.4 IS THE IDENTIFICATION NUMBER. 00460
STA NREG1 00470
ANT 36,1 INITIALIZE 36 00480
STZ 1+1,1 LOCATIONS 00490
TIX 1+1,1 TO ZERO. 00500
STD ILOC1 MAKE NON-TIERD. 00510
CLA 3,4 3,4 IS THE LOCATION OF THE TABLE. 00520
STA LOCFC PREPARE 00530
STA NREG1 00540
ADD '1035 THE 00550
STA LOPA ARGUMENT STORAGE 00560
STA LOCBL 00570
TSX CLEAR 4 CLEAR THE YAR REGION.
LOCAL CLA =0076100000000 INHIBIT READING UNTIL 00580
STO READ. ARRAY RECORD REFRESHED 00590
ART 43,£ 43 FORCES RECORD TO BE FILLED 00600
SKX 1,2 IN CHRCTR 00610
REM LOOK AT THE FIRST CHARACTER ON THE FIRST CARD 00620
REM IN SEARCH OF A $ SIGN. 00630
LOCAL TXT CHRCTR-4 CHECK FOR A $ SIGN 00640
LOCAL, STD WORD 00650
TXT COMPAR,4 OCT 242517630000 D, E, FILE FLAG, T 00660
LX4 HNEG1,4 ZER0 IF $0 HAS BEEN READ. 00670
TXX **+2,4,0 TXI **+1,2,4 BEFORE $0 ADD 4 TO INDEX 2. 00700
TXH ERKH,2,7 JUNK 00720
TXH SGNOUT,2,6 $17 BEFORE $0. FILE FLAG. OFF 00730
TXH **+9,2,3 $E BEFORE $D. 00740
TXX READ,4,1,4 CRASH READ GATE 00750
TRA LOLA1 SHOULD NOW HAVE $0 CARD 00760
TXH LOLA0,2,4 FIRST $D. 00770
TXH LOLA1,2,3 $F AFTER $0. 00780
TXH ERKH,2,2 $E AFTER $D. 00790
REM 00800
LOCAL CLA READ,4,4 $D AFTER $D. TEST IF BUFFER 00810
TXX ERKH,4,0 OVERWRITTEN 00830
REM THIS IS THE PROGRAM RETURN. 00840
RTN LXX INUX,1 RESET INDEX A. 00850
LXX INUX,1,2 RESET INDEX 0. 00860
INUX,+2,4 RESET INDEX 0. 00870
TRA 4,4 RETURN TO CALLING PROGRAM. 00880
REM HUNT FOR THE * = SIGN OF THE $ DATA CARD. 00890
LOCAL CLA =0076100000000 INHIBIT READING UNTIL 00900
STD READ. $DATA FIELD SCANNED 00910
TXT CHRCTR-4 00920
TXT COMPAR,4 00930
BCH 1,+00000 CARD SCANNED OUT. 00940
TRA **+3,2,2 JUNK 00950
TRA ERKH JUNK 00960
TRA LOLAD ALPHABETIC 00970
TRA ERRK NUMERIC 00980
SKX ALF,4 + SIGN 00990
REM USE ALF MODE TO TEST ALL CHARACTERS. 01000
REM COMES HERE WHEN + SIGN HAS BEEN FOUND. GET THE 01010
REM IDENTIFICATION NUMBER FROM THE CARD. 01020
LOCAL LXX 1,4 01030
TXT **+4,4,43 01040
TXH LOLA0,4,42 CARD SCANNED OUT. 01050
TXT CHRCTR-4 01070
TEXT COMPAR,4 01080
BCH 1,4=0,0 SET TO BY PASS. 01090
TRA **+9,2,2 JUNK 01100
TRA ERKK ALPHABETIC 01120
LOCAL TXT BINARY,4 FORM BIN MD IN VAR 01130
TRA LOLA1 BLANK 01140
SKX ERSK,2 MINUS SET TO BY PASS. 01150
TRA LOLAF PLUS NO EFFECT. 01160
STU $12H DOLLARS 01170
REM COMES HERE TO CHECK THE REGION CODE AND THE 01180
REM VALUE APPEARING ON THE DATA CARD. 01190
LOCAL CLA VAR COMMA 01200
TXX ERKK DATA SET NO. MISSING 01210
ALS 16 01220
STD ** SAVE IDENT AT TABLE(1). 01230
NREGI ** SUB ** PLACE FIRST ARG IN THIS ADDRESS. 01240
TXX RTN 0 IF CALL CODE $0 DATA CODE 01250
STI ALF ALF - 0 MEANS NO ALF INTU. 01260
SXa NREG1,0 INDICATE $0 DATA IS READ. 01270
REM INST. 0600 ALSO EXECUTED AT READ. PLACED THERE BY CHRCTR 01280
TXXS RTN READ,1,4 HERE SNEAK PAST READ. GATE 01290
SKX TEST,0 01300
TRA LOCAN 01310
REM 01320
REM COMES HERE IF IT WAS A $ TABLE CARD. 01330
LOCAL TXT TABLE,4 01340
TRA LOCAN3 01350
REM 01360
REM COMES HERE IF AN ALPHABETIC CHARACTER WAS FOUND. 01370
LOCAL TXT NAME,4 01380
