FUNCTIONAL SPECIFICATIONS OF
NRTPOD PROGRAM MODIFICATIONS

10 NOVEMBER 1966
Revised February 1967

Prepared for
Massachusetts Institute of Technology
Lincoln Laboratory

Under Contract No. CC 939
Distribution of this document is unlimited.
FUNCTIONAL SPECIFICATIONS OF NRTPOD PROGRAM MODIFICATIONS

10 NOVEMBER 1966
Revised February 1967

Prepared for
Massachusetts Institute of Technology
Lincoln Laboratory

Under Contract No. CC 939
PRIME CONTRACT AF 19(628)-5167
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>2.</td>
<td>GENERAL DESCRIPTION AND OPERATING INSTRUCTIONS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Parameterization of Drag</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Functional Standard Deviation</td>
<td>2-2</td>
</tr>
<tr>
<td>2.3</td>
<td>Diagonalized Covariance Matrix</td>
<td>2-6</td>
</tr>
<tr>
<td>2.4</td>
<td>R Observable</td>
<td>2-8</td>
</tr>
<tr>
<td>2.5</td>
<td>Steering Ephemeris</td>
<td>2-9</td>
</tr>
<tr>
<td>2.6</td>
<td>Linear Constraints</td>
<td>2-10</td>
</tr>
<tr>
<td>3.</td>
<td>INPUT</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>Deck Set-up</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>JDC Card</td>
<td>3-1</td>
</tr>
<tr>
<td>3.3</td>
<td>Preliminary Data Input</td>
<td>3-3</td>
</tr>
<tr>
<td>3.4</td>
<td>Sensor Cards</td>
<td>3-7</td>
</tr>
<tr>
<td>3.5</td>
<td>Observation Cards</td>
<td>3-10</td>
</tr>
<tr>
<td>3.6</td>
<td>Ephemeris Cards</td>
<td>3-10</td>
</tr>
<tr>
<td>4.</td>
<td>OUTPUT</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>JDC Print</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>Input Listing</td>
<td>4-1</td>
</tr>
<tr>
<td>4.3</td>
<td>Run Header Data</td>
<td>4-3</td>
</tr>
<tr>
<td>4.4</td>
<td>Sensor Information</td>
<td>4-3</td>
</tr>
<tr>
<td>4.5</td>
<td>Observations</td>
<td>4-3</td>
</tr>
<tr>
<td>4.6</td>
<td>Residuals</td>
<td>4-3</td>
</tr>
<tr>
<td>4.7</td>
<td>Functional Standard Deviations</td>
<td>4-7</td>
</tr>
<tr>
<td>4.8</td>
<td>Mean and Standard Deviations by Sensor</td>
<td>4-7</td>
</tr>
<tr>
<td>4.9</td>
<td>Iteration Summary</td>
<td>4-10</td>
</tr>
<tr>
<td>4.10</td>
<td>Trajectory and Steering</td>
<td>4-10</td>
</tr>
<tr>
<td>4.11</td>
<td>Matrix Update</td>
<td>4-15</td>
</tr>
<tr>
<td>5.</td>
<td>PROGRAM FUNCTIONAL DESCRIPTION</td>
<td>5-1</td>
</tr>
<tr>
<td>5.1</td>
<td>NRTPD2 Subroutine Overlay</td>
<td>5-1</td>
</tr>
<tr>
<td>5.2</td>
<td>Subroutine Glossary</td>
<td>5-3</td>
</tr>
<tr>
<td>5.3</td>
<td>Brief Subroutine Descriptions</td>
<td>5-10</td>
</tr>
<tr>
<td>5.4</td>
<td>Subroutine Descriptions</td>
<td>5-13</td>
</tr>
</tbody>
</table>
## CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>NRTPD2 STORAGE MAP</td>
<td>6-1</td>
</tr>
<tr>
<td>6.1</td>
<td>New/Revised COMMON Arrays</td>
<td>6-1</td>
</tr>
<tr>
<td>6.2</td>
<td>New Cells Added to Data Storage</td>
<td>6-5</td>
</tr>
<tr>
<td>6.3</td>
<td>Blank COMMON</td>
<td>6-6</td>
</tr>
<tr>
<td>7.</td>
<td>PREPOD PROGRAM</td>
<td>7-1</td>
</tr>
<tr>
<td>7.1</td>
<td>Input</td>
<td>7-3</td>
</tr>
<tr>
<td>7.2</td>
<td>Output</td>
<td>7-6</td>
</tr>
<tr>
<td>7.3</td>
<td>PREPOD Storage Map</td>
<td>7-13</td>
</tr>
<tr>
<td>7.4</td>
<td>PREPOD Subroutine Descriptions</td>
<td>7-14</td>
</tr>
<tr>
<td>8.</td>
<td>REFERENCES</td>
<td>8-1</td>
</tr>
</tbody>
</table>
# ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Representative Altitude Dependent Drag Model</td>
<td>2-1</td>
</tr>
<tr>
<td>2-2</td>
<td>Representative Radar Cross-Section Profile as a Function of the Aspect Angle</td>
<td>2-3</td>
</tr>
<tr>
<td>2-3</td>
<td>Sequential Rotations of the UVW Coordinate System to Align it with the Principal Axes</td>
<td>2-7</td>
</tr>
<tr>
<td>2-4</td>
<td>Constraint Matrix</td>
<td>2-12</td>
</tr>
<tr>
<td>4-1</td>
<td>Sample JDC Print</td>
<td>4-2</td>
</tr>
<tr>
<td>4-2</td>
<td>Sample Run Header</td>
<td>4-4</td>
</tr>
<tr>
<td>4-3</td>
<td>Sample Functional Sigma Input Page</td>
<td>4-5</td>
</tr>
<tr>
<td>4-4</td>
<td>Sample Observations Print</td>
<td>4-6</td>
</tr>
<tr>
<td>4-5</td>
<td>Sample Residuals Print</td>
<td>4-8</td>
</tr>
<tr>
<td>4-6</td>
<td>Sample Functional Standard Deviations Print</td>
<td>4-9</td>
</tr>
<tr>
<td>4-7</td>
<td>Sample Iteration Summary Print</td>
<td>4-11</td>
</tr>
<tr>
<td>4-8</td>
<td>Sample Trajectory Print</td>
<td>4-13</td>
</tr>
<tr>
<td>4-9</td>
<td>Sample Trajectory Plus Steering Ephemeris Output</td>
<td>4-14</td>
</tr>
<tr>
<td>4-10</td>
<td>Sample Printout of a Single Trajectory-Steering-Matrix Update Output Block</td>
<td>4-16</td>
</tr>
<tr>
<td>5-1</td>
<td>NRTPD2 Subroutine Overlay Structure</td>
<td>5-2</td>
</tr>
<tr>
<td>5-2</td>
<td>BODY Flow Diagram</td>
<td>5-26</td>
</tr>
<tr>
<td>5-3</td>
<td>DAUX Flow Diagram</td>
<td>5-40</td>
</tr>
<tr>
<td>5-4</td>
<td>DPROS Flow Diagram</td>
<td>5-48</td>
</tr>
<tr>
<td>5-5</td>
<td>Iteration Techniques Used in Subroutine FALSI</td>
<td>5-62</td>
</tr>
<tr>
<td>5-6</td>
<td>FALSI Flow Diagram</td>
<td>5-63</td>
</tr>
<tr>
<td>Figures</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>5-7</td>
<td>FIT Flow Diagram</td>
<td>5-68</td>
</tr>
<tr>
<td>5-8</td>
<td>HINT Flow Diagram</td>
<td>5-82</td>
</tr>
<tr>
<td>5-9</td>
<td>INPUT Flow Diagram</td>
<td>5-86</td>
</tr>
<tr>
<td>5-10</td>
<td>OBSIN Flow Diagram</td>
<td>5-112</td>
</tr>
<tr>
<td>5-11</td>
<td>PLAMDA Flow Diagram</td>
<td>5-119</td>
</tr>
<tr>
<td>5-12</td>
<td>RADR Flow Diagram</td>
<td>5-154</td>
</tr>
<tr>
<td>5-13</td>
<td>TRAJ Flow Diagram</td>
<td>5-182</td>
</tr>
<tr>
<td>5-14</td>
<td>TRJTAP Flow Diagram</td>
<td>5-202</td>
</tr>
<tr>
<td>7-1</td>
<td>PREPOD Flow Diagram</td>
<td>7-2</td>
</tr>
<tr>
<td>7-2</td>
<td>PREPOD Input Deck</td>
<td>7-4</td>
</tr>
<tr>
<td>7-3</td>
<td>PREPOD Sample Input Sheet for Two Stacked Cases</td>
<td>7-8</td>
</tr>
<tr>
<td>7-4</td>
<td>Sample NAMELIST Input Listing</td>
<td>7-9</td>
</tr>
<tr>
<td>7-5</td>
<td>Sample Output of Program Constants, Sensor Location and Biases, and Observations</td>
<td>7-10</td>
</tr>
<tr>
<td>7-6</td>
<td>Sample Initial Conditions Page</td>
<td>7-11</td>
</tr>
<tr>
<td>7-7</td>
<td>Listing of Punched Card Output</td>
<td>7-12</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This document describes the six modifications which have been added to the NRTPOD program in the form of extended capabilities. In addition, a separate stand-alone program which derives a preliminary estimate of an orbit and which is designed to be used in conjunction with NRTPOD is described in this document.

For purposes of distinction within this document the modified NRTPOD program is designated NRTPD2. However, it should be emphasized that this term is used to differentiate between different versions of the same program only; the subroutine descriptions still refer to the main program as NRTPOD.

This report is intended to be an analyst's guide to the NRTPOD modifications as well as an operational handbook with input/output instructions. An entire section of this report is devoted to an analytical and operational discussion of the six modifications which have been added to the NRTPOD program. The mathematical derivations of the various program options are not given completely in this section, although pertinent references (in which complete derivations are presented) are given in each respective section. The program descriptions given in this document are intended to be supplemental to the mathematical and operational descriptions of NRTPOD as described in Reference 1, Functional Specifications of Lincoln Laboratory Orbit Determination Programs. Since NRTPOD is a derivative of the ESPOD Program, considerable reference is made to the original ESPOD documentation, Reference 2.

In addition to the functional specifications of the NRTPOD modifications and the input/output instructions, a section of new and modified subroutines has been included. The standard description and accompanying flow diagrams of the modified subroutines have been abbreviated or eliminated in many instances in order to keep the volume of documentation to a manageable level. In these instances, reference to the main source of subroutine documentation is given.
PREPOD, a preliminary orbit determination program which derives an initial estimate of an orbit, is described in the last section of this report. Since this program is distinct from NRTPOD, the complete input/output and subroutine descriptions are given in their entirety in one section. The orbit determination method depends on the availability of two or more position fixes from an observing station over some free-flight arc. The preliminary conditions are derived by fitting an orthogonal polynomial to the components (topocentric range, azimuth, and elevation) of the position fixes.

The six modifications which have been added to NRTPOD are described briefly below.

**Parameterization of Drag**

This modification enhances the simulation of drag forces and, in particular, the capability of reducing data from reentering vehicles. The ballistic parameter $C_D A/M$ is represented by an altitude dependent, tabular function which is linearly interpolated in a given altitude range. This modification allows the analyst to simulate and regress on as many as 15 functional (altitude) values of the ballistic parameter.

**Functional Standard Deviation**

A trajectory functional standard deviation has been incorporated into NRTPOD to allow the assignment of data weights as a function of topocentric range, the radar cross section of the vehicle, and the particular sensor-vehicle geometry. The functional standard deviation is added to the nominal standard deviation and is usually negligible for nominal tracking distances.

**Diagonalized Covariance Matrix Output**

Whenever the normal matrix update option is exercised, diagonalized covariance matrix information is printed automatically. The printed output consists of the square roots of the eigenvalues and associated eigenvectors of the position and
velocity partitions of the orbit plane (UVW) covariance matrix. In addition, the three sequential rotation angles to align the UVW axes with the principal axes of the error ellipsoid are output. Finally, the determinants of specific covariance matrices are printed.

\( \ddot{R} \) Observable

The NRTPOD program has been modified to accept and process \( \ddot{R} \) data in addition to its conventional observables. This modification requires no special operational procedures, since the new observable is processed in the same manner as the observables which were in the program originally.

Steering Ephemeris

A radar steering option has been incorporated into the trajectory link of NRTPOD. By calling this option, the computed observables of every sensor in the sensor input list based on the current trajectory are printed at specified intervals for the length of the trajectory propagation. In addition to printing the steering ephemeris of each sensor, the current altitude, ballistic coefficient, and atmospheric density at the specified intervals are printed.

Linear Constraints

The capability of imposing linear constraints on the solution variables has been added to NRTPOD. This modification permits the analyst to require that any one of the solution variables to be a linear combination of any of the others in accordance with the requirements of the physical problem. An example of a physical constraint that should be accounted for in the tracking problem would be the precise knowledge of the relative locations of two observing radar stations. In this case, the linear constraints formulation would force the relative station geometry to remain fixed throughout the differential correction process.
The above modifications were coded in the FORTRAN IV programming language. Checkout and final integration of the modifications into the NRTPOD program were done on the IBM 7094 computer at TRW Systems facilities. A double precision version of NRTPD2 has been installed on the IBM 360/67 computer using the level H FORTRAN compiler. In fact, the program can be installed with only minor system interface modifications on any computer which accepts the ASA Standard FORTRAN IV language.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office
2. GENERAL DESCRIPTION AND OPERATING INSTRUCTIONS

This section is concerned with a functional and operational description of the NRTPOD modifications. It is assumed that the reader is already familiar with the basic operation of the NRTPOD program as described in Reference 1.

2.1 PARAMETERIZATION OF DRAG

The NRTPD2 drag model is represented by an altitude dependent, linearly interpolated, tabular function. Up to fifteen altitude bands, ranging from sea level up to altitudes where the mean free path of atmospheric molecules is larger than the maximum spacecraft dimension, can be simulated in the trajectory model. The ballistic parameter, \( C_{DA}/M = C(h) \), is linearly interpolated in a given altitude range, and the differential correction regresses on the functional (altitude) values.

![Figure 2-1. Representative Altitude Dependent Drag Model](image)

The altitude is defined above the ellipsoid and used with the Lockheed-Jacchia atmosphere model. The required partial derivatives of the observations with respect to the ballistic solution parameters are obtained by simultaneously integrating the variational equations associated with the ballistic parameters (and hence, altitudes) bounding the vehicle at a particular time. Whenever an altitude range is crossed, the variational equations are reinitialized and the numerical integration is restarted. The detailed mathematical techniques associated with regression using the altitude dependent ballistic coefficients are treated in greater detail in Reference 3.
The linear constraints option discussed in Section 2.6 is a useful research tool to employ in conjunction with the variable drag model. It provides the analyst with flexibilities which the usual constraining techniques—that is, bounds—do not provide.

In an actual orbit determination using real data, the variable drag model should not be used initially. Since the drag variation has a relatively small effect on the trajectory, unless the vehicle is reentering the atmosphere, and considering that the larger solution vector enhances the computer running time appreciably, it is more efficient to regress for the position and velocity and possibly constant drag on the first determination with the real data. After bad observations have been rejected and a fairly good determination of the state vector has been established, the variable drag solution with the improved estimate of the state vector is attempted. This method is usually more successful than attempting to solve for the state vector and a variable drag model without a reasonably good estimate of the position and velocity of the vehicle.