TXN SET-1 01390
LXX JKI,1 01400
REM IF JKI DID NOT INCREASE THEN 01410
SKX TOL3TJtK,2 SAVE JKI FOR NEXT TEST. 01420
CLA ILDC SAVE SIGN OF TABLE ENTRY. 01430
STO ILLCI 01440
TRA LOCAN2 01450
REM 01460
REM JKI,2 PREPARE TO ACCUMULATE THE NUMBERS 01470
SET CLA **+2 IN THE PSEUDO ACCUMULATOR. 01480
STO TEMP 01490
CLA ILLC 01500
TXX LOLAUF MINUS MEANS FLOAT THE NUMBER. 01510
TXH FLT,4 01520
TRA LOLAM 01530
REM 01540
REM COMES HERE IF NUMERIC FIELD. 01550
LOCAL TXT NUMBER,4 01560
STD TEMP 01570
LOCAM TSX ACCUM.4 ACCUMULATE RESULTS IN ACC.
TSX CLRASM.4
LXI WORD.4
PRA 0,4 * WORD IN ACC FOR LOGAR
TAL LOGAR.458 NOT COMMA
TAM LOGAR.459 NOT COMMA
LKD J4.2 COMMA
CLA ACC
STZ ACC
LOQ ILOGI IS THIS VARIABLE FIXED POINT.
TOP LOI NEGATIVE IS FIXED POINT.
TSX FIX.4
LOG1 STD **.2 STORE THE NUMBER RELATIVE TO BASE.
LOG2N TRI **.11.1 RAISE STORING INDEX BY ONE.
LOG2N SKD J4.2 SAVE IT.
TXL OPER.1 ANY OPERATORS LEFT OVER.
E RRL TSX ERROR.4
BCI 1,0(E) ANY DATA LEFT OVER.
TAZ ERRKL ANY DATA LEFT OVER.
REM CALL THIS THE SWITCH HOUSE.
LOGAD TSX CLEAR.4
LOGAF TSX COMPAR.4
LOGAG TSX COMPAR.4
BCI 1,00000
TRA **.2,2
TRA LOGAR
TRA LOGAL ALPHABETIC
TRA LOGAL NUMERIC
TRA LOGA ( SIGN
TRA LOGAL DECIMAL
LOGAR LXD OPER.1 ANY OPERATORS LEFT OVER.
SUB =HID00000 SPLIT OFF & FROM OTHERS
TPL LOGA.
REM WHAT KIND OF OPERATOR IS THIS?
TSX COMPAR.4
BCI 1,0-?.0
TXH ERRX.2,5 REMOVE THE JUNK.
TXM LOCAN.2,4 COMMA
SKD OPER.1,4 COMMA
TRA LOGAP AFTER BOTH OPERANDS HAVE BEEN FOUND
REM COMES HERE IF OCT OR ALF MODE.
LOGAT TSX CMCHR.4
TSX COMPAR.4
BCI 1,0D000
TRA **.5,2
TRA EARL JUNK
TRA LOGA A CHARACTER
TRA LOGA Q CHARACTER
REM COMES HERE IF EMPTY PARENTHESES WERE FOUND.
TSX CMCHR.4 JSIGN, GET NEXT CHARACTER.
TOP **.2 MINUS FOR NEW CARD
TXM TEST.4 INSERT COMMA IF NEEDED.
CLA ILOG1 PREPARE TO GET VALUE OF
STD ILOC1 CURRENT LEFT SIDE.
LXTL LOGA.4 Current LEFT SIDE.
TRA SET REM COMES HERE IF OCTAL MODE.
LOGAU TSX CMCHR.4
SUB =HID0000
TRA LOGAL SIGN
LOCAV LDQ VAR
ROL 3 REPLACE TOP 3 B1TS
LGR 3 BY NEXT OCTAL CHARACTER
ROL 3 PUT IN BOTTOM OF MQ
STQ VAR
REM COMES HERE WHEN J IS FOUND.
LOGAX TSX CMCHR.4
TOP **.2 MINUS FOR NEW CARD
TXM TEST.4 CHARACTER TO INC
LXI WORD.4 CHARACTER TO INC
TXL LOGAV.47 OCTAL DIGITS
TXL ERKJ.458 ALPHABETIC, JUNK, B, 9.
TXH ERX.459 SPLITS I
LOGAX LXD J4.2 COMMA
CLA VAR
TRA LOGI CONVERT THE NUMBER TO BINARY.
LOGAY TSX BINARY.4
REM COMES HERE IF OCTAL MODE.
LOGAZ TSX CMCHR.4
TSX COMPAR.4
BCI 1,00000
TRA **.5,2
TRA EARL JUNK
TRA LOGA ALPHABETIC
TRA LOGA NUMERIC
REM COMES HERE WHEN J IS FOUND.
**LOC6A**

```
LXX VAR+,1 J SIGN 02540
TXA ERRK+,0 ALF COUNT WAS ZERO. 02550
SKO ALF+1 02560
TSK CLEAR+,4 02570
TSX CHKCTR+,4 PULL THROUGH CHARACTERS AND STORE 02580
SUB =O17 FILE FLAG, NEVER NEG. 02590
TIZ ERRB COUNT WENT PAST E O JOB. 02600
TSX STORE+,4 THEY ONE AT A TIME. 02610
TXX ++,1,1 GO BACK TILL NCHAR = 1 02620
LXX J+,1 02630
LXD MSHIFT+,4 02640
CAL BLANK 02650
LCG =+,1 02660
DIRS VAR+,1,1 FILL IN PARTIAL WORD WITH BLANKS. 02670
```

**LOC8B**

```
AXT 1,4 TAC TO 1 02680
CLD J+,1,2 02690
CLA J PREPARE TO STORE ALF WDS 02700
STD LOKLC1 02710
LOCBC SKO J+,1,2 02720
CLA VAR+,1 02730
LDC **,2 02740
TXI +1,4,1 J = J + 1 02750
LOCBC1 TXI LOKRC,2,2 JK=JK+1 02770
TXI LOKCR,2,1 02780
LOCBO TIZ ALF 02790
TSX CLEAR+,4 LUCK AT NEXT CHARACTER. 02800
TXX CHKCTR+,4 MINUS FOR NEW CARD 02810
TSX TES+,4 PUT IN COMMA IF NEEDED. 02820
SUB >HUGDO, GU RAISE AND STORE JK+. 02830
TIZ LOKAN 02840
REM 02850
REM THESE ARE ERROR CALLS 02860
ERRB TXS ERROR+,4 02870
BCI 1,J0B 02880
ERRD TXS ERROR+,4 02890
BCI 1,J1D 02900
ERRJ TXS ERROR+,4 02910
BCI 1,J0J 02920
ERRK TXS ERROR+,4 02930
BCI 1,J0X 02940
ERRM TXS ERROR+,4 02950
BCI 1,M1J 02960
ERRU TXS ERROR+,4 02970
BCI 1,U1J 02980
REM 02990
REM 46 COMES HERE AFTER $0 03000
REM 03010
LOCBG CLA =UU761000000 NOP 03010
STD *S(,1) close backspace gate 03020
TRI LOKLC return 03030
REM PURPOSE OF SEND CARO IS TO PROTECT FORIEGN DATA FROM 03040
REM BACKSPACE WHEN CHAIN IS CALLED. 03050
REM 03060
REM END OF THE MAIN SEGMENT 03070
REM THIS IS A ROUTINE TO BACKSPACE THE INPUT TAPE WHEN A 03080
REM CALL CHAIN IS GOING TO SPILL THE BUFFER. 03090
REM THIS ROUTINE IS EXECUTED FROM CHAIN VIA THE ONE 03100
REM WORD SUBROUTINE (LINK) WHICH CONTAINS EITHER TSX OR NOP 03110
REM 03120
TXXBS TXS LOKBS,4 TO BE STORED AT (LINK) 03120
LOCBS SKO **,4 SAVE INDEX 4 03130
CLA INTAPE INPUT TAPE NUMBER 03140
CALL $(10G) SELECT INPUT TAPE 03150
XEC* $(05R) BACKSPACE IT 03160
AXT **,4 RESTORE INDEX 03170
TRA 1,4 RETURN TO THE CHAIN ROUTINE 03180
EJECT 03190
REM THIS IS A SUBROUTINE CHKCTR. IT STORES SUCCESSIVE 03200
REM CHARACTERS FROM THE CARD AT LOCATION WORD, READS 03210
REM SUCCESSIVE CARDS INTO THE ARRAY RECORD, AND PRINTS 03220
REM ** TYPE CARDS, THE FIRST CHARACTER FROM A NEW CARD 03230
REM IS STORED IN WORD WITH A MINUS SIGN. 03240
REM 03250
REM 03260
REM 03270
CHKCTR SKO TEMP-10,2 03280
SKO TEMP-17,4 03290
LXX I+,2 CARO COL COUNT, SAW COUNT 03290
TXH **,2,83 TUG EARLY TO READ. 03300
XEC READ. GATE MAY BE CLOSED 03310
LDQ Q HAS UNUSED CHARACTERS FROM BEFORE 03320
CLA SLN ZERO OR $ GOES TO TAG 03330
LOCDA ALS 6 SHIFT LEFT I CHARACTER 03340
SLM TAG CLEAR OR PRELOADS TAG 03350
LOCDB LXX ALF+,4 NONZERO MEANS ALF MODE. 03360
LOCDC LXX LOKC0,2,43 SAW COUNT GIVES COL. B1 = 43. 03370
LOCDC0 LXX LOKC0,2,4 WAS COL 80 PROCESSED. 03380
LOCDD PXD 0,0 CLEAR ACCUMULATOR. 03390
LXL 6 SHIF T NEXT CHARACTER INTO ACC. 03400
TXX LOKC0,2,14 COUNT DOWN BY 14 03410
LDQ RECORD+,3,2 LOAD NEXT WORD 03420
TRI *=,1,2,69 JUMP BACK COUNTER. 03430
LOCGE TXH LOKCF+,0,4 RETURN IF ALF MODE. 03440
PAK 0,1 MOV E CHAR. INTO INDEX 1 03450
TXH LOKCF+,1,40 TKG MEANS GOOD CHARACTER. 03460
TXH LOKCF+,1,47 TKG IF BLANK 03470
TXH LOKCF+,1,48 TKG IF GOOD CHARACTER. 03480
TXH LOKCF+,1,42 TKG IF GOOD CHARACTER. 03490
ZET TAG HERE ON $ 03500
TRA PRINT HERE ON $ GO PRINT 03510
TRA TOLCA $ G0ES TO TAG. 03520
REM 03530
```
LOC08