The normal matrix generated during the differential correction can be updated to an arbitrary time by flagging the trajectory and update link following a curve fit. The full normal matrix including the variable drag terms can be updated. The input requirements for the drag options and other operational considerations are treated in Section 3.3.2.

2.2 FUNCTIONAL STANDARD DEVIATION

A functional form of the standard deviation has been incorporated into NRTPD2. The standard deviation is a function of the trajectory relative to the sensor and the topocentric range. The functional form reduces essentially to the nominal standard deviation for the tracking of satellites in the normal operating range. However, as the topocentric range increases beyond the normal tracking distances, the range dependency results in a substantially higher standard deviation.

2.2.1 Mathematical Formulation

The functional form of the functional standard deviation is as follows

$$\sigma_f = \sqrt{\frac{A_1^2}{B_i R^4} + f(\theta)} \quad (1)$$
where

\[ A_i = \text{nominal standard deviation} \]
\[ B_i = \text{sensor dependent constant} (i = 1 \ldots 5) \text{ denoting} \]
\[ A, E, R, \dot{R}, \ddot{R} \]
\[ R = \text{topocentric range} \]
\[ f(\theta) = \text{radar cross section} \]
\[ \theta = \text{aspect angle} \]

The quantity \( f(\theta) \) is the radar cross section as a function of the angle away from the nose-on direction of an axially symmetric vehicle. The argument \( \theta \) is the angle between the drag velocity vector and the topocentric range vector. This angle is trajectory dependent and its computation is derived in Reference 4. The function \( f(\theta) \) is obtained by linear interpolation from the input table \( [\theta_j, f(\theta)_j] \) \( j = 1 \ldots n \), where \( n \leq 7 \); each table is sensor specific.

Graphically, the radar cross section function might appear as shown below, Figure 2-2.

![Figure 2-2. Representative Radar Cross-Section Profile as a Function of Aspect Angle](2-3)
Given the argument $\theta$ (which is computed from the position and velocity of the spacecraft and the coordinates of the sensor) where $\theta_j < \theta \leq \theta_{j+1}$ the corresponding value of $f(\theta)$ is obtained by linearly interpolating between $[\theta_j, f(\theta)_j]$ and $[\theta_{j+1}, f(\theta)_{j+1}]$. The function is assumed constant between $0$ and $\theta_1$ and between $\theta_n$ and $\pi$. Mathematically

$$f(\theta) = \begin{cases} f(\theta_1), & \text{if } 0 \leq \theta \leq \theta_1 \\ f(\theta_n), & \text{if } 0 \leq \theta \leq \pi \end{cases}$$

If there is either one entry or none in the $[\theta, f(\theta)]$ table, the following conditions apply:

$$f(\theta) = \begin{cases} f(\theta_1), & \text{if } n = 1 \quad (0 \leq \theta \leq \pi) \\ 1, & \text{if } n = 0 \quad (0 \leq \theta \leq \pi) \end{cases}$$

2.2.2 Operational Specifications

The $[\theta, f(\theta)]$ table and the sensor constants $B_i$ are input on sensor cards. See Section 3.4 for a detailed description of the sensor cards. The functional standard deviation option is flagged by the presence of a Type 4 sensor card ($B_i$ constants) in the input deck. Even though the $[\theta, f(\theta)]$ table (Types 5 and 6 sensor cards) is not input, the functional standard deviation option is still operational under the assumption that $f(\theta) = 1$; see the previous section.

The sensor constants $B_i$ are usually computed by the analyst given the tracking standard deviations as a function of topocentric range for a particular sensor. By rewriting the defining equation of the functional standard deviation as equation (2), a convenient method of calculating the sensor and observable dependent constants, $B_i$, can be illustrated.

$$B_i = (\sigma_f^2 - A_i^2) f(\theta)/R^4 \quad (2)$$

Given the nominal standard deviation of a sensor, the radar constant can be evaluated if the functional standard deviation is specified for a satellite of known radar cross section and range. The radar constant is in mixed units as the individual terms in equation (2) are in set units. The standard deviations (both functional and nominal) are specified in card input units, as shown in Table 2-1 below.
Table 2-1. Units of Standard Deviations of Observables Used to Evaluate Sensor Constants

<table>
<thead>
<tr>
<th>Observable</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azimuth</td>
<td>Degrees</td>
</tr>
<tr>
<td>Elevation</td>
<td>Degrees</td>
</tr>
<tr>
<td>Range</td>
<td>Kilometers</td>
</tr>
<tr>
<td>Range-rate</td>
<td>Kilometers/second</td>
</tr>
<tr>
<td>Range acceleration</td>
<td>Meters/second/second</td>
</tr>
</tbody>
</table>

The radar cross section $f(\theta)$, is in (meters)$^2$, and the specific range for which the constant is being evaluated is in earth-radii. For example, if the nominal standard deviation in range of a sensor is 40 meters, and the functional standard deviation is 50 meters for a satellite of 2 m$^2$ in radar cross section at 4,000 km range, the computation of $B_R$ would be carried out as shown below:

$$B_R = \left( \sigma_f^2 - A_i^2 \right) f(\theta)/R^4$$

$$= \left[ (0.05)^2 - (0.04)^2 \right] (2.0)/(4000.0/6378.0)^4$$

$$B_R = 11.6 \times 10^{-3}$$

The $A_i$ constant in the defining equation of the functional standard deviation is equal to the nominal standard deviation, which is the standard deviation the program uses if the functional form is not called. If the standard deviation is entered on both the observation card ($\sigma_o$) and the sensor card ($\sigma_s$), the observation card value is used.

Mathematically,

$$A_i = \begin{cases} 
\sigma_s, & \text{if } \sigma_o = 0 \\
\sigma_o, & \text{if } \sigma_o > 0 
\end{cases}$$

Since the aspect angle, and hence the functional standard deviation, is trajectory dependent, the weight applied to a particular residual varies
from iteration to iteration. The standard deviations of each observation are printed as a companion page to the residuals print. See the sample output in Section 4.6. The functional standard deviations print is optional and is called by setting Column 43 of the JDC equal to one.

2.3 DIAGONALIZED COVARIANCE MATRIX OUTPUT

Diagonalized covariance matrices and associated quantities are output at each update (DELTT) time whenever a trajectory and matrix update are performed. No flags are required to obtain this output as it is automatically computed during the process of a matrix update.

The eigenvalues and associated eigenvectors of the upper 3 x 3 (position) and lower 3 x 3 (velocity) partitions of the UVW covariance matrix are computed at each update time. The UVW system is a vehicle-centered coordinate system; see Section 6.4 of Reference 1. The UVW covariance matrix is not an output quantity of NRTPD2, although it is internally computed from the Cartesian covariance matrix. The square roots of the eigenvalues (and the associated eigenvectors) are output; see Section 4.11 for a sample printout.

The orientation of the position and velocity error ellipsoids with respect to the U, V, W axes is such that the principal axes are identified by the nearest axes of the U, V, W set. From Figure 2-3 and on the assumption that the spin axis of the vehicle is coincident with the V axis (downrange direction), the sense of positive rotations can be derived. In this configuration, a rotation about the U axis results in a yaw of the vehicle; a positive rotation about U is defined as V x W, hence yaw positive is turning to the left. A rotation about the V axis is a roll maneuver; a positive rotation is W x U or clockwise. A rotation about the W axis results in a pitch of the vehicle (V) axis; positive pitch is defined as U x V, or down. The ordered rotations to align the UVW coordinates with the principal axes of the error ellipsoid for the NRTPD2 program will be sequential rotations about (1) the U axis, (2) the V axis, and (3) the W axis; i.e., yaw-roll-pitch. The ordered rotations and associated definitions are summarized in Table 2-2.
Table 2-2. Summary of UVW Coordinate System Rotation Conventions

<table>
<thead>
<tr>
<th>Spin Axis</th>
<th>Positive Rotation Definition</th>
<th>Alignment Nomenclature</th>
<th>Positive Direction</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 U</td>
<td>V x W</td>
<td>yaw</td>
<td>left</td>
<td>0 1</td>
</tr>
<tr>
<td>2 V</td>
<td>W x U</td>
<td>roll</td>
<td>clockwise</td>
<td>0 2</td>
</tr>
<tr>
<td>3 W</td>
<td>U x V</td>
<td>pitch</td>
<td>down</td>
<td>0 3</td>
</tr>
</tbody>
</table>

The ordered yaw-roll-pitch rotations of the U, V, W coordinate axes which will align them with the error ellipsoid are illustrated in Figure 2-3.

![Figure 2-3](image)

Figure 2-3. Sequential Rotations of the UVW Coordinate System to Align it with the Principal Axes.

In this illustration, according to the definitions in Table 2-2, yaw (about U) and roll (about V) are positive rotations and pitch (about W) is a negative rotation. For both the position and velocity partitions of the UVW
covariance matrix the ordered yaw-roll-pitch rotations of the U, V, W axes which will align them with the principal axes of the error ellipsoid are output.

In addition to the diagonalized covariance matrix output, the determinants of the following (sub) matrices are computed and output:

1) Square root of the determinant of the upper (position) 3 x 3 partition of the Cartesian covariance matrix (km$^3$)

2) Square root of the determinant of the lower (velocity) 3 x 3 partition of the Cartesian covariance matrix (km$^3$/sec$^3$)

3) Square root of the determinant of the 6 x 6 Cartesian covariance matrix (km$^6$/sec$^6$)

The Cartesian covariance matrix itself is not output; it is computed internally from the polar spherical (ADBARV) covariance matrix. The determinants are computed by a matrix decomposition method. The LEGS2 subroutine uses such a method in the process of solving the normal equations; hence, the subroutine was used to evaluate the determinants. See Reference 5 for a mathematical discussion of the evaluation of determinants as accomplished in this program. The output units of the square roots of the determinants are indicated above. See Section 4.11 for a sample output of the quantities described in this section.

2.4 $\ddot{R}$ DATA

Range acceleration has been added to the list of observables that are acceptable to NRTPD2. There are no specific instructions as to its use other than the input and output descriptions. (See Sections 3 and 4.) The only exception involves the solution of observation time bias. Since the partial derivative of range acceleration with respect to time is not included in the $\ddot{R}$ partials, it is not possible to regress for time bias when using $\ddot{R}$ data only.

The equations required to implement this modification may be found in the equation section of the pertinent subroutines. The modified subroutines are: DRDP, PRELIM, PUPB, RADR (Section 5). The mathematical formulation of the required $\ddot{R}$ partial derivatives and associated computations are developed in Reference 6.
2.5 STEERING EPHEMERIS

The radar steering package is a trajectory or post-differential correction option of NRTPD2. When the option is called, the following trajectory parameters are printed at each specified (DELT) time:

1) \( h = \) Height (km) of the vehicle above the ellipsoid, as used for entry in the Lockheed-Jacchia atmosphere model.

2) \( C(h) = C_D A/m \) (m\(^2\)/kg), the ballistic coefficient.

3) \( \rho(h) = \) Atmospheric density (kg/m\(^3\)).

In addition, the following radar (topocentric) parameters are printed for each sensor in the Master Sensor Table:

4) \( A = \) Azimuth of the vehicle (degrees)
5) \( E = \) Elevation of the vehicle (degrees)
6) \( R = \) Range to the vehicle (km)
7) \( \dot{R} = \) Range rate of the vehicle (km/min) or (km/sec)
8) \( \ddot{R} = \) Range acceleration of the vehicle (km/min\(^2\)) or (m/sec\(^2\))

The three trajectory only dependent parameters \( h, C(h), \) and \( \rho(h) \) are computed in the atmosphere related subroutines, that is, DRAG, JACHIA, and ATM59. The sensor dependent parameters are computed in subroutine STEER. The equations for the radar steering ephemeris are developed in Reference 7.

The radar steering option is called by setting Column 54 of the JDC card. The steering information for a particular time is printed following the trajectory block. If it is desired, the trajectory print can be suppressed. The tabulation below summarizes the steering options.

<table>
<thead>
<tr>
<th>JDC Column</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>0</td>
<td>No steering</td>
</tr>
<tr>
<td>54</td>
<td>1</td>
<td>Trajectory print and steering</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Steering only</td>
</tr>
</tbody>
</table>
When a differential correction precedes a trajectory propagation and steering run, the radar ephemeris is based on the converged or best estimate of the trajectory at the termination of the differential correction. If the ballistic coefficient is included in the solution vector, it is transferred to the trajectory link also for trajectory and steering computations. If drag is not included in the solution vector or if there is no differential correction, initial estimates of the ballistic coefficient may be input on preliminary data cards. See Section 3.3.2 for a description of the preliminary data input cards which specify the drag model.

The steering ephemeris is constrained to print only when the vehicle is above a nominal horizon of $-5^\circ$. Since there is no search for rise and set times, the steering output begins (ends) at the first (last) point above the horizon. The nominal horizon ($-5^\circ$) may be changed with a preliminary input card—ECRIT (Section 3.3.3). The output units of some of the steering ephemeris parameters can be varied (Section 3.3.3).

2.6 LINEAR CONSTRAINTS

The linear constraints option provides the analyst with additional control over the solution vector. This method provides the capability of specifying the correction of a particular variable in terms of a linear combination of another variable, whereas the bounds technique only permits the analyst to specify the maximum correction to a solution variable on a given iteration.

2.6.1 Applications

Constraints among the solution variables are often a part of the physical problem. An example of a physical constraint with application to the tracking problem would be the precise knowledge of the relative locations of two radar stations. If the actual locations are among the solution variables in a differential correction, it is important to constrain the corrections so that the relative locations are preserved. If a nonlinear constraint is required, it is still possible to apply it to the solution variables, although the formulation will only be valid for one iteration. Keeping the orbital period constant is an example of a nonlinear constraint. In actual practice, linear constraints are widely used in the solution of sensor biases, especially if an orbit determination involves a particular tracking
net with similar tracking equipment. When certain parameters of a physical system which are usually uncorrelated are related, it is advantageous to account for it in the solution. For example, if pass-by-pass bias solutions are desired for a single radar station, a different station identification can be assigned to each pass and then, by linear constraints, require that the station location biases for each identification be equal.

2.6.2 Constraint Matrix

The constrained solution is implemented by introducing linear constraints of the form

$$x = By$$

where $$x_{n 	imes 1}$$ is the original solution variable, $$B$$ is the constraint matrix, and $$y_{m 	imes 1}$$ is the reduced set of solution variables. Therefore, the constraint matrix $$B$$ is a rectangular matrix $$(n \times m)$$, $$m \leq n$$ whose elements describe the linear relation between the solution variables. The mathematical formulation for the implementation of linear constraints into NRTPD2 is developed in Reference 8.