REM CHARACTER AN0 TESTS IT AGAINST THE CHARACTERS
REM THEN INDEX 2 CORRESPONDS TO THE TOTAL TESTS
REM EN0 OF THE SAP SUBROUTINE CHRCT.

REM EN0 OF THE SAP SUBROUTINE CLEAR. IT INITIALIZES
WAIT FOR QUIET REAO BUFFER.

XEC+ $ITE5) WAIT FOR QUIET INPUT BUFFER.

OUT BCI 06:1 END OF FILE INPUT TAPE JOB COMPLETE

REM REM END OF THE SAP SUBROUTINE CHRCT.

EJECT REM THIS IS SUBROUTINE CLEAR, IT INITIALIZES
REM NECESSARY PARAMETERS FOR SUBROUTINE STORE.

CLEAN SXD $ave J V0 SET J TO 0.

STZ VAR CLEAR VARJI.

SXD MSHIFT,0 RESET MSHIFT.

TRA 14 RETURN TO CALLING PROGRAM

REM REM END OF THE SAP SUBROUTINE CLEAR.

REM REM THIS 1s FUNCTION COMPARE, IT EXAMINES THE CURRENT
REM CHARACTER AND TESTS IT AGAINST THE CHARACTERS
REM FOUND IN THE ARGUMENT, ALPHABETIC AND NUMERIC
REM SPLITs ARE MADE IF THE CHARACTER IS NOT FOUND
REM IN THE ARGUMENT. THESE TESTS ARE COUNTED AND
REM THE NUMBER LEFT IN INDEX 2 CORRESPONDS TO THE
REM SUCCESSFUL TEST IF NO TEST IS SUCCESSFUL
REM THEN INDEX 2 CORRESPONDS TO THE TOTAL TESTS ".

COMPAR SXD 14 USE FIRST ARGUMENT IN CALLING

LOCDX PXD 0,0

LKL 6 PULL IN 1ST TEST CHARACTER.

LZE LOLOD DUE IF ZERO.

CAS WOLOD CHECK TEST WORD AGAINST CARD

TXI LOLOD,2 1 characterize.

TRA LOLOD EQUA.

LORC8 TXI LOLOD,2 1 CHARACTER.

TRA 24 PROGRAM RETURN.
LOC00 CLA 2.2
USE SECOND ARGUMENT IN THE CALLING
PDX 0.1
SEQUENCE INCREMENT AS THE TEST
TXN LOC0C,1024 FOR ALPHABETIC-NUMERIC SPLIT.
SKD LOP0,12
RECEIVES INCREMENT
LXA WORD,1
CHARACTER TO IRA
TLX LOC0C,19
NUMERIC
THX LOC0C,157
SPECIAL 0 ZONE, $X
THX LOC0C,149
ALPHABETIC 0 ZONE, 6
TIX "+1,1,16
KNOCK OFF 12 ZONE EXCEPT +
TIX "+1,1,15
KNOCK OFF 12 ZONE EXCEPT 0
REM * AND - SIGNS WILL BE 111110, WILL BE 111110.
THX LOC0C,19
SPECIAL
LOC0F TXI LOC0C,2,1
ADJUST IRB FOR ALPHABETIC
LOC0F TXI LOC0C,2,2
ADJUST IRB FOR SPLIT
REM
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 SNA **2,4
SAVE SOURCE
XEC+ SITE80
WAIT FOR QUIET BUFFERS
ART **4
CLA 1,4
GET PRINT ARGUMENT
STQ OUTBUF+5,4
ART 1,1
CAS 6
TRI **3
S THROUGH V
TXI **1,1
R
Messa PXD BLANK+4,1 A THROUGH N
ANA =0,17
AJS 16
ACL M65A
STA 102,
ART 4,9
IO2.
LDQ ***4
STQ OUTBUF+5,4
TIX "+1,1,16
COUNT CHARACTER PART.
ART 1,1,4
CLA 19,2
BLANK
STQ OUTBUF+19,2
TIX "+1,2,1
LDQ \n * PICK UP *
LXO 1,2
SAW COUNT
TIX "+1,2,71
BACK UP IF OVER 71
TRI **1,3,1,169
ROTATE ACCORDING TO CHAR PART.
TIX "+1,2,16
COUNT CHARACTER PART.
STQ OUTBUF+19,2 STORE ACCORDING TO RESIDUAL
TIX PRIN4,4 PRINT THE *
XEC+ SITE80 WAIT FOR THE * TO BE PRINTED
LXO ERRR,4 PICK UP ERROR SWITCH.
TXX L0L0UN+1,40 NUM ZERO MEANS TRY NEXT SET
ART 120,4 BYPASS MARK
SAD BLANK+4 MARK BYPASSED CARDS
LXA NREL,4 NONZERO IF THIS $DATA CARD.
TXX **2,4,0
XEC+ SITE80 WAIT FOR QUIET BUFFER
ART 19,2
CLA BLANK
STQ OUTBUF+19,2
TIX "+1,2,1
LDQ \n * PICK UP *
LXO 1,2
SAW COUNT
TXX **1,2,71 BACK UP IF OVER 71
TRI **1,3,1,169 ROTATE ACCORDING TO CHAR PART.
TXX "+1,2,16 COUNT CHARACTER PART.
STQ OUTBUF+19,2 STORE ACCORDING TO RESIDUAL
TXX PRIN4,4 PRINT THE *
XEC+ SITE80 WAIT FOR THE * TO BE PRINTED
LXO ERRR,4 PICK UP ERROR SWITCH.
TXX L0L0UN+1,40 NUM ZERO MEANS TRY NEXT SET
ART 120,4 BYPASS MARK
SAD BLANK+4 MARK BYPASSED CARDS
LXA NREL,4 NONZERO IF THIS $DATA CARD.
TXX **2,4,0
XEC+ SITE80 WAIT FOR QUIET BUFFER
ART 19,2
CLA BLANK
STQ OUTBUF+19,2
TXX "+1,2,1
LDQ \n * PICK UP *
LXO 1,2
SAW COUNT
TXX **1,2,71 BACK UP IF OVER 71
TRI **1,3,1,169 ROTATE ACCORDING TO CHAR PART.
TXX "+1,2,16 COUNT CHARACTER PART.
STQ OUTBUF+19,2 STORE ACCORDING TO RESIDUAL
TXX PRIN4,4 PRINT THE *
XEC+ SITE80 WAIT FOR THE * TO BE PRINTED
LXO ERRR,4 PICK UP ERROR SWITCH.
TXX L0L0UN+1,40 NUM ZERO MEANS TRY NEXT SET
ART 120,4 BYPASS MARK
SAD BLANK+4 MARK BYPASSED CARDS
LXA NREL,4 NONZERO IF THIS $DATA CARD.
REM REM END OF THE SAP SUBROUTINE ERROR.
EJECT
REM THIS IS SUBROUTINE LOOK, IT SEARCHES THE TABLE
REM FOR THE NAME STORED AT LOCATION VARN. 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 LDFE
ART 4,4
J K = Z IN INDEX B
LDFEA CAL **2
CAL TABLE13.
TXX LOP0,4
NO ENTRY IN THIS VARIABLE
STD LOP0,4
DECREMENT HAS NEXT
ACL +037777700000
ANA +037777700000
ENTRY LOC, SAVE DEC
SUB J 2,1
ONLY, CHECK ENTRY LENGTH.
TNZ L0C0F IF NOT THE SAME, LOOK AT NEXT ENTR
PD0 0,2
JM = JK IN INDEX C.
POX 0,4
J