2.6.3 Examples

As a first example, suppose a differential correction was being made for the initial conditions of the spacecraft, the station locations of stations AA and BB of known angular separation, and the angle biases of station BB. By constraining the changes to the station locations to be the same, the known angular separation will be preserved. The constraint matrix $$B$$ would look like Figure 2-4., where the rows are the original solution variables and the columns, the constrained variables. The preliminary input which specifies this constrained solution consists of the two following cards: (See Section 3.3.4 for linear constraints input.)

$$BIJ = 101, 202, 303, 404, 505, 606, 707, 808, 909, 1010, 1107, 1208$$
$$XIJ = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$

As was mentioned previously, linear constraints can be used to keep the energy of an orbit constant during a differential correction. A second example follows to familiarize the reader with the process of setting up a linear constraints case. From the energy equation

$$v^2 = \mu \left(\frac{2}{r} - \frac{1}{a}\right)$$

2-11
### Constraint Matrix

<table>
<thead>
<tr>
<th></th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(A)</th>
<th>(r)</th>
<th>(V)</th>
<th>(AA)</th>
<th>(BB)</th>
<th>(AA)</th>
<th>(BB)</th>
<th>(BB)</th>
<th>(BB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta)</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A)</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(r)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(V)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(AA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(BB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(AA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(BB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(BB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2-4. Constraint Matrix**
it is easily derived that, from the condition that the energy remain con-
stant, the permissible change in velocity in terms of position and velocity
is given by equation (5)

\[ \Delta v = -\Delta r \left( \frac{\mu}{r^2 v} \right) \]  

(5)

where \( \mu \) is the gravitational constant.

Given

\[ \mu = 5.5303934 \times 10^{-3} \text{ e.r.}^3/\text{min}^2 \]

\[ r = 9567.2475 \text{ km} = 1.50 \text{ earth radii (e.r.)} \]

\[ v = 8.50422 \text{ km/sec} = 0.08 \text{ e.r./min} \]

Substituting into equation (5), one finds

\[ \Delta v = -\Delta r \frac{\mu}{(1.5)^2 (0.08)} \]

\[ \Delta v = -\Delta r (0.0307) \]  

(5a)

Hence, for this second example, the preliminary data input which controls
the linear constraints option looks like the two input cards illustrated
below

\[ \begin{align*}
\text{BIJ} &= 101, 202, 303, 404, 505, 605 \\
\text{XIJ} &= 1, 1, 1, 1, 1, -0.0307,
\end{align*} \]

It should be noted that the energy constraint (that it should remain
constant) is valid for the first iteration only since the value of the constant
(\( \mu/r^2v \)) changes with succeeding iterations. However, this is only true
when the proportionality constant is a function of the solution variables,
such as this constant energy constraint. Also, the proportionality constant
is computed and input in internal (program) units; that is, earth radii and
minutes.
2.6.4 Output

The printed output of a linear constraints case is nearly identical to
the standard output. The normal matrix is accumulated in the constrained
system and hence is output similarly; that is, the matrix is an \( m \times m \),
where \( m \) is the number of solution variables in the constrained system.
The variance-covariance matrix and the correlation matrix are output in
terms of the original solution variables; that is, the dimension is \( n \times n \).
3. PROGRAM INPUT

New inputs have been provided for control of the options that have been added to NRTPOD. The program options and the respective input requirements are listed below.

<table>
<thead>
<tr>
<th>Program Option</th>
<th>Input Requirements</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameterization of Drag</td>
<td>Preliminary data cards</td>
<td>3.3.2</td>
</tr>
<tr>
<td>Functional Standard</td>
<td>JDC card</td>
<td>3.2</td>
</tr>
<tr>
<td>Deviation</td>
<td>Sensor cards</td>
<td>3.4</td>
</tr>
<tr>
<td>Diagonalized Matrix Output</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>( \vec{R} ) Observable</td>
<td>Observation cards</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>Sensor cards</td>
<td>3.4</td>
</tr>
<tr>
<td>Steering Ephemeris</td>
<td>JDC card</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>Preliminary data cards</td>
<td>3.3.3</td>
</tr>
<tr>
<td></td>
<td>Sensor cards</td>
<td>3.4</td>
</tr>
<tr>
<td>Linear Constraints</td>
<td>Preliminary data cards</td>
<td>3.3.4</td>
</tr>
<tr>
<td>Mean Elements Input</td>
<td>Preliminary data cards</td>
<td>3.3.1</td>
</tr>
<tr>
<td></td>
<td>(Six-card element set)</td>
<td></td>
</tr>
</tbody>
</table>

3.1 DECK SET-UP

The input deck sequence of NRTPD2 is identical to the input sequence of NRTPOD, as described in Section 2.2.1 of Reference 1.

3.2 JDC—JOB DESCRIPTION CARD

The JDC card is the control card for the flow of information through NRTPD2. This card is always the first card of an input data deck. It selects certain program options and defines the program sections to be used. A short arbitrary remark is permitted on the card.

<table>
<thead>
<tr>
<th>Column</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>JDC</td>
<td>Identifies JDC card</td>
</tr>
<tr>
<td>4-7</td>
<td></td>
<td>Vehicle number</td>
</tr>
<tr>
<td>8-17</td>
<td></td>
<td>Vehicle name</td>
</tr>
<tr>
<td>18-29</td>
<td></td>
<td>User's header</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Not used at present</td>
</tr>
</tbody>
</table>

3-1
<table>
<thead>
<tr>
<th>Column</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>0 or blank</td>
<td>Sensor and observation data not to be processed</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Sensor and observation data to be processed</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Sensors only</td>
</tr>
<tr>
<td>32</td>
<td>0 or blank</td>
<td>Do not print sensor data</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Print sensor data</td>
</tr>
<tr>
<td>33</td>
<td>0 or blank</td>
<td>Do not print observations</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Print observations</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>Do not print functional standard deviation input</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Print functional standard deviation input</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>Observations not presorted, fewer than 345 cards.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Observations presorted, no maximum</td>
</tr>
<tr>
<td>36-40</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td></td>
<td>at present</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0 or blank</td>
<td>Curve fit not desired</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Curve fit desired</td>
</tr>
<tr>
<td>42</td>
<td>0 or blank</td>
<td>A priori S matrix not input on this run</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>A priori S matrix is input on this run</td>
</tr>
<tr>
<td>43</td>
<td>0 or blank</td>
<td>Do not print functional standard deviations</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Print functional standard deviations</td>
</tr>
<tr>
<td>44-50</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td></td>
<td>at present</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>0 or blank</td>
<td>Trajectory print not desired on this run</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Trajectory print is desired on this run</td>
</tr>
<tr>
<td>52</td>
<td>0 or blank</td>
<td>A priori UPMAT (covariance) matrix not input</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>A priori UPMAT matrix is input on this run</td>
</tr>
<tr>
<td>53</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td></td>
<td>at present</td>
<td></td>
</tr>
</tbody>
</table>
The NRTPD2 Program accepts Kozai mean elements for the specification of position and velocity of a satellite at a given time. The elements are input Type 4. The elements need not be referenced to the epoch of the run in question since the capability to update the input mean elements to an arbitrary epoch has been provided.

Mean elements in the revised SPADATS/SPACETRACK format are input. This is a six-card element set, which is described in detail in Section 1.2.4 of Reference 1. When mean elements are input (TYPE = 4), an ITIME card is not required as it is for the other types. The TNULL card is used to specify the epoch in conjunction with the mean elements cards.

The TNULL card specifies the time to which the mean elements are to be updated relative to the epoch given on the six-card element set. The updated epoch is the epoch of a given run, such as the time associated with the initial estimate of position and velocity of a satellite in a differential correction.

The epoch is given in Julian days on Card 2 of the six-card set. TNULL is a three entry array as shown below:

\[
\begin{align*}
\text{TNULL} &= \text{DAYS.}, \ \text{HOURS.}, \ \text{MINUTES.}, \\
\text{TNULL}(2) &= \text{HOURS.}, \ \text{MINUTES.}, \\
\text{TNULL}(3) &= \text{MINUTES.},
\end{align*}
\]
3.3.2 Parameterization of Drag

Four input variables are required to exercise the variable drag options of NRTPD2.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTS</td>
<td>The ALTS card specifies the altitudes (in kilometers) of the ballistic coefficients as given on the CLAMDA card. The trajectory of the vehicle must be enveloped by the top and bottom drag layers; if the vehicle goes outside these regions, the program will exit with an error message.</td>
</tr>
<tr>
<td>CLAMDA</td>
<td>This card specifies the ballistic coefficients, $C_{DA}/m$ (m$^2$/kg), in a one-to-one correspondence with the altitudes as specified on the ALTS card.</td>
</tr>
<tr>
<td>CATLM</td>
<td>The CATLM array indicates to the program the ballistic coefficients which are to be solved for. The first entry relates to the uppermost layer in the atmosphere. As in the CAT1 and CAT2 cards, a &quot;1&quot; indicates the variable is to be solved for and a &quot;0&quot; indicates the variable is not to be solved for. The order of the solution flags corresponds to the ordered entries of the ALTS and CLAMDA cards. See the following example.</td>
</tr>
<tr>
<td>CHEPS</td>
<td>This card is a single entry that specifies the altitude cut-off criterion. In the process of integrating a trajectory, the program iterates for the position and velocity of the vehicle as it crosses a defined layer. When a layer has been crossed, an iterative procedure is initiated; the nominal criterion for convergence is $10^{-6}$ earth-radii.</td>
</tr>
</tbody>
</table>

Example:

Given the following atmospheric drag model:

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>$C_{DA}/m$ (m$^2$/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>.008</td>
</tr>
<tr>
<td>100</td>
<td>.006</td>
</tr>
<tr>
<td>90</td>
<td>.003</td>
</tr>
<tr>
<td>85</td>
<td>.001</td>
</tr>
</tbody>
</table>

To solve for the drag at the two lower altitudes, the input cards that specify the trajectory model and the solution vector would look as follows:
ALTS = 120., 100., 90., 85.,
CLAMDA = .008, .006, .003, .001,
CATLM = 0, 0, 1, 1,

The maximum number of drag coefficients that may be solved for is fifteen. The six Category 1 variables must be included in the solution vector whenever drag coefficients are solved for. If the position and velocity are known sufficiently, further change during the differential correction process can be prevented by imposing small bounds. It should be noted that bounds must be specified for the drag solution variables; the bounds for the drag variables are entered on the BNDS card immediately following the state variables and preceding any Category 2 variables.

3.3.3 Steering Ephemeris

The steering ephemeris option is flagged by setting Column 54 of the JDC card. The radar parameters are computed from the reference trajectory and output for each sensor in the current sensor table. Two input variables are provided for the control of horizon limits and output units.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECRIT</td>
<td>The ECRIT card (critical elevation) specifies the elevation in degrees above which the steering ephemeris is printed. The nominal value is -5 degrees.</td>
</tr>
<tr>
<td>RDFLG</td>
<td>The RDFLG card is a flag for the output units of range rate and range acceleration in the steering ephemeris. If RDFLG = 0, the nominal setting, the output units of range rate and range acceleration are kilometers/minute and kilometers/(minute)^2 respectively; if RDFLG = 1, the units are kilometers/second and meters/(second)^2 respectively.</td>
</tr>
</tbody>
</table>

3.3.4 Linear Constraints

Two input arrays are required to specify a linear constraints case: BIJ and XIJ. The arrays define the constraint matrix, B, which is sparse. The set-up of the constraint matrix can be best explained by example.

Assume that there are $n$ parameters to be solved for, $(X_1, X_2, \ldots, X_n) = X$. The order of $X$ corresponds to the order of the solution variables; that is, the Category 1 variables, the drag variables (CATLM), and the
Category 2 variables. Also assume that there are \( m \) linear constraints to be placed on these variables. If \( n = 8 \) and \( m = 3 \), and the constraints are as listed below:

\[
\begin{align*}
X_3 &= X_4, \\
5X_5 &= X_6, \\
X_7 &= 2X_8
\end{align*}
\]

then the dimension of the constrained system is \( d = n - m = 5 \) for this example. Stating the problem in the functional form:

\[
X = BX
\]

where \( X \) is the vector of constrained variables.

\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
X_4 \\
X_5 \\
X_6 \\
X_7 \\
X_8
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 5 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0.5
\end{bmatrix}
\begin{bmatrix}
\bar{X}_1 \\
\bar{X}_2 \\
\bar{X}_3 \\
\bar{X}_4 \\
\bar{X}_5
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIJ</td>
<td>Each entry of the BIJ array is separated by a comma in the NAMELIST convention and corresponds to a non-zero element. The ( i, j )th element of the constraint matrix ( B ) is entered as ( 100i + j ). For example, if ( i = 12 ), and ( j = 9 ) is a non-zero element, it is entered as 1209 in the BIJ array. For the constraint matrix given above, the BIJ card would look like the following:</td>
</tr>
<tr>
<td>XIJ</td>
<td>The XIJ card is an array of numerical values of the non-zero elements of the constraint matrix ( B ). The input sequence of the XIJ entries is in a one-to-one correspondence with the BIJ array. The XIJ card for the constraint matrix given above is as follows:</td>
</tr>
</tbody>
</table>

\[
\text{BIJ} = 101, 202, 303, 403, 504, 604, 705, 805,
\]

\[
\text{XIJ} = 1., 1., 1., 1., 1., 5., 1., 0.5,
\]
3.4 SENSOR CARDS

The sensor card format of the NRTPD2 program is given on the following pages. There are six types of sensor cards, the first three of which are acceptable to the PREMOD and NRTPOD programs as described in Reference 1. The remaining three sensor cards have been added to provide the input for the functional standard deviation option.

The sensor identification is (up to) three alphanumerical characters entered in the first three columns. Two additional identifying columns are provided to permit biases to be different from pass to pass. The full identification parallels the format on the observation cards.

The type column indicates the type of information on the card according to the following key:

<table>
<thead>
<tr>
<th>Field</th>
<th>Locations</th>
<th>2</th>
<th>Standard Deviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field 4</td>
<td>Latitude</td>
<td>Azimuth bias</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>Field 5</td>
<td>Longitude</td>
<td>Elevation bias</td>
<td>$\sigma_E$</td>
</tr>
<tr>
<td>Field 6</td>
<td>Height</td>
<td>Range bias</td>
<td>$\sigma_R$</td>
</tr>
<tr>
<td>Field 7</td>
<td>$\dot{R}$ bias</td>
<td>$\dot{R}$ bias</td>
<td>$\sigma_{\dot{R}}$</td>
</tr>
<tr>
<td>Field 8</td>
<td>$\ddot{R}$ bias</td>
<td>$\ddot{R}$ bias</td>
<td>$\sigma_{\ddot{R}}$</td>
</tr>
<tr>
<td>Field 9</td>
<td>Station name</td>
<td>Time bias</td>
<td></td>
</tr>
<tr>
<td>Field 10</td>
<td>Station name</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Sensor Constants</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field 4</td>
<td>$B_A$</td>
<td>$\theta_1$</td>
<td>$f(\theta_1)$</td>
</tr>
<tr>
<td>Field 5</td>
<td>$B_E$</td>
<td>$\theta_2$</td>
<td>$f(\theta_2)$</td>
</tr>
<tr>
<td>Field 6</td>
<td>$B_R$</td>
<td>$\theta_3$</td>
<td>$f(\theta_3)$</td>
</tr>
<tr>
<td>Field 7</td>
<td>$\dot{B}_R$</td>
<td>$\theta_4$</td>
<td>$f(\theta_4)$</td>
</tr>
<tr>
<td>Field 8</td>
<td>$\ddot{B}_R$</td>
<td>$\theta_5$</td>
<td>$f(\theta_5)$</td>
</tr>
<tr>
<td>Field 9</td>
<td></td>
<td>$\theta_6$</td>
<td>$f(\theta_6)$</td>
</tr>
<tr>
<td>Field 10</td>
<td></td>
<td>$\theta_7$</td>
<td>$f(\theta_7)$</td>
</tr>
</tbody>
</table>

3-7
The data fields are each nine columns wide. Data may be input in any of the conventional FORTRAN arrangements, that is, with either implicit decimal point, and with or without a right adjusted exponent of ten preceded by a punched plus or minus sign. If the first column of a field is not punched -(minus), the contained value is assumed positive. The implicit decimal point is located between the first and second column of each field. On the card, implicit decimal points appear between the following pairs of columns: 9-10; 19-20; 29-30; 39-40; 49-50; 59-60; and 69-70.