LOCFB CLA VAR+1,15:3 IF VAR AND THIS.
LOCF1 CLA **:* ENTRY AGREE.
TRA **2,15 IF SQ, CHECK REST OF NAME.
TRI **2,4,1 RAISE 2M BY ONE.
LOCFD TXI LOCFD-I,2,25:3 IF NOT SQG, GO TO NEXT ENTRY.
TXI **1,1,1 RAISE 2J BY ONE.
LOCFE TXL LOLF1,1,1 FINISHED IF J1 IS GREATER THAN J.
TXS **:* CLEAR IF THE ENTRY AGREES.
LOCF CLA LUCFA 05620:3
STD ILUC SAVE COMMON INDEX AT ILOC.
LOCFG TXD TEMP-12,4 PREPARE TO RETURN.
TRA 1,4 RETURN TO THE CALLING PROGRAM.
REM 05650:3
REM 05670:3
REM END OF THE SAP SUBROUTINE LOOK.
REM 05680:3
REM 05690:3
REM THIS IS SUBROUTINE NAME. IT IS USED TO
REM 05700:3
REM MEMORY LOCATIONS BY REFERRING TO THE TABLE.
REM 05720:3
REM 05730:3
REM 05740:3
NAME SDK TEMP-20,4 SAVE INDEX C.
REM GET THE REST OF THE VARIABLE NAME. STOP AT ANY
REM 05750:3
REM NON ALPHANUMERIC CHARACTER.
REM 05770:3
LOCG TSX STORE+,4 05780:3
LOCG TSX CHAR+,4 05790:3
TCP **2,1 MINUS FOR NEW CARD.
TCP **2,1 TEST,4 COMMA MAY BE NEEDED.
YNZ **3,1 LOOK FOR ZERO. IF ZERO, MAKE IT
REM 05800:3
ACL =NOZERO00 A LETTER O
REM 05820:3
STD WORD 05830:3
LOCGE TSX COMPAR+,4 05850:3
BCI 1=10000 05860:3
TRA **5,2,1 05870:3
TRA LOLCF JUNK OR OPERATORS.
TRA LOLGB NUMERIC OR ALPHABETIC.
TRI LOLDG 1 SIGN.
TRI ILUC1 * SIGN.
REM 05880:3
REM 05890:3
REM 05900:3
REM GO TO THE TABLE LOOKUP ROUTINE IF AN = SIGN.
REM 05920:3
REM 05930:3
REM OR AN OPERATOR WAS FOUND.
REM 05940:3
LOCGF TXD LOOK+,4 FIND THE NAME IN TABLE.
TZE ERRAT NAME WAS FOUND IN TABLE IF NON-ZER
REM 05950:3
REM 05960:3
REM 05970:3
REM 05980:3
REM 05990:3
REM 06000:3
REM GET TO THE TABLE VARIABLE LOOKUP ROUTINE IF A
REM 1 SIGN WAS FOUND.
REM 06020:3
REM 06030:3
REM 06040:3
REM CONVERT THE INDEX TO BINARY.
REM 06050:3
REM 06060:3
REM 06070:3
REM GET THE NUMERICS FOR THE INDEX TO THE VARIABLE.
REM 06080:3
LOCGJ TSX CHCTR+,4 06090:3
TXL LUG1+,2,19 NUMERIC.
TXL ERC1,1,27 JUNK.
TXL ERKC1,1,29 JUNK.
TXS CHCTR+,4 J SIGN. GET NEXT CHARACTER.
TCP **2,1 MINUS FOR NEW CARD.
TCP TEST,4 COMMA MAYBE NEEDED.
TXS COMPAR+,4 06140:3
BCI 1=20000 06150:3
TRA **4,2,1 06170:3
TRA LOLDG OPERATORS.
TRA ERR1 ALPHABETIC AND NUMERIC.
STZ ILUC1 * SIGN.
REM 06180:3
REM 06190:3
REM 06200:3
REM 06210:3
LOCGK CLA VAR COMPUTE STORING INDEX.
ACL ILUC 06220:3
ACL 0,2 STORE ADDRESS AT DECREMENT WITHOUT
REM 06240:3
REM 06250:3
REM 06260:3
REM 06270:3
REM 06280:3
REM 06290:3
REM 06300:3
ERRC TSX ERROR+,4 06310:3
BCI 1=0(G) 06320:3
REM 06330:3
REM 06340:3
REM 06350:3
REM 06360:3
REM END OF THE SAP SUBROUTINE NAME.
REM 06370:3
REM 06380:3
REM 06390:3
NUMBER SDK TEMP-29,4 SAVE INDEX C.
SDK KNT1+,4 INITIALIZE.
STZ KNT3 THE SUBROUTINE.
STZ KNT1 BRANCH PARAMETERS.
STZ KNT4.
STZ TEMP.
TRA LOG1B.
REM 06400:3
REM 06410:3
REM 06420:3
REM 06430:3
REM 06440:3
REM 06450:3
REM 06460:3
REM 06470:3
LUCFA TSX CHCTR+,4 TCP **2,1 MINUS MEANS FROM NEW CARD.
TXS TEST,4 MINUS MEANS FROM NEW CARD.
TXS TEST,4 MINUS MEANS FROM NEW CARD.
REM 06480:3
REM 06490:3
REM 06500:3
LOC16 TSX COMPAR,4
BCI 1.80000
TRA **y,2.2
TRA LOCHK JUNK OR AN OPERATOR
TRA ERR0 ALPHABETIC
TRA LOCHG NUMERIC
TRA LDLNE E
CLA KN11 Decimal point
TNZ **# ZERO MEANS THIS IS THE SECOND POINT
TXS ERROR,4
BCI 1.x(OH)
STA KN12
STZ NEXP
TRA LOCHG
LOC16 CLA NEXP COUNT THE NUMBER OF DIGITS BEHIND
ADD =1035 THE. IF THERE IS ONE
STA NEXP.
LOC16 IKA KN111
TSH LOCHG,1.10 DO NOT ACCUMULATE PAST 10
ZLE LOCHG DO NOT COUNT LEADING ZEROS.
AFL LOCHG
LOC162 TXS ***1.1 COUNT TOTAL NO. OF DIGITS
SST 0.730
TRA LOCHG
REM COMES HERE WHEN THE EXPONENT FIELD IS
LOC16 CLA KN11 ENCONTNERED.
TNZ LOCHG THERE MUST BE AT LEAST ONE DIGIT
TXS ERROR,4 BEFORE THE E OF AN E FORMAT NUMBER
BCI 1.x(OH)
LOC16 IKA SEE IF EXPONENT DIGITS HAVE ARRIVE
TRA **+2
LOC16 CLS KN13 SEE IF EXPONENT DIGITS HAVE ARRIVE
TNZ LOCHG-2 NON ZERO MEANS SIGN IS OPERATOR.
STD TEMP STORE SIGN OF EXPONENT.
CLA KN14
TNZ ERR0 NONZERO MEANS MORE THAN 1 EXP SIGN
SST KN14.2 MAKE NONZERO.
LOC16 TSX CHCTR,4
TOP **+,2 MINUS MEANS FROM NEW CARD
TXS TEST,4
TXS COMPAR,4
BCI 1.x++.000
TRA **+,2.2
TRA LOCHG-2 OTHERS
TRA ERR0 ALPHABETIC
TRA LOCHG NUMERIC
TRA ERR0 DECIMAL
TRA LOCHG MINUS
TRA LOCHG PLUS
REM COMES HERE WHEN AN OPERATOR WAS FOUND.
LOC16 CLA KN11 TEST FOR THE PRESENCE OF EXPONENT.
TZE ERR0 ZERO MEANS NO EXPONENT CAME.
LOC16 CLA KN12
TZE **+2
STZ NEXP
CLA KN11 SEE IF MORE THAN TEN NUMBERS HAVE
SST =10835 BEEN CONVERTED
TPL **+2 IF SO, USE THE DIFFERENCE IN THE
PRA 0.0 COMPUTATION OF THE EXPONENT.
SST KN12 NUMBER OF NON-ZERO MEANS SIGN.
ADD TEMP
STD NEXP
REM MANTISSA IN VAR AND THE EXPONENT IS IN NEXP.
CLA VAR
TZE L0CHL SHORT CUT IF ZERO.
LOC G03300000000 CHARACTERISTIC FOR LOW BITS
LGR 8 LOW 8 BITS TO REG
RQL 8
RSL 0 BRING SIGN
STA VAR +G0330000000 CHARACTERISTIC FOR HIGH BITS
FAD VAR
FAN
ST0 VAR
STA VAR THE EXPONENT
AT 1.2
L0D =1. PUT 1 IN MQ
LOC16 LBT EXPONENT IN ACCU
TRA LOCHM FOUND NO BIT.
TSH ERR0.2.6 EXPONENT EXCEEDS 63
STD VAR-2 THIS FORS 10 **NEXP
FMP TAD+1.2 SAVE IN MQ
XCA VAR-2
LOC16 AMS 1 TZE LOCHM 10**NEXP FINISHED.
TRI LOCHL.2.1 MULTIPLY IF PLUS.
LOC16 TRI LOCH0 MULTIPLY IF PLUS.
FMP VAR
FNR
FAD VAR
TRA LOCHG
LOC16 STQ VAR-2 DIVIDE IF NEXP IS MINUS.
PDP VAR-2
XCA
LOC16 LKD TEMP-23.4 RETURN TO CALLING PROGRAM.
TRA 1.+4 REM THESE ARE THE ERROR CALLS FOR SUB NUMER.
07570
ERRE TSX ERROR,4
BCI 1.0(6)
ERRF TSX ERROR,4
BCI 1.0(5)
ERRV TSX ERROR,4
BCI 1.0(4)
REM THIS IS THE FLOATING PT. TABLE USED IN OBC
DEC 1E+32,1E+16,1E+8,1E+4,1E+2
CONVERSION TABLE
REM\DEC 1E+1
REM REM END OF THE SAP SUBROUTINE NUMBER.
REM EJECT
REM THIS IS SUBROUTINE STORE. IT STORES CHARACTERS
REM AT THE ARRAY VAR.
REM STORE
SKD TEMP=13.1 SAVE INDEX A.
SKD TEMP=14.2 SAVE INDEX B.
LXD J.1 PUT J INTO INDEX REGISTER A.
LOCJA LCD MSBEFT,2 LOAD INDEX B WITH MSHIFT.
TIX LOCJB.2.6 ADVANCE MSHIFT.
ARR J.2 REFRESH MSHIFT.
TII **I+1,1 RAISE J BY ONE IF MSHIFT IS OVER
STI VAR+1 CLEAR NEXT CALL
LOCA CLA WORD LEAVE SIGN BEHIND
LOQ 40.1 MOVE CHARACTER
STQ TEMP-7 PLACES TO THE LEFT.
CAL TEMP-7 PLACES TO THE RIGHT.
ORS VAR+1.1 STORE THE CHARACTER AT VAR.