<table>
<thead>
<tr>
<th>Field</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-3</td>
<td>Station identification</td>
</tr>
<tr>
<td>2</td>
<td>4-5</td>
<td>Pass number: applicable to Type 2 (biases) cards only</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blank or 0: Error on input, disregarded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Interpret $\phi$, $\lambda$, h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Interpret $A_b$, $E_b$, $R_b$, $\dot{R}_b$, $\ddot{R}_b$, $t_b$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: Interpret $\sigma_A$, $\sigma_{E'}$, $\sigma_R$, $\sigma_{\dot{R'}}$, $\sigma_{\ddot{R}}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4: Interpret $B_A$, $B_{E'}$, $B_{R'}$, $B_{\dot{R'}}$, $B_{\ddot{R}}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5: Interpret $\theta_1$, $\theta_2$, $\theta_3$, $\theta_4$, $\theta_5$, $\theta_6$, $\theta_7$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6: Interpret $f(\theta_1)$, $f(\theta_2)$, $f(\theta_3)$, $f(\theta_4)$, $f(\theta_5)$, $f(\theta_6)$, $f(\theta_7)$</td>
</tr>
<tr>
<td>4</td>
<td>9-17</td>
<td>Type 1: Geodetic latitude; $\phi$, degrees (positive north)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 2: Bias in Azimuth; $A_b$, degrees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 3: $\sigma_A$, degrees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 4: $B_A$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 5: $\theta_1$, degrees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 6: $f(\theta_1)$, $m^2$</td>
</tr>
<tr>
<td>5</td>
<td>19-27</td>
<td>Type 1: Longitude; $\lambda$, degrees (positive east of Greenwich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 2: Bias in Elevation; $E_b$, degrees</td>
</tr>
<tr>
<td>Field</td>
<td>Columns</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
|       | 28      | Type 3: $\sigma_{E'}$, degrees  
Type 4: $B_E$  
Type 5: $\theta_{E'}$, degrees  
Type 6: $f(\theta_{E'})$, m$^2$  
Space = blank |
| 6     | 29-37   | Type 1: Height; $h$, meters (positive above ellipsoid)  
Type 2: Bias in Range; $R_b$, km  
Type 3: $\sigma_{R'}$, km  
Type 4: $B_{R'}$  
Type 5: $\theta_{R'}$, degrees  
Type 6: $f(\theta_{R'})$, m$^2$  
Space = blank |
| 7     | 39-47   | Type 1:  
Type 2: Bias in first time derivative of range;  
$\dot{R}_b$, km/sec  
Type 3: $\sigma_{\dot{R}'}$, km/sec  
Type 4: $B_{\dot{R}'}$  
Type 5: $\theta_{\dot{R}'}$, degrees  
Type 6: $f(\theta_{\dot{R}'})$, m$^2$  
Space = blank |
| 8     | 49-57   | Type 1:  
Type 2: Bias in second time derivative of range;  
$\ddot{R}_b$, km/sec/sec  
Type 3: $\sigma_{\ddot{R}'}$, km/sec/sec  
Type 4: $B_{\ddot{R}'}$  
Type 5: $\theta_{\ddot{R}'}$, degrees  
Type 6: $f(\theta_{\ddot{R}'})$, m$^2$  
Space = blank |
| 9     | 59-67   | Type 1:  
Type 2: Bias in assigned time of observation;  
$\dot{t}_b$, sec  
Type 3:  
Type 4:  
Type 5: $\theta_{6'}$, degrees  
Type 6: $f(\theta_{6'})$, m$^2$  
Space = blank |
<table>
<thead>
<tr>
<th>Field</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>68</td>
<td>Space = blank</td>
</tr>
<tr>
<td>10</td>
<td>69-77</td>
<td>Type 1: Station Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 2:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 3:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 4:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 5: ( \theta_7 ), degrees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type 6: ( f(\theta_6) ), m^2</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>Space = blank</td>
</tr>
<tr>
<td>11</td>
<td>79-80</td>
<td>Not used, to be punched with some unambiguous mnemonic to identify this card conveniently as a sensor card.</td>
</tr>
</tbody>
</table>

The sensor constants are in mixed units so that the term \( B_1 R^4 / f(\theta) \) is dimensionally consistent with \( \sigma^2 \), where \( R \) is in earth-radii and \( f(\theta) \) in meters squared. See Section 2.2 for a more complete discussion of the functional standard deviation option.

3.5 OBSERVATION CARDS

The observation card format for the NRTPD2 program is the same as for the other Lincoln Laboratory orbit determination programs; the only difference is the program response. Since the NRTPOD program does not accept the \( \bar{R} \) observable, it will ignore these observations if input. See Section 1.2.6 of Reference 1 for a description of the observation cards.

3.6 EPHEMERIS CARDS

The format and use of the ephemeris cards is unchanged.
4. PROGRAM OUTPUT

The printed output of the NRTPOD modifications is explained and supplemented with samples in this section. The format is generally the same as NRTPOD; in most instances, the formats have been extended from or added to the existing version. A complete output guide is given below; however, if the format is unchanged from NRTPOD, it is not treated in this section.

<table>
<thead>
<tr>
<th>Data</th>
<th>Number of Pages</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDC Print</td>
<td>1</td>
<td>4.1</td>
</tr>
<tr>
<td>Input Listing</td>
<td>1</td>
<td>4.2</td>
</tr>
<tr>
<td>Run Header</td>
<td>1</td>
<td>4.3</td>
</tr>
<tr>
<td>Sensor Information</td>
<td>1</td>
<td>4.4</td>
</tr>
<tr>
<td>Observations</td>
<td>1 or more</td>
<td>4.5</td>
</tr>
<tr>
<td>Residuals</td>
<td>1 or more</td>
<td>4.6</td>
</tr>
<tr>
<td>Functional Standard Deviations</td>
<td>1 or more</td>
<td>4.7</td>
</tr>
<tr>
<td>Mean and RMS by Sensor and Type</td>
<td>1</td>
<td>4.8</td>
</tr>
<tr>
<td>Iteration Summary</td>
<td>1</td>
<td>4.9</td>
</tr>
<tr>
<td>Trajectory and Steering</td>
<td>1 or more</td>
<td>4.10</td>
</tr>
<tr>
<td>Matrix Update</td>
<td>1 or more</td>
<td>4.11</td>
</tr>
</tbody>
</table>

4.1 JDC PRINT

The JDC Print is the first page of a given run. Across the top of the page is a facsimile card image of the JDC card. Below this line is a JDC column content itemization and a short description of the option which is called by that particular JDC flag. Figure 4-1 is an example of the JDC print page.

4.2 INPUT LISTING

See Section 2.3.2 of Reference 1.

*Sample output not included in this section. See Section 2.3 of Reference 1.
<table>
<thead>
<tr>
<th>CARD COLUMN</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>1</td>
<td>SENSORS AND OBS</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>PRINT SENSORS</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>PRINT OBS</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>PRINT FUNCTIONAL SIGMA INPUT</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>CURVE FIT DESIRED</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>FUNCTIONAL SIGMA PRINT</td>
</tr>
<tr>
<td>51</td>
<td>1</td>
<td>TRAJECTORY PROPAGATION DESIRED</td>
</tr>
<tr>
<td>54</td>
<td>1</td>
<td>STEERING EPSHEMERIS WITH TRAJ PRINT</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>UPDATE DESIRED</td>
</tr>
</tbody>
</table>

Figure 4-1. Sample JDC Print
4.3 RUN HEADER

The run identification page remains essentially as in NRTPOD. The layered drag model, which is used in the integration of the trajectory, is listed under "DRAG MODEL." The selected value of the drag parameter ($C_D A/m$, $m^2/kg$) at a particular altitude (km) is listed. Figure 4-2 is a sample of the Run Header.

4.4 SENSOR INFORMATION

The sensor data associated with the functional standard deviation option can be printed on option. When Column 34 of the JDC card is flagged, a separate page titled "FUNCTIONAL SIGMA INPUT" is printed, following the Run Header. There are three lines of output for each sensor, representing the information contained on sensor card Types 4, 5, and 6 respectively. For each sensor having functional standard deviation input, the following is printed:

Line 1: $\text{ID } B_R B_{AZ} B_{EL} B_R B_R$

Line 2: $\text{ID } \theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6 \theta_7$

Line 3: $\text{ID } f(\theta_1) f(\theta_2) f(\theta_3) f(\theta_4) f(\theta_5) f(\theta_6) f(\theta_7)$

where the $B$'s are sensor dependent constants; $\theta$, the aspect angle; and $f(\theta)$, the radar cross section. These quantities are defined in Section 2.2.1. The second column refers to the type of sensor card. Figure 4-3 is a sample output of functional standard deviation information by sensor.

4.5 OBSERVATIONS

The observations print has been expanded to include the $\ddot{R}$ observables. A "TYPE" column has also been added between the "ID" and "T-TO" columns for the purpose of identifying the observable type. The Type 0 (zero) observables are $R$, $A$, $E$, and $\ddot{R}$. The Type 1 observable is $\dddot{R}$. This method of identifying observables is explained in Reference 1, Section 1.2.6. The units of $\ddot{R}$ as printed are meters/second/second. Figure 4-4 is a sample Observations Print.

4.6 RESIDUALS

The residuals print has been extended to include $\ddot{R}$ residuals. As in the observations print, a "TYPE" column has been added to signify the
## Figure 4-2. Sample Run Header
<table>
<thead>
<tr>
<th>ID</th>
<th>TYPE</th>
<th>θ R</th>
<th>θ A</th>
<th>θ E</th>
<th>θ R</th>
<th>θ R</th>
<th>θ A</th>
<th>θ E</th>
<th>θ A</th>
<th>θ E</th>
<th>θ A</th>
<th>θ E</th>
<th>θ A</th>
<th>θ E</th>
<th>θ A</th>
<th>θ E</th>
<th>θ A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2.5000E-05</td>
<td>6.6700E-06</td>
<td>6.6700E-06</td>
<td>2.5000E-09</td>
<td>5.6250E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.0000E-01</td>
<td>5.0000E-01</td>
<td>6.0000E-01</td>
<td>7.0000E-01</td>
<td>9.0000E-01</td>
<td>1.2000E+02</td>
<td>1.5000E+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.0000E+00</td>
<td>7.0000E+00</td>
<td>9.0000E+00</td>
<td>1.0890E+00</td>
<td>6.0000E+00</td>
<td>5.9500E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>3.5000E-05</td>
<td>7.6700E-06</td>
<td>7.6700E-06</td>
<td>3.5000E-09</td>
<td>6.6250E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.0000E+01</td>
<td>6.0000E+01</td>
<td>7.0000E+01</td>
<td>8.0000E+01</td>
<td>9.0000E+01</td>
<td>1.0000E+02</td>
<td>1.2000E+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5.0000E+01</td>
<td>8.0000E+01</td>
<td>8.0000E+01</td>
<td>8.0000E+01</td>
<td>8.0000E+01</td>
<td>8.0000E+01</td>
<td>4.0000E+00</td>
<td>4.0000E+00</td>
<td>3.0000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>4.5000E-05</td>
<td>8.6700E-06</td>
<td>8.6700E-06</td>
<td>4.5000E-09</td>
<td>7.6250E-03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5.0000E+00</td>
<td>7.0000E+00</td>
<td>8.0000E+00</td>
<td>9.0000E+00</td>
<td>1.1000E+02</td>
<td>1.3000E+02</td>
<td>1.5000E+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2.0000E+01</td>
<td>2.4000E+01</td>
<td>3.5000E+01</td>
<td>4.8000E+01</td>
<td>4.1000E+01</td>
<td>3.9000E+01</td>
<td>1.2000E+01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-3. Sample Functional Sigma Input Page
### Observation Type

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>T-TO</th>
<th>YR</th>
<th>MN</th>
<th>DY</th>
<th>HR</th>
<th>MIN</th>
<th>SECS</th>
<th>R</th>
<th>AZ SIGMA R</th>
<th>SIGMA A</th>
<th>EL SIGMA E</th>
<th>RDOT SIGMA R</th>
<th>R SIGMA R</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>0</td>
<td>0.400</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>23.998</td>
<td>1983.7720</td>
<td>278.8511</td>
<td>0.2217</td>
<td>-0.9296</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.400</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>23.998</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.115255E-00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>0.600</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>35.996</td>
<td>1900.6350</td>
<td>278.8350</td>
<td>1.0100</td>
<td>-0.9276</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.600</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>35.996</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.228848E-00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>0.800</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>47.995</td>
<td>1817.5310</td>
<td>278.8084</td>
<td>1.8360</td>
<td>-0.9240</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.800</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>47.995</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.359458E-00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>1.000</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>59.995</td>
<td>1734.4800</td>
<td>278.7813</td>
<td>2.7048</td>
<td>-0.9188</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.000</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>59.995</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.511309E 00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>1.200</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>11.992</td>
<td>1651.5020</td>
<td>278.7536</td>
<td>3.6231</td>
<td>-0.9117</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.200</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>11.992</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.689934E 00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>1.400</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>23.992</td>
<td>1568.6240</td>
<td>278.7252</td>
<td>4.5987</td>
<td>-0.9022</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.400</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>23.992</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.902679E 00</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>1.600</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>35.991</td>
<td>1485.8770</td>
<td>278.6960</td>
<td>5.6409</td>
<td>-0.8898</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.600</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>35.991</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.115940E 01</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>1.800</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>47.989</td>
<td>1403.2970</td>
<td>278.6658</td>
<td>6.7614</td>
<td>-0.8741</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.800</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>47.989</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.147352E 01</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>2.000</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>59.989</td>
<td>1320.9300</td>
<td>278.6346</td>
<td>7.9748</td>
<td>-0.8542</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.000</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>59.989</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.186363E 01</td>
</tr>
<tr>
<td>05</td>
<td>0</td>
<td>2.200</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>11.987</td>
<td>1238.8830</td>
<td>278.6520</td>
<td>9.2992</td>
<td>-0.8290</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.200</td>
<td>65</td>
<td>12</td>
<td>5</td>
<td>17</td>
<td>8</td>
<td>11.987</td>
<td></td>
<td>9.9999998E-01</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>10.00000E-03</td>
<td>0.0</td>
<td>0.100000E 00</td>
</tr>
</tbody>
</table>

Figure 4-4. Sample Observations Print
type of observable in each line of output. The units of \( \ddot{R} \) residuals are meters/second/second. Figure 4-5 is a sample Residuals Print.

### 4.7 FUNCTIONAL STANDARD DEVIATIONS

When the functional standard deviations option is used, the computed standard deviation as applied to each residual can be printed along with the residual output for each iteration of the differential correction. For each page of residuals, there is a corresponding functional standard deviations page. Both pages have the residual number (N), hence it is easy to match a functional standard deviation with its particular residual and observation. Figure 4-6 is a sample Functional Standard Deviations Print. The column symbols and their description are as follows:

<table>
<thead>
<tr>
<th>Column Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Observing station's identification</td>
</tr>
<tr>
<td>TYPE</td>
<td>Type of observable</td>
</tr>
<tr>
<td>0: RAER</td>
<td></td>
</tr>
<tr>
<td>1: ( \ddot{R} )</td>
<td></td>
</tr>
<tr>
<td>TIME (MIN)</td>
<td>Time in minutes from epoch</td>
</tr>
<tr>
<td>N</td>
<td>Serial number assigned to each residual for identification purposes. It is constant through the run.</td>
</tr>
<tr>
<td>SIGMA R (KM)</td>
<td>Computed standard deviation of range in kilometers</td>
</tr>
<tr>
<td>SIGMA A (DEG)</td>
<td>Computed standard deviation of azimuth in degrees</td>
</tr>
<tr>
<td>SIGMA E (DEG)</td>
<td>Computed standard deviation of elevation in degrees</td>
</tr>
<tr>
<td>SIGMA R. (KM/SEC)</td>
<td>Computed standard deviation of range rate in kilometers/second</td>
</tr>
<tr>
<td>SIGMA R. (MT/SEC**2)</td>
<td>Computed standard deviation of range acceleration in meters/second/second</td>
</tr>
</tbody>
</table>

### 4.8 MEAN AND STANDARD DEVIATIONS BY SENSOR

The mean and standard deviation of \( \ddot{R} \) data has been added to the \( R, A, E, \ddot{R} \) list. See Section 2.3.5 of Reference 1.
f?

/
<

—
o

OO

-* o o

o
o o
o o

o
o

K UJ

o

O

o o o ooo o o o
1
1
1 1 1

*-< c*l
o
«© «3 o m
at o m o IT
*0© o o
o o o o
.-H

H

>
.
2

o o o o o o o
O o o o o o o
© o o o o o o

o o o
o
o

o o
o o
ooo o
1

o OO o

o o

o

ooo ooo ooo

o •* a MOO o •t o 0* O 0s
o o o -4 © o o o o ml O •*
X o o o o o o o o o o a ooo o o o ooo
* oo o o o oo o ooo O o O ooo ooo
o o o o o oo o ooo OOO ooo ooo
moo O CO o r» o o
«on o * o >too
in o in O ITi o >o o >o
xoo o o o o o o o o o ooo
oo o © o OQO o o o o O o
1

1

oo o o
1

cc

'

1

1

CM

-.0
M fft

o
o
o
o
o o o o

</» -*

1*1

»»o
to
xo

1

<
a
—
UJ

or

O
t- UJ
O v>

•'

m

UJ

—

o

O

o

o
o

oo o o o

<

o
o

UJ © o
O
1

"•

x

a. *

«-

CO
fM

in

o
1

f*'
N
(V

1

1

CJ
rINI

«»•

CO
CM

o

m

CO
-<

Q
M

m
in
fM

1

1

1

o
o
o
o
o
o
ooo o

1

1

o o o
1 1 1

ooo
Til

o
o
o
o

o

o

O
O

OOO o
1
1

o o
1 1

o fl O «<g
o »o»
o t~ O !•»
o ooo
CJ ooo
1 T i i

1

1

1

o
o
o
o
o
o
ooo o o
1

9<M
mi
JO

fM
•*
CO
m

o
o
ooo

o o o
1
i

i

1

o

r>*
o
o
o

1

o
o
o
o
ooo
1

-»•

in

o
fo

<0
-O
o

»

1

1

o
o
o o ©
1

o

«M

ro

1

t

o
o
o
o
o
o
ooo o o

'

in
•C

co

ho

o

1

'

o
o
o
o
o
o
o o ooo

o
<M
-o

o o
1

r^
m «v
00
co
m
CM
M
o
o
o
o
o
o
o
o
o
ooo ooo

1

o

>»•

CM
o
o
o

CM

f

* mt m' •-J CM >-H ri> o m o en o fo ooo o •H O- ^
o mlQmt o -c O —i
o ooo ooo o

o

O 00 o
o -+ o
o CO o
o o o

ooo o

9
O
O
O

o o

o

mom o m o
ooo o in o
cr o «o o 9 o
ooo o o o
ooo ooo

ra»
o
o

o o o
1
i

i

<

o •* <•>
f>» o o
mo*
ooo
o ooo ooo ooo
1 1 1 t 1 I 1 i I i
a

o
•o
CM
o
o
o

o
o
o
o
o
o
o o ooo o
1
1
t

m
m
er>
fo*
o
<\J
(V
€V»
o
o
o
O
o
o
o
o
o
ooo ooo

o

o
o
o
o

-0

1

ml ml ml 9
o m, O m
o •a o V3
o o o o

o o o
1 i t

o
PH

*
*

mi
in
f\J

ooo

o
o
o
o

.-*
<_>

1

o •t o OOCO o p-4
o o> o -•> o •© o r>o o o N o •« o r~
ooo ooo o o

o
sO
PJ
o
o
o

in
in
fsl

o
o
o
o
o
o
o
o
o
o
ooo ooo ooo
1

o o

1

1

1

m

o
CD
CM
in
o

<o

1

ooo o o

>*

1

o
o

o

1

o
o
o
o
o
o
o . o
o . o
o
o
o ooo ooo

1

—

1

o

to mt tf\ ml f«- ml CO ml O —1 ml ml <*\ ml lf\ ml h- _J -e-* t> ~H CM
•*• O IA O >0 o 0> O •"« o ro O <o o o r> o -J O —i o
NOIV O IM o Nom o m o mom o rO o >»• om o m
o o © o o o OOO o © o ooo o O o ooo o o

0C1
CM

o
o
o
o -i
<=>
o
a: X
o
_ c o o o o
1

1

o ooo o

ooo

'

#-4

1

o o

o
O
o
o o

o J3 o <* -• M ^H CO o m o o» o f> o
o o © O O CM o o o o O —i o -. o
o o o ooo o o o ooo o o o
ooo ooo o o o ooo ooo

o ooo ooo OOO ooo ooo o o
1 t 1 1 1
1 1 * 1 1 1 1 1 1 1 1 1

9
I>-

!>fM

1

o o

1

O -0 o *H o
*» O fM O CO o
X Om 0*0

1

o o OOO
o o © o o
o o O O ©

^^

»4 —1 ml n\ —4
— oo o IM o
ION o CM o
3 «: oo o o o

•
•

o o ooo o Ci O
t
1

o m o CM o *iO>0
•». O »•* o o o OO r-

«:<*

ooo a
ooo o
ooo o
ooo o

t-l MSi-t 0v T-t O —1 Nnift •-4 0> ml t*~**Ki r-t rtfV mi h- •-H CM
o «-o CO O mi u ~« O r- o -O O 0>ON o MJ o o^o •* o CO
o «o in o >0 o f~ O >0 o r^ o NO^ o CO o CO o 0* o 0>
o OO O o o o ooo o o o ooo o o o ooo o o

1

>

o o
o a
o c

ooo
o o ©
© o o

»

~>
•#
IM

ml
in
in
CO
o

0>
>0
•<

o
o
o
o
o
o
ooo
1

a<
CO
in
o
o
o
o

1

*

a
-J

r-

o
o
o

o

o
1

r~
9
©

9
co
O

^^

1

o
1

m
ml

*-4

o
o
ooo

eo

r-

in
o
o
o

-H

o
o
o oo
1

m
Lfl

9

o
o
o
o
o
o
o
o
o
o
o
o
o
o
o
ooo ooo ooo

1

1

o

o
o

o

ooo

o o o

—

m

o

CO
CM

«

Q>

mt

<*l

oo
m
o
o
ooo

o o o © o o o o o ooo o o o
1
1
1
1
1
1
1
1

•c

o
o o o

©
o

o
o
o

•H

-o
-a
CM
in
co
m
•o
o
o
o
ooo

o

o o

.*•

o

-H
f-

ml

^4

r-

ro

o
o o

o

CT-

—*

-^

o -< o -< o

^-<

-

ooo

o p4 o m, a

o ml ml r-4

K

-< o

Q

in in in in m m m in in in m in m in in in m in in m m in in in m m in in m in in in in
© © o o o © o o o o o ooo o o o ooo ooo

•""•

o

o <-t o -J o

-J

O -l ©

4-8

o

O mi O

ml

UD

o
o
o
o
ooo o

a- 0 a- 9
9 9 9 0> 0>
C* 0* o 0* 0> 0> &• 0* CJN o- a a 0* 0s O 0- cr a- 9 9 9 cr o> o 9
«•» a o- 9 CT- 0- c o o> o> a- 9 0s CT* O a o> 0> 0> <J> o- a a C7- 0> 0^ 0* C7- a o O^ 9 9 a- 9 9 9
ZK1K1 in r- r- ff> 0> -< —« m m m m r~ I-- 0"> o ml ml tf\ m m in h- r- m ro m m SSff a- ml ml m

mt

to

0)

*

X f- r- r- r- r~ r- r~ oo eo co CO 00 CO' CO CD co co (MM> 0- a- cr o> o> o o o o

a

1

i

UJ

r

I—I

1

i

© ml u-\ JO o HH<I O -J in -o © :«* in <o o —i in NO o ml in •0 O ml in ~o o rtin< o —i m •e
00 00 00 en 9 0* 9 9 o o o O m1 ml f-J -H IM CM M r\) rr, tn m m «*• * Jin m m in o -o -c •o
(M r\j CM r>j IN eg <M f\l CM CM fM fM fM CM CM CM (M CM CM CM rM CM CM fM CM CM fM CM

i-

CO

l-l

o
p-l
o
o
o
o

z

CT-

•rJ

V

o
o
o
o
o
o
o
o
o
o
o
o
o
o
ooo ooo ooo ooo o

o

o o o

f~
©

1

o

1

m

m
(V
o
o
o
ooo

1

o
c

. •-<
<0

in

^J

o^ o
m m

in

00


4.9 ITERATION SUMMARY

This page shows the results and convergence status for a given iteration. The format and content of this page remains unchanged except that the Category 1 list has been extended to include the altitude dependent drag parameters. The drag parameters are labeled "LAMBDA 1, LAMBDA 2, . . . LAMBDA N, where LAMBDA 1 is the uppermost layer and LAMBDA N the lowest. The maximum number of layers which may be included in the solution vector is fifteen. Figure 4-7 is a sample iteration summary of a determination including the satellite position and velocity and drag parameters at nine altitudes in the solution vector. The iteration summary includes the normal matrix, the variance-covariance matrix, and the correlation matrix. Since the solution vector is of dimension 15, the matrix print of the three aforementioned matrices does not appear entirely on the first page. Hence, only part of the correlation matrix appears in Figure 4-7, the first page of the iteration summary.

4.10 TRAJECTORY AND STEERING

The trajectory output has been extended to include the Kozai mean elements at each print (DELT) time. Immediately following the standard trajectory print, the following mean elements are output:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Mean semi-major axis - earth-radii</td>
</tr>
<tr>
<td>E</td>
<td>Mean eccentricity - N. D.</td>
</tr>
<tr>
<td>I</td>
<td>Mean orbital inclination - degrees</td>
</tr>
<tr>
<td>NODE</td>
<td>Mean right ascension of the ascending node - degrees</td>
</tr>
<tr>
<td>OM</td>
<td>Mean argument of perihelion - degrees</td>
</tr>
<tr>
<td>M</td>
<td>Mean mean anomaly - degrees</td>
</tr>
<tr>
<td>NDOT</td>
<td>Rate of change of right ascension of ascending node - degrees/day</td>
</tr>
<tr>
<td>ODOT</td>
<td>Rate of change of argument of perihelion - degrees/day</td>
</tr>
</tbody>
</table>
### Iteration Summary

**Iteration Number 1**

<table>
<thead>
<tr>
<th>Category 1 Variables</th>
<th>Delta</th>
<th>Old</th>
<th>New</th>
<th>Sigma</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alpha</strong></td>
<td>-0.2234113E-06</td>
<td>0.1466796E 03</td>
<td>0.1466796E 03</td>
<td>0.44523887E-04</td>
<td>0.9999999E 00</td>
</tr>
<tr>
<td><strong>Delta</strong></td>
<td>-0.37836007E-05</td>
<td>0.10725830E 02</td>
<td>0.10725827E 02</td>
<td>0.74620796E-04</td>
<td>0.9999999E 00</td>
</tr>
<tr>
<td><strong>Beta</strong></td>
<td>-0.35990185E-04</td>
<td>0.11195949E 03</td>
<td>0.11195945E 03</td>
<td>0.12248758E-02</td>
<td>0.9999999E 00</td>
</tr>
<tr>
<td><strong>LAMBDA</strong></td>
<td>-0.019343250E-03</td>
<td>0.12493834E 04</td>
<td>0.12493830E 04</td>
<td>0.34371310E-02</td>
<td>0.9999999E 01</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td>0.034548036E-04</td>
<td>0.64700545E 01</td>
<td>0.64700890E 01</td>
<td>0.81705334E-03</td>
<td>0.20000000E 00</td>
</tr>
<tr>
<td><strong>LAMBDA 1</strong></td>
<td>-0.6066888E-04</td>
<td>0.3999999E 02</td>
<td>0.3999999E 02</td>
<td>0.29211566E-00</td>
<td>0.9999999E 02</td>
</tr>
<tr>
<td><strong>LAMBDA 2</strong></td>
<td>0.39860153E-03</td>
<td>0.3999999E 02</td>
<td>0.3999999E 02</td>
<td>0.15198049E-01</td>
<td>0.9999999E 02</td>
</tr>
<tr>
<td><strong>LAMBDA 3</strong></td>
<td>0.19079982E-03</td>
<td>0.1999999E 02</td>
<td>0.1999999E 02</td>
<td>0.18269469E-02</td>
<td>0.1999999E 02</td>
</tr>
<tr>
<td><strong>LAMBDA 4</strong></td>
<td>0.3104688E-03</td>
<td>0.2000000E 03</td>
<td>0.2000000E 03</td>
<td>0.27952631E-03</td>
<td>0.2000000E 03</td>
</tr>
<tr>
<td><strong>LAMBDA 5</strong></td>
<td>0.37571841E-04</td>
<td>0.9999999E 04</td>
<td>0.9999999E 04</td>
<td>0.68232170E-04</td>
<td>0.9999999E 04</td>
</tr>
<tr>
<td><strong>LAMBDA 6</strong></td>
<td>0.36120541E-04</td>
<td>0.9999999E 04</td>
<td>0.9999999E 04</td>
<td>0.20285668E-04</td>
<td>0.5999999E 04</td>
</tr>
<tr>
<td><strong>LAMBDA 7</strong></td>
<td>0.20552998E-04</td>
<td>0.9999999E 04</td>
<td>0.9999999E 04</td>
<td>0.64225130E-05</td>
<td>0.4999999E 04</td>
</tr>
<tr>
<td><strong>LAMBDA 8</strong></td>
<td>0.10146292E-04</td>
<td>0.3999999E 04</td>
<td>0.3999999E 04</td>
<td>0.28184474E-05</td>
<td>0.3999999E 04</td>
</tr>
</tbody>
</table>

**Solution is Converging**

**Solution is Affected by Bounds**

<table>
<thead>
<tr>
<th>Current RMS</th>
<th>1.060577</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted RMS</td>
<td>0.971998</td>
</tr>
<tr>
<td>Best RMS</td>
<td>1.060577</td>
</tr>
</tbody>
</table>

**Mean Elements from New**

\[ A 1.9886610D 00 \quad \text{NODE 2.65141090 02(DEG)} \quad \text{NODE -2.12855070-02(DAY)} \]

\[ E 1.15104660-02 \quad \text{OM 2.18432680 02(DEG)} \quad \text{OM -2.47965880-01(DAY)} \]

\[ I 0.86426130 01(DEG) \quad \text{M 1.15175180 02(DEG)} \]

**Correlation Matrix**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
</tr>
<tr>
<td>2</td>
<td>0.56526903</td>
<td>0.00000800</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>3</td>
<td>0.16520824</td>
<td>0.20959999</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>4</td>
<td>0.87413456</td>
<td>0.00117178</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
</tr>
<tr>
<td>5</td>
<td>0.34328789</td>
<td>0.00117178</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>6</td>
<td>0.30485194</td>
<td>0.00117178</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
<tr>
<td>7</td>
<td>0.31981392</td>
<td>0.00117178</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

Figure 4-7. Sample Iteration Summary Print (complete output not shown)
Figure 4-8 is a sample output of the trajectory print with mean elements.

The steering ephemeris appears in the trajectory and update print section, and is printed at each update time ($\Delta t$, $t$). Following the trajectory block, if it is requested, the following output constitutes the steering ephemeris:

**First Block**

The first block contains atmospheric parameters only (which are sensor independent); hence, the following quantities are printed once per update time.

<table>
<thead>
<tr>
<th>Column Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEIGHT</td>
<td>Altitude of spacecraft above reference ellipsoid in kilometers</td>
</tr>
<tr>
<td>(KM)</td>
<td></td>
</tr>
<tr>
<td>DENSITY</td>
<td>Density of the atmosphere at the altitude given in the previous column in kilograms/(meters)$^3$</td>
</tr>
<tr>
<td>(KG/M**3)</td>
<td></td>
</tr>
<tr>
<td>CDAM</td>
<td>$C_pA/M$, Ballistic coefficient at current altitude in (meters)$^2$/kilogram</td>
</tr>
<tr>
<td>(M**2/KG)</td>
<td></td>
</tr>
</tbody>
</table>

**Second Block**

In the second block of the steering output, there is one line of print for each sensor.

<table>
<thead>
<tr>
<th>Column Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATION</td>
<td>Sensor identification tag</td>
</tr>
<tr>
<td>AZIMUTH</td>
<td>Topocentric azimuth of vehicle from north, in degrees.</td>
</tr>
<tr>
<td>(DEG)</td>
<td></td>
</tr>
<tr>
<td>ELEVATION</td>
<td>Topocentric elevation of vehicle from horizon, in degrees.</td>
</tr>
<tr>
<td>(DEG)</td>
<td></td>
</tr>
<tr>
<td>RANGE</td>
<td>Topocentric range of vehicle in kilometers</td>
</tr>
<tr>
<td>(KM)</td>
<td></td>
</tr>
<tr>
<td>RDOT</td>
<td>Range rate of vehicle in kilometers/minute</td>
</tr>
<tr>
<td>(KM/MIN)</td>
<td></td>
</tr>
<tr>
<td>or RDOT</td>
<td>Range rate of vehicle in kilometers/second</td>
</tr>
<tr>
<td>(KM/SEC)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4-8: Sample Trajectory Print

**Table 1:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Epoch (Min.)</th>
<th>Distance (Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 March 1966</td>
<td>20 hr</td>
<td>8 min</td>
<td>42.96 sec</td>
</tr>
<tr>
<td>X</td>
<td>0.557236850</td>
<td>03 XDOT</td>
<td>-0.294519530</td>
</tr>
<tr>
<td>Y</td>
<td>0.205959350</td>
<td>05 YDOT</td>
<td>-0.422819770</td>
</tr>
<tr>
<td>Z</td>
<td>0.479308050</td>
<td>04 ZDOT</td>
<td>0.159989320</td>
</tr>
</tbody>
</table>

**Mean Elements:**

A 2.378209600 (DEG) 00 (ER) 00

E 3.95966130-01

I 3.22060160 01 (DEG)

**Table 2:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Epoch (Min.)</th>
<th>Distance (Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 March 1966</td>
<td>21 hr</td>
<td>8 min</td>
<td>42.96 sec</td>
</tr>
<tr>
<td>X</td>
<td>-0.912834130</td>
<td>04 XDOT</td>
<td>-0.208214150</td>
</tr>
<tr>
<td>Y</td>
<td>0.134419960</td>
<td>05 YDOT</td>
<td>-0.355324730</td>
</tr>
<tr>
<td>Z</td>
<td>0.862237520</td>
<td>04 ZDOT</td>
<td>0.322355590</td>
</tr>
</tbody>
</table>

**Mean Elements:**

A 2.378209600 (DEG) 00 (ER) 00

E 3.95966130-01

I 3.22060160 01 (DEG)

**Figure 4-8:** Sample Trajectory Print
<table>
<thead>
<tr>
<th>START TRAJECTORY</th>
<th>END TRAJECTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE RATE</td>
<td>PERIGEE RATE</td>
</tr>
<tr>
<td>4.6549908E 01</td>
<td>6.6425353E 01</td>
</tr>
</tbody>
</table>

22 SEPTEMBER 1994 13 HR 38 MIN 4.50 SEC MINUTES FROM EPOCH G. DAYS FROM OHR JAN 265,568138

X 0.0104139E 04 XDOT -0.1284317E 04 ALFA 0.1561151E 02 AZ 0.2360317E 03 ALT 0.6903801E 02
Y 0.1720593E 04 YDOT -0.5125584E 04 ULTA 0.1031445E 02 R 0.6505336E 04 LAT 0.1038043E 02
Z 0.1164358E 04 ZDOT 0.3619701E 01 BETA 0.1202012E 03 V 0.6404949E 01 LON 0.1696504E 01

SMA 0.4889446E 04 NWDE 0.1807307E 03 UX 0.9475505E 00 RPVX 0.2281962E 00 ALAT 0.6196381E 03
ECC 0.5794820E 00 UMG 0.3117501E 03 UY 0.2676623E 00 RPYV -0.5512309E 02 TAU 0.1719159E 02
INC 0.1446814E 03 M 0.2623625E 03 UZ 0.1796159E 00 RPVZ -0.3925918E 00 PRD 0.5670844E 02
L/A 0.1394725E 01 E 0.4156997E 00 APUG 0.7234023E 03 PRG -0.2331901E 04 ELLIPSE

<table>
<thead>
<tr>
<th>HEIGHT (KM)</th>
<th>DENSITY (KG/M**3)</th>
<th>CDAM (M**2/ KG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1279505E 03</td>
<td>0.8295629E-08</td>
<td>0.20485266E-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATION</th>
<th>AZIMUTH (Deg.)</th>
<th>ELEVATION (Deg.)</th>
<th>RANGE (KM)</th>
<th>RDOT (KM/MIN)</th>
<th>RDDOT (KM/MIN**2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>65.296174</td>
<td>24.671690</td>
<td>0.2932173E 03</td>
<td>-0.4022734E 03</td>
<td>-0.82043088E 01</td>
</tr>
<tr>
<td>TB</td>
<td>85.252126</td>
<td>48.068399</td>
<td>0.1705762E 03</td>
<td>-0.3539072E 03</td>
<td>0.19994877E 03</td>
</tr>
<tr>
<td>TC</td>
<td>124.330031</td>
<td>46.786778</td>
<td>0.1533012E 03</td>
<td>-0.2169709E 03</td>
<td>0.58047771E 03</td>
</tr>
<tr>
<td>TD</td>
<td>189.794375</td>
<td>53.627551</td>
<td>0.1579225E 03</td>
<td>-0.2328246E 02</td>
<td>0.10041345E 04</td>
</tr>
<tr>
<td>TE</td>
<td>184.361022</td>
<td>32.699956</td>
<td>0.2312485E 03</td>
<td>0.6196693E 02</td>
<td>0.67105278E 03</td>
</tr>
<tr>
<td>TF</td>
<td>210.841110</td>
<td>27.117791</td>
<td>0.2706715E 03</td>
<td>0.1923874E 03</td>
<td>0.44991525E 03</td>
</tr>
<tr>
<td>TG</td>
<td>205.887253</td>
<td>20.016548</td>
<td>0.3491978E 03</td>
<td>0.1994252E 03</td>
<td>0.34031614E 03</td>
</tr>
<tr>
<td>TH</td>
<td>32.712980</td>
<td>16.086232</td>
<td>0.4167752E 03</td>
<td>0.2563541E 03</td>
<td>0.22211976E 03</td>
</tr>
<tr>
<td>TI</td>
<td>14.146610</td>
<td>12.917997</td>
<td>0.4323313E 03</td>
<td>0.2517488E 03</td>
<td>0.19229797E 03</td>
</tr>
</tbody>
</table>

Figure 4-9. Sample Trajectory plus Steering Ephemeris Output
RDDOT (KM/MIN\(^2\)) Range acceleration of vehicle in
kilometers/minute/minute
or RDDOT (MT/SEC\(^2\)) meters/second/second

Figure 4-9 is an example of a trajectory plus steering ephemeris print. The entire output given is for a single point in time, as printed on the first line of output. In this particular example, the steering ephemeris is printed for nine sensors; the locations of the sensors which are included in the steering ephemeris are normally printed on the Header Page.

4.11 MATRIX UPDATE

The matrix update output now includes eigenvalues, associated eigenvectors, and determinants as well as the normal matrix and a "sigma and rho" matrix. At each update time, immediately following the normal matrix, the following quantities are output:

a) Eigenvalues and Eigenvectors. The uppermost elements of the six (6) columns of print are the square roots of the eigenvalues of the position partition (upper 3 x 3) and velocity partition (lower 3 x 3) of the UVW covariance matrix. The three components of the associated normalized eigenvector are printed below the respective eigenvalue.

b) Principal Axis Alignment. The ordered yaw-roll-pitch rotation to align the UVW system with the principal axes of the error ellipsoid are printed for both the position and velocity partitions of the UVW covariance matrix. The positive rotation of yaw-roll-pitch are left, clockwise, and down respectively.

c) Determinants. The square roots of the determinants of the following (sub) matrices are printed:

1) The position partition (upper 3 x 3) of the Cartesian covariance matrix
2) The velocity partition (lower 3 x 3) of the Cartesian covariance matrix
3) The 6 x 6 Cartesian covariance matrix

Figure 4-10 (two pages) is a sample matrix update for one update time. The matrix update example also includes steering ephemeris output which is printed between the trajectory block and the matrix update output. It should be noted that the output units of range rate and range acceleration
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Distance from Epoch</th>
<th>Days from CHR Jan 339.727081</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 December 1965 04</td>
<td>04.2696196</td>
<td>0.61212540E 04</td>
<td>0.69473498E 02</td>
</tr>
<tr>
<td>05</td>
<td>96.572600</td>
<td>-25.357612</td>
<td>0.61212540E 04</td>
</tr>
<tr>
<td>15</td>
<td>75.554939</td>
<td>-1.103245</td>
<td>0.62382565E 01</td>
</tr>
<tr>
<td>25</td>
<td>242.267208</td>
<td>-1.264774</td>
<td>0.64792875E 02</td>
</tr>
</tbody>
</table>

**Figure 4-10. Sample Printout of a Single Trajectory-Steering-Matrix Update Output Block**
### Normal Matrix Polar Spherical Coordinates

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.36342783E 09</td>
<td>0.36342783E 09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.13026765E 09</td>
<td>0.13768023E 09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.15898948E 09</td>
<td>0.65239861E 08</td>
<td>0.27614056E 09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.69494040E 07</td>
<td>0.21563090E 08</td>
<td>-0.13632600E 07</td>
<td>0.44456534E 08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.17655420E 07</td>
<td>0.71863795E 06</td>
<td>0.35197375E 07</td>
<td>-0.10597323E 05</td>
<td>0.57972430E 05</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.44337454E 09</td>
<td>-0.17841548E 09</td>
<td>0.32146509E 09</td>
<td>-0.67776295E 07</td>
<td>0.33367630E 07</td>
<td>0.96274466E 10</td>
</tr>
<tr>
<td>7</td>
<td>-0.16620645E 06</td>
<td>-0.69142073E 05</td>
<td>-0.40388336E 05</td>
<td>-0.36175053E 03</td>
<td>-0.91801873E 03</td>
<td>0.11276838E 07</td>
</tr>
<tr>
<td>8</td>
<td>-0.35613564E 05</td>
<td>-0.15515512E 05</td>
<td>0.82223288E 03</td>
<td>0.47241709E 02</td>
<td>0.99588989E 02</td>
<td>0.10229062E 06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>8</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.22696743E 03</td>
<td>0.71955980E 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.25070845E 02</td>
<td>0.71955980E 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### SQUARE ROOTS OF THE EIGENVALUES

|                  | 0.92123412E-02 | 0.13850620E-01 | 0.13647936E-01 | 0.17321800E-04 | 0.36008557E-04 | 0.21783436E-04 |

#### ASSOCIATED EIGENVECTORS OF THE UVW COVARIANCE MATRIX

|                  | 0.97622900E 00 | -0.20727696E 00 | 0.63349668E 01 | 0.88570171E 00 | 0.46394157E 00 | 0.17049901E 01 |
|                  | 0.21664581E 00 | 0.82449434E 00  | -0.31364799E 00| -0.46365653E 00| 0.88582748E 00| -0.18229607E 01|
|                  | 0.64452989E 02 | 0.31916385E 00  | 0.94742373E 00 | -0.23560698E 01| 0.82406808E 02 | 0.99968845E 00 |

#### TO ALIGN U,V,W WITH PRINCIPAL AXES

|                  | YAW LEFT     | ROLL CLOCKWISE | PITCH DOWN |
|                  | 19.3173 DEG  | -3.6321 DEG   | 11.9873 DEG |

#### TO ALIGN U,V,W WITH PRINCIPAL AXES

|                  | YAW LEFT     | ROLL CLOCKWISE | PITCH DOWN |
|                  | 1.0447 DEG   | -0.9769 DEG   | -27.6461 DEG|

#### SQUARE ROOTS OF DETERMINANTS OF CARTESIAN COVARIANCE MATRIX

|                  | POSITION     | VELOCITY     |
|                  | 0.17414310E-05 | 0.13587050E-13 | 0.86811937E-20 |

Figure 4-10. Sample Printout of a Single Trajectory-Steering-Matrix Update Output Block (Continued)
differ from the units as shown in Figure 4-8. See Section 3.3 regarding input requirements for steering ephemeris output units.

The diagonalized covariance matrix output described above is a permanent change to the program and is always printed when a matrix update is requested.
5. PROGRAM FUNCTIONAL DESCRIPTION

This section contains the subroutine descriptions of the NRTPD2 program. In addition to the standard subroutine descriptions, a complete subroutine glossary with functional descriptions is presented in Section 5.2. Because many of the subroutines were modified only slightly, a section describing the logic changes only has been included.

5.1 NRTPD2 SUBROUTINE OVERLAY

Figure 5-1 is a subroutine breakdown of the NRTPD2 overlay structure, although the overlay structure in terms of the principal options of the program remains unchanged. From this figure it can be seen which subroutines are in core as a function of the particular option of the program which is being used. The NRTPD2 subroutine overlay is similar to a corresponding diagram in Section 5.1 of Reference 1; the new subroutine and the ones which are included in one version only are identified for comparison purposes.
Figure 5-1. NRTPD2 Subroutine Overlay Structure
5.2 SUBROUTINE GLOSSARY

This section lists the subroutines which are used by the NRTPD2 program; that is, the list also includes the new subroutines which have been added as a result of the program modifications which this document describes.

Since many of the original subroutines remain unchanged, not all subroutine descriptions are given in this document. The documentation code which is explained below indicates how the program modifications affected the subroutine, and where the particular subroutine documentation is to be found. The following symbols constitute the documentation code:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Subroutine has been modified</td>
</tr>
<tr>
<td>N</td>
<td>New subroutine added as a result of new program options or system 360 conversion</td>
</tr>
<tr>
<td>U</td>
<td>Subroutine is unchanged</td>
</tr>
<tr>
<td>*</td>
<td>Subroutine modifications or descriptions given in Section 5.3.</td>
</tr>
<tr>
<td>( )</td>
<td>Reference document in which main subroutine documentation is to be found. The absence of ( ) indicates that the revised or new subroutine documentation is in Section 5.4.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Code</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOS</td>
<td>N*</td>
<td>Arc cosine routine</td>
</tr>
<tr>
<td>APPLY</td>
<td>M*(1)</td>
<td>Applies correction to solution vector and prints iteration summary</td>
</tr>
<tr>
<td>ASIN</td>
<td>U(2)</td>
<td>Arc sine routine</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>M</td>
<td>Establishes storage assignments for arrays in variable storage</td>
</tr>
<tr>
<td>ATM59</td>
<td>U(1)</td>
<td>Computes density of a static atmosphere (ARDC 1959 Model)</td>
</tr>
<tr>
<td>BCDOBS</td>
<td>M</td>
<td>Reads an observation card</td>
</tr>
<tr>
<td>BIJC</td>
<td>N</td>
<td>Sets up the linear constraints matrix in variable storage</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Code</td>
<td>Functional Description</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td>BNDSM</td>
<td>N</td>
<td>Sets up an appropriate set of bounds for the constrained solution vector</td>
</tr>
<tr>
<td>BODY</td>
<td>M</td>
<td>Computes perturbative acceleration due to sun and moon</td>
</tr>
<tr>
<td>BOUNDS</td>
<td>U(2)</td>
<td>Scales bounds with given scale factor</td>
</tr>
<tr>
<td>CDCD</td>
<td>N(1)</td>
<td>Modulates the input time</td>
</tr>
<tr>
<td>CLTIME</td>
<td>U(1)</td>
<td>Computes Gregorian time</td>
</tr>
<tr>
<td>CONST</td>
<td>N</td>
<td>Computes constants which are functions of input constants</td>
</tr>
<tr>
<td>CORMAT</td>
<td>U(1)</td>
<td>Computes correlation matrix</td>
</tr>
<tr>
<td>CTOM</td>
<td>N</td>
<td>Converts a Cartesian state vector to mean elements</td>
</tr>
<tr>
<td>CTOP</td>
<td>U(2)</td>
<td>Converts a Cartesian state vector to polar spherical coordinates (ADBARV)</td>
</tr>
<tr>
<td>DAUX</td>
<td>M</td>
<td>Driver for evaluating acceleration in integration</td>
</tr>
<tr>
<td>DCITER</td>
<td>U(1)</td>
<td>Driver for the computation of the normal matrix and one iteration of the DC</td>
</tr>
<tr>
<td>DLSTV</td>
<td>N</td>
<td>Computes the differentials for the MTOC and CTOM conversions</td>
</tr>
<tr>
<td>DOT</td>
<td>U(1)</td>
<td>Computes scalar product</td>
</tr>
<tr>
<td>DPRLM</td>
<td>M</td>
<td>Data initializing (partial)</td>
</tr>
<tr>
<td>DPROS</td>
<td>M</td>
<td>Driver for loading observation and sensor cards</td>
</tr>
<tr>
<td>DRAG</td>
<td>M</td>
<td>Computes drag perturbations</td>
</tr>
<tr>
<td>DRDP</td>
<td>M</td>
<td>Computes partials of observations w.r.t. category 1 variables</td>
</tr>
<tr>
<td>DSQRT</td>
<td>N*</td>
<td>Double precision square root routine</td>
</tr>
<tr>
<td>ELEM</td>
<td>N</td>
<td>Accesses the i,j element of a matrix stored in triangular form</td>
</tr>
<tr>
<td>EVERT</td>
<td>U(1)</td>
<td>Interpolates position of sun and moon</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Code</td>
<td>Functional Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>------------------------</td>
</tr>
<tr>
<td>FALSI</td>
<td>N</td>
<td>Determines altitude cutoffs</td>
</tr>
<tr>
<td>FIT</td>
<td>M</td>
<td>Logic control for DC options</td>
</tr>
<tr>
<td>FIXFLT</td>
<td>N*</td>
<td>Stores integer in floating array</td>
</tr>
<tr>
<td>FLTFIX</td>
<td>N*</td>
<td>Fixes a floating argument and stores it in a floating array</td>
</tr>
<tr>
<td>FSIGMA</td>
<td>N</td>
<td>Computes the functional standard deviation</td>
</tr>
<tr>
<td>FTHET</td>
<td>N</td>
<td>Interpolates the radar cross section from an input table</td>
</tr>
<tr>
<td>FVE</td>
<td>N</td>
<td>Determines the flags which indicate the ballistic coefficients which are to be solved for</td>
</tr>
<tr>
<td>GPERT</td>
<td>U(1)</td>
<td>Computes acceleration due to Earth's potential</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>N</td>
<td>Computes the altitude of a vehicle above the Earth</td>
</tr>
<tr>
<td>HINT</td>
<td>N</td>
<td>Defines the two drag layers which bound the vehicle at a given time</td>
</tr>
<tr>
<td>HUMAH</td>
<td>U(1)</td>
<td>Converts a vector, $A^T A$ matrix, or $(A^T A)^{-1}$ from machine units to human units or vice-versa</td>
</tr>
<tr>
<td>INPUT</td>
<td>M</td>
<td>Main driver for input processor</td>
</tr>
<tr>
<td>IPRNT</td>
<td>M*(1)</td>
<td>Prints header page</td>
</tr>
<tr>
<td>JACHIA</td>
<td>U(1)</td>
<td>Computes air density using Lockheed-Jacchia atmospheric model</td>
</tr>
<tr>
<td>JCS</td>
<td>U(1)</td>
<td>Sets up vector of zonal coefficients $J_2 \ldots J_{12}$ and two matrices of $C$'s and $S$'s for GPERT</td>
</tr>
<tr>
<td>JDCPRT</td>
<td>N</td>
<td>Prints the JDC card and describes the JDC options which have been flagged</td>
</tr>
<tr>
<td>JTOC</td>
<td>N(1)</td>
<td>Converts Julian date to calendar date</td>
</tr>
<tr>
<td>LEGS1</td>
<td>U(1)</td>
<td>Forms $A^T A$ and $A^T B$ given $A$ and $B$</td>
</tr>
<tr>
<td>LEGS2</td>
<td>M</td>
<td>Least squares package solves $AX = B$</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Code</td>
<td>Functional Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LESK</td>
<td>N</td>
<td>Complex linear equation solver</td>
</tr>
<tr>
<td>LINES</td>
<td>M*(1)</td>
<td>Ejects page and prints heading at top of page</td>
</tr>
<tr>
<td>LODOBS</td>
<td>M*(1)</td>
<td>Main control for observation card processor</td>
</tr>
<tr>
<td>LODSEN</td>
<td>M*(1)</td>
<td>Main control for sensor card processor</td>
</tr>
<tr>
<td>MABAT</td>
<td>U(1)</td>
<td>Multiplies $ABA^T$ where $B$ is a lower triangular matrix</td>
</tr>
<tr>
<td>MATCH</td>
<td>U(1)</td>
<td>Compares two floating point variables</td>
</tr>
<tr>
<td>MATMLT</td>
<td>N</td>
<td>Forms the matrix product of two matrices</td>
</tr>
<tr>
<td>MATPT</td>
<td>U(1)</td>
<td>Prints lower triangular matrix</td>
</tr>
<tr>
<td>MOVEVS</td>
<td>U(1)</td>
<td>Moves observation set from variable to working storage</td>
</tr>
<tr>
<td>MTOC</td>
<td>N</td>
<td>Converts mean elements to Cartesian coordinates</td>
</tr>
<tr>
<td>MULT</td>
<td>U(1)</td>
<td>Multiplies a $3 \times 3$ matrix by a succession of $1 \times 3$ vectors</td>
</tr>
<tr>
<td>NRTPOD</td>
<td>U(1)</td>
<td>Main control for NRTPOD</td>
</tr>
<tr>
<td>NTOM</td>
<td>N</td>
<td>Reduces a row of partial derivatives to the constrained system</td>
</tr>
<tr>
<td>NXN</td>
<td>N</td>
<td>Expands the constrained solution vector and $(A^TA)^{-1}$ to the unconstrained system</td>
</tr>
<tr>
<td>OBSIN</td>
<td>M</td>
<td>Moves observations from buffer to permanent storage</td>
</tr>
<tr>
<td>OBSSRT</td>
<td>U(1)</td>
<td>Sorts observations to time sequence</td>
</tr>
<tr>
<td>OUTER</td>
<td>U(1)</td>
<td>Computes product of column and row vector</td>
</tr>
<tr>
<td>PAGE1</td>
<td>M*(1)</td>
<td>Accumulates residuals and prints</td>
</tr>
<tr>
<td>PAGE2</td>
<td>N</td>
<td>Prints the functional standard deviations</td>
</tr>
<tr>
<td>PAROUT</td>
<td>M*(1)</td>
<td>Computes residuals for residuals print</td>
</tr>
<tr>
<td>PARSET</td>
<td>U(1)</td>
<td>Initializes station data for partial derivative package</td>
</tr>
<tr>
<td>PASTOR</td>
<td>U(1)</td>
<td>Monitors residual rejection</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Code</td>
<td>Functional Description</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td>PIMOD</td>
<td>U(1)</td>
<td>Modulates an argument between 0 and 2\pi</td>
</tr>
<tr>
<td>PLAMDA</td>
<td>N</td>
<td>Computes partial derivatives of position, velocity, and acceleration w.r.t. current drag parameters</td>
</tr>
<tr>
<td>PLTEL</td>
<td>U(2)</td>
<td>Converts polar elements to indeterminacy free and orbital elements</td>
</tr>
<tr>
<td>POTENT</td>
<td>U(1)</td>
<td>Driver for geopotential model</td>
</tr>
<tr>
<td>POPPC</td>
<td>N</td>
<td>Computes the transformation matrix from Cartesian to orbit plane (up, down, cross) coordinates</td>
</tr>
<tr>
<td>PPLPC</td>
<td>M</td>
<td>Computes partial of ADBARV w.r.t. Cartesian</td>
</tr>
<tr>
<td>PRAUPD</td>
<td>M</td>
<td>Updates a covariance matrix to a specified time</td>
</tr>
<tr>
<td>PRAXIS</td>
<td>N</td>
<td>Computes the eigenvalues and associated eigenvectors of real symmetric 3 x 3 matrix</td>
</tr>
<tr>
<td>PRCONS</td>
<td>U(1)</td>
<td>Prints program constants</td>
</tr>
<tr>
<td>PRECES</td>
<td>U(1)</td>
<td>Precesses lunar-solar ephemerides from mean of 1950.0 to true of epoch coordinates</td>
</tr>
<tr>
<td>PRELIM</td>
<td>M</td>
<td>Makes preliminary calculations in partials package</td>
</tr>
<tr>
<td>PRSSTB</td>
<td>M*(2)</td>
<td>Computes and prints mean, RMS, and number for residuals by sensor and type</td>
</tr>
<tr>
<td>PRTATA</td>
<td>M</td>
<td>Stores and prints the $A^T A$ matrix</td>
</tr>
<tr>
<td>PRUDRV</td>
<td>M</td>
<td>Main driver for trajectory print and update package</td>
</tr>
<tr>
<td>PTOC</td>
<td>U(2)</td>
<td>Converts polar coordinates to Cartesian coordinates</td>
</tr>
<tr>
<td>PUPB</td>
<td>M</td>
<td>Computes partials of observation w.r.t. category 2 variables</td>
</tr>
<tr>
<td>RADR</td>
<td>M</td>
<td>Computes residuals; driver for partials package</td>
</tr>
<tr>
<td>RADSQ</td>
<td>U(1)</td>
<td>Computes magnitude and $(\text{magnitude})^2$ of a 3-D vector</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Code</td>
<td>Functional Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td>RDDATA</td>
<td>M</td>
<td>Reads NAMELIST input cards, ephemeris cards, and mean elements cards</td>
</tr>
<tr>
<td>REJECT</td>
<td>M*(1)</td>
<td>Monitors the acceptance or rejection of an observation</td>
</tr>
<tr>
<td>RMAX</td>
<td>M*(1)</td>
<td>Compares residual quantities with table of maximum values</td>
</tr>
<tr>
<td>ROTRU</td>
<td>U(2)</td>
<td>Rotates a set of vectors from mean of 1950.0 to true of date coordinates</td>
</tr>
<tr>
<td>RPRESS</td>
<td>U(1)</td>
<td>Computes perturbative acceleration due to radiation pressure</td>
</tr>
<tr>
<td>SDELET</td>
<td>U(1)</td>
<td>Moves deletion list from buffer to permanent storage</td>
</tr>
<tr>
<td>SELECT</td>
<td>M</td>
<td>Selects next observation time</td>
</tr>
<tr>
<td>SENIN</td>
<td>U(1)</td>
<td>Moves sensor data from buffer to permanent storage</td>
</tr>
<tr>
<td>SENRD</td>
<td>M</td>
<td>Reads six types of sensor cards</td>
</tr>
<tr>
<td>SETCON</td>
<td>M</td>
<td>Sets constants for program</td>
</tr>
<tr>
<td>SETIC</td>
<td>M</td>
<td>Initializes integration list</td>
</tr>
<tr>
<td>SETSTR</td>
<td>U(1)</td>
<td>Converts drag and radiation pressure parameters from external to internal units</td>
</tr>
<tr>
<td>SETTAB</td>
<td>M*(1)</td>
<td>Sets tables concerning solution vector in variable storage</td>
</tr>
<tr>
<td>SSTB</td>
<td>M*(2)</td>
<td>Accumulates sum, sum of squares, and number of residuals by sensor and data type</td>
</tr>
<tr>
<td>STEER</td>
<td>N</td>
<td>Computes the radar steering ephemeris and prints the summary values</td>
</tr>
<tr>
<td>STSMAT</td>
<td>U(2)</td>
<td>Converts upper triangular S matrix from human units to machine units</td>
</tr>
<tr>
<td>SUPMAT</td>
<td>U(2)</td>
<td>Moves input update matrix from buffer to permanent storage</td>
</tr>
<tr>
<td>TGDJD</td>
<td>U(2)</td>
<td>Converts Julian to calendar date from integration time and prints</td>
</tr>
<tr>
<td>TINIT</td>
<td>U(2)</td>
<td>Sets up initial time, computes $\alpha_0$</td>
</tr>
</tbody>
</table>

5-8
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Code</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>U(2)</td>
<td>Converts Y, M, D, H, M, S to Julian date: days plus fraction</td>
</tr>
<tr>
<td>TMOD</td>
<td>N*</td>
<td>Modulates an argument</td>
</tr>
<tr>
<td>TPRLM</td>
<td>U(2)</td>
<td>Sets up data for integration</td>
</tr>
<tr>
<td>TPRNT</td>
<td>M*(2)</td>
<td>Prints trajectory output</td>
</tr>
<tr>
<td>TRAJ</td>
<td>M</td>
<td>Integrates the equations of motion and variational equations of motion to a specified time</td>
</tr>
<tr>
<td>TRJGEN</td>
<td>M</td>
<td>Main driver for trajectory package</td>
</tr>
<tr>
<td>TRJGET</td>
<td>M</td>
<td>Reads trajectory record from trajectory tape from DC package</td>
</tr>
<tr>
<td>TRJOUT</td>
<td>N</td>
<td>Prepares a variable length trajectory word for the trajectory tape</td>
</tr>
<tr>
<td>TRJPRO</td>
<td>M</td>
<td>Main driver for DC, trajectory, and update interfaces</td>
</tr>
<tr>
<td>TRJTAP</td>
<td>M</td>
<td>Writes trajectory tape</td>
</tr>
<tr>
<td>UBSGET</td>
<td>U(1)</td>
<td>Gets next observation time from variable storage</td>
</tr>
<tr>
<td>UPPER</td>
<td>N</td>
<td>Converts an N x N lower triangular matrix to an upper triangular matrix with an augmented column</td>
</tr>
<tr>
<td>UVECT</td>
<td>U(1)</td>
<td>Unitizes a 3-dimensional vector</td>
</tr>
<tr>
<td>VAREQ</td>
<td>U(1)</td>
<td>Computes second derivatives of variational equations</td>
</tr>
<tr>
<td>VPERT</td>
<td>U(1)</td>
<td>Initializes variational equations</td>
</tr>
<tr>
<td>WEOFT</td>
<td>U(1)</td>
<td>Writes an ending sentinel block on observation tape</td>
</tr>
<tr>
<td>WRTOBS</td>
<td>U(1)</td>
<td>Generates observation tape</td>
</tr>
<tr>
<td>XCROSS</td>
<td>U(2)</td>
<td>Performs the cross product of two 3-dimensional vectors</td>
</tr>
<tr>
<td>YRAE</td>
<td>U(2)</td>
<td>Computes Y vector when range, azimuth, and elevation are given</td>
</tr>
</tbody>
</table>
5.3 BRIEF SUBROUTINE DESCRIPTIONS

Brief descriptions of subroutine changes are given in this section. In most instances, the particular modifications are extensions and/or changes in the logic rather than input/output modifications. The parenthetical code following the subroutine name refers to the reference in which the major description of the particular subroutine is given, since it does not appear in this document. There are also five library or utility type subroutines which are described here; these particular subroutines exist in the System/360 version only.

ACOS
ACOS is a double precision arc cosine routine which is present in the System/360 version of NRTPD2 only. This function (single precision) exists as a library subroutine in the 7094 version.

APPLY(l)
The logic of APPLY was extended to accommodate the extended solution vector due to the multiple ballistic coefficients and the $\bar{R}$ sensor biases. The routine was modified for the output of mean elements in the iteration summary also.

DSQRT
DSQRT is a double precision square root routine which is present in the System/360 version of NRTPD2 only. This function (single precision) exists as a library subroutine in the 7094 version.

FIXFLT
FIXFLT is a routine which stores an integer (a fixed number) into a floating array. This subroutine is used in the System/360 version of NRTPD2 only.

FLTFIX
FLTFIX is a routine which fixes a floating number and stores it in a floating array. This subroutine is used in the System/360 version of NRTPD2 only.

IPRNT(l)
IPRNT has been modified to print the input $C_{DA/m}$ table as a function of altitude on the header page.

LINES(l)
The logic of LINES has been extended to print $\bar{R}$ residuals.

LODOBS(l)
The logic of LODOBS has been extended for the printing of $\bar{R}$ observations.

LODSEN(l)
The logic of LODSEN has been extended for the reading, processing, and printing of the three additional sensor cards.

PAGE1(l)
PAGE1 has been modified to convert and print $\bar{R}$ residuals.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAROUT(1)</td>
<td>PAROUT has been modified for the computation of residuals.</td>
</tr>
<tr>
<td>PRSSTB(2)</td>
<td>PRSSTB has been modified to calculate the mean and RMS of the $\bar{R}$ residuals by station.</td>
</tr>
<tr>
<td>REJECT(1)</td>
<td>The logic of REJECT has been extended to include $\bar{R}$ residuals editing.</td>
</tr>
<tr>
<td>RMAX(1)</td>
<td>The logic of RMAX has been extended for the processing of $\bar{R}$ residuals editing.</td>
</tr>
<tr>
<td>SETTAB(1)</td>
<td>SETTAB has been modified to include the $\bar{R}$ bias variables.</td>
</tr>
<tr>
<td>SSTB(2)</td>
<td>SSTB has been modified to accumulate the sum, sum of squares, and number of residuals of $\bar{R}$ data by station.</td>
</tr>
<tr>
<td>TMOD</td>
<td>TMOD is a routine which modulates argument A by argument B. This subroutine is used in the System/360 version of NRTPD2 only.</td>
</tr>
<tr>
<td>TPRNT(2)</td>
<td>TPRNT has been modified to print the mean elements and the steering ephemeris.</td>
</tr>
</tbody>
</table>
ASSIGN

5.4 SUBROUTINE DESCRIPTIONS

SUBROUTINE IDENTIFICATION

A. Title
ASSIGN

B. Segment
NRTPOD - INPUT PROCESSOR

C. Called by subroutine
INPUT

FUNCTION

The function is to establish storage assignments for the arrays to be located in variable storage (VSTR). This routine also establishes NPR, NDPR, and NICPR.

USAGE

A. Calling sequence
Call ASSIGN

B. Input
1. COMMON
   /INPP/ DATA (1000)

2. Calling sequence
   -

C. Output
1. COMMON
   NPR Total number of all parameters to solve for
   NDPR Number of differential and initial parameters to solve for (Category 1)
   NICPR Number of initial condition parameters to solve for
   NAROW Starting location where one row of augmented matrix (A, B) is stored
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NATA</td>
<td>Starting location of where the triangular $A^TA$ is stored</td>
</tr>
<tr>
<td>NBDNS</td>
<td>Starting location for the bounds used by LEGS</td>
</tr>
<tr>
<td>NDPAR1</td>
<td>Starting locations where the 4 sets of solution vectors will be stored</td>
</tr>
<tr>
<td>NDPAR2</td>
<td></td>
</tr>
<tr>
<td>NDPAR3</td>
<td></td>
</tr>
<tr>
<td>NDPAR4</td>
<td></td>
</tr>
<tr>
<td>NIDLED</td>
<td>Starting location of where the observation deletion table begins</td>
</tr>
<tr>
<td>NIDENT</td>
<td>Number of entries in the NIDLED list</td>
</tr>
<tr>
<td>NIDP</td>
<td>Identifier for table indicating CAT 1 type variables to be solved for</td>
</tr>
<tr>
<td>NPAR</td>
<td>Identifies the starting location for the parameter list</td>
</tr>
<tr>
<td>NPBIS</td>
<td>Identifies table for current estimates of CAT 2 variables</td>
</tr>
<tr>
<td>NPRCD</td>
<td>Identifies table for definition of CAT 2 variables to be solved for</td>
</tr>
<tr>
<td>NR</td>
<td>Starting location of where the inverse $A^TA$ is stored (in triangular form)</td>
</tr>
<tr>
<td>NRTMP</td>
<td>Identifies the starting location of temporary storage for special handling of the R matrix</td>
</tr>
<tr>
<td>NSCALE</td>
<td>Location of the list of conversion factors which convert all solution vectors and associated matrices from machine to output units and vice versa</td>
</tr>
<tr>
<td>NSTAT</td>
<td>Starting location of the master sensor table</td>
</tr>
<tr>
<td>VSTR</td>
<td>Floating point variable storage</td>
</tr>
<tr>
<td>MPR</td>
<td>Size of solution vector after constraint matrix has been applied. 0 if no constraints.</td>
</tr>
<tr>
<td>IMAX</td>
<td>Number of non-zero element of constraint matrix</td>
</tr>
<tr>
<td>CFLG</td>
<td>≠ 0 if additive constants are present in constraint problem.</td>
</tr>
<tr>
<td>MBNDS</td>
<td>Variable storage pointer for bounds corresponding to constrained system</td>
</tr>
<tr>
<td>NB</td>
<td>Variable storage pointer for constraint matrix</td>
</tr>
<tr>
<td>NC</td>
<td>Variable storage pointer for constraint matrix</td>
</tr>
<tr>
<td>NIJ</td>
<td>Variable storage pointer for indices of non-zero entries of constraint matrix</td>
</tr>
<tr>
<td>NST</td>
<td>Variable storage pointer for temporary cells used for linear constraints</td>
</tr>
<tr>
<td>NLAMS</td>
<td>Number of drag parameters in solution vector</td>
</tr>
<tr>
<td>NLID</td>
<td>Starting location in VSTR of the identifiers for the drag parameters appearing in the solution vector</td>
</tr>
</tbody>
</table>
2. Calling sequence

D. Error/action messages

---

**SUBROUTINES USED**

A. Library

---

B. Program

---

**EQUATIONS**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NICPR</td>
<td>Number of orbital elements to solve for</td>
</tr>
<tr>
<td>NDPR</td>
<td>CAT1 variables</td>
</tr>
<tr>
<td>NPR</td>
<td>CAT1 + CAT2</td>
</tr>
<tr>
<td>NLAMS</td>
<td>Number of drag solution variables</td>
</tr>
<tr>
<td>NDP</td>
<td>1</td>
</tr>
<tr>
<td>NLID</td>
<td>NICPR + NLAMS + NDP</td>
</tr>
<tr>
<td>NPRCD</td>
<td>NLAMS + NLID</td>
</tr>
<tr>
<td>NIDLED</td>
<td>NPRCD + NPR - NDPR</td>
</tr>
<tr>
<td>NIJ</td>
<td>NIDLED + NPR</td>
</tr>
<tr>
<td>NPBIS</td>
<td>1</td>
</tr>
<tr>
<td>NH</td>
<td>NPR - NDPR + NPBIS</td>
</tr>
<tr>
<td>NPALM</td>
<td>2 * NLAM + NH</td>
</tr>
<tr>
<td>NPXLM</td>
<td>6 * NLAM + NPALM</td>
</tr>
<tr>
<td>NPXDLM</td>
<td>6 * NLAM + NPXLM</td>
</tr>
<tr>
<td>NAROW</td>
<td>3 * NLAM + NPXDLM</td>
</tr>
<tr>
<td>NBNDS</td>
<td>NPR + NAROW + 1</td>
</tr>
<tr>
<td>NPAR</td>
<td>NBDNS + NPR</td>
</tr>
<tr>
<td>NDPAR1</td>
<td>2 * NPR + NPAR</td>
</tr>
<tr>
<td>NDPAR2</td>
<td>NPR + NDPAR1</td>
</tr>
<tr>
<td>NDPAR3</td>
<td>NPR + NDPAR2</td>
</tr>
<tr>
<td>NDPAR4</td>
<td>NPR + NDPAR3</td>
</tr>
<tr>
<td>NSCALE</td>
<td>NPR + NDPAR4</td>
</tr>
<tr>
<td>NB</td>
<td>NSCALE + NPR</td>
</tr>
<tr>
<td>NC</td>
<td>NB + IMAX</td>
</tr>
<tr>
<td>I</td>
<td>CFLG</td>
</tr>
<tr>
<td>MBNDS</td>
<td>NC + I * NPR</td>
</tr>
<tr>
<td>NST</td>
<td>MBNDS + MPR</td>
</tr>
</tbody>
</table>
ASSIGN

EQUATIONS

\[
\begin{align*}
\text{NATA} & = \frac{\text{NST} + \left[ \text{MPR} \left( \text{MPR} + 1 \right) \right]}{2} + \text{MPR} \\
\text{NR} & = \frac{\left( \text{NPR} + 1 \right) \times \left( \text{NPR} + 2 \right)}{2} + \text{NATA} \\
\text{NRTMP} & = \frac{\left( \text{NPR} + 2 \right) \times \left( \text{NPR} + 3 \right)}{2} - 1 + \text{NR} \\
\text{NSTAT} & = \frac{\left( \text{NPR} + 1 \right) \times \text{NPR}}{2} + 1 + \text{NRTMP}
\end{align*}
\]

FLOW CHART

See EQUATIONS for order of computation.
SUBROUTINE IDENTIFICATION

A. Title
BCDOBS

B. Segment
NRTPOD - INPUT PROCESSOR

C. Called by subroutine
LODOBS

FUNCTION

To read one observation card and process the estimated standard deviations carried on the observation cards. Additional functions include processing of types 1 and 2 observation cards (Lincoln Laboratory Format) and detecting the last observation card to be processed.

USAGE

A. Calling sequence
   Call BCDOBS (A, SEOF)

B. Input
   1. COMMON
      KOUT Symbolic output tape number
      KIN Symbolic input tape number

   2. Calling sequence
      -

C. Output
   1. COMMON
      A(1) Satellite ID (A)
      A(2) Year
      A(3) Month
      A(4) Day

5-17
2. Calling sequence

   SEOF       End of observation card read - signals end of observation data = ±1
   
   SEOF = -1  more obs to be processed
   SEOF = +1  no more obs to be processed

D. Error/action messages

1. Off line comment when program encounters type 2 observation cards:

   "PROGRAM IGNORES TYPE 2 OR GREATER OBSERVATION CARDS"

2. Action

   Program proceeds to process next observation card.

(A) Indicates alphanumeric
SUBROUTINE IDENTIFICATION

A. Title
   BIJC

B. Segment
   NRTPOD

C. Called by subroutine
   INPUT

FUNCTION

BIJC sets up the constraint matrix in variable storage by defining two arrays B and IJ. If IJ(k) = 100i + j, then B(k) contains the element b_{ij}.

USAGE

A. Calling sequence
   Call BIJC

B. Input
   1. COMMON
      VSTR