SKD MSBEFT,2 SAVE MSHIFT.
SKD J.1 SAVE J.
LXD TEMP=13.1 RESTORE INDEX A.
LKD TEMP=14.2 RESTORE INDEX B.
TRA J.4 RETURN TO CALLING PROGRAM.
REM REM END OF THE SAP SUBROUTINE STORE.
REM EJECT
REM THIS IS SUBROUTINE TABLE. IT IS USED TO
REM CONSTRUCT A TABLE OF NAMES TO BE USED ON CARDS
REM AND THEIR MEMORY LOCATIONS RELATIVE TO
REM THE CALLING SEQUENCE.
REM TABLE
SKD TEMP=15.4 SAVE INDEX C.
STZ TEMP
LOCKA TSX CMCThk,4
TSX COMPARE,4
BCI 12,000000
TRA **352.2 JUNK
TRA LOdKD+1 ALPHABETIC
TRA LOCKA NUMERIC
TRA LOCK+1 COMMA
LOCKB STZ TEMP
TRA LOCKD REM COMES HERE TO CONVERT THE ADDRESS TO OLTAL FOR
LOCCK CLA TEMP THE TABLE.
ALS 2 ADD TEMP
ALS 1 LOAD WORD
STD TEMP ADDS TO MAGNITUDE
REM REM COMES HERE TO GET NUMERICS.
LOCKD TSX CMCThk,4
TSX COMPARE,4
BCI 12,000000
TRA **352.2 JUNK
TRA ERRA ALPHABETIC
TRA ANERRA NUMERIC
TRA LOCK CHARACTER
TRA LOCKF SIGN
REM REM COMES HERE IF A DECIMAL PT WAS FOUND.
LOCKE CAL TEMP DECIMAL POINT
CHS TEMP MAKE SIGN MINUS
STD TEMP
TRA LOCKD REM REM COMES HERE IF AN * SIGN WAS FOUND.
LOCKF TSX CLEAR,4
LOCKG TSX CMCThk,4
TOP **2 4 MINUS MEANS FROM NEW CARD
TSX T33T4
TSE LOCKH
TSX COMPARE,4
BCI 1.000000
TRA **352.2 JUNK
TRA ERRG ALPHABETIC OR NUMERIC
SKD B.2 COMMA
SKD B.2 SLASH
TRA LOCKG REM REM COMES HERE TO STORE CHARACTER.
LOCKH ACL +9000000 REPLACE ZERO WITH LETTER D
STD WORD
LOCKJ TSX STURE,4
TRA LOCKG REM REM COMES HERE AT END OF NAME.
REM PSEUDO ACCUMULATOR IACCI FOR EACH OPERATION
REM NUMBERS AND FORM ARITHMETIC RESULTS IN THE
REM FIX FLOATING POINT NUMBERS. FLOAT FIXE0 POINT
REM THE FOLLOWING FOUR SUBROUTINES ARE USE0
REM CHARACTEK OR CARO 0tGlNS
REM A NEW CARD. ACOMMA WILL BE PUT INTO THE CURRENT
REM ALPHABtTIC AN0 AN = SIGN
REM ALL
REM THIS IS SUBROUTINE TEST. IT LOOKS AHEAD TO CLASSIFY
REM A NEW CARD. ACOMMA WILL BE PUT INTO THE CURRENT
REM CHARACTER POSITION ONLY IF EITHER (1) THE NEXT
REM CARD BEGINS WITH A $ SIGN FOLLOWED BY SOME OTHER
REM CHARACTER OR (2) THE NEXT CARD BEGINS WITH AN
REM ALPHABETIC AND AN + SIGN IS FOUND AND IT PRECEDES
REM 2 STORE KEY INTO TABLE
REM KEEP ZONE OF 1ST VAR CH-
REM TRANSFER WHEN DONE
REM REM THE FOLLOWING FOUR SUBROUTINES ARE USED TO
REM REM CONVERT DECIMAL, DIGITS TO BINARY IN VAR.
REM REM FIX FLOATING POINT NUMBERS, FLOAT FIXED POINT
REM REM NUMBERS, AND FORM ARITHMETIC RESULTS IN THE
REM REM PSEUDO ACCUMULATOR IACCI FOR EACH OPERATION
REM REM ON A CARD.
REM REM END

EJECT
ACLU

'ACCUMULATE A SERIES OF BASE 10 Digits in Binary in Var.'

ADD VAR

ALS 1

ACL WOD

STD VAR

TRA 1

REM

FLT

'CONVERT TO FLOATING POINT THE CONTENTS OF THE STORAGE CALLED TEMP.'

ORA +02330000000000

FAO +02330000000000

STD TEMP

TRA 1

REM

FIX

'CONV TO FIXED PT THE CONTENTS OF THE ACCUMULATOR.'

LRS 0

AMA +0377777

LRS 0

ALS 1

TRA 1

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM

LKD

'BRANCH FOR OPERATOR PREPARE FOR NEXT OPERATOR.'

STI OPER

CLA TEMP

TRI +992

TRA ACN

TRA

REM
"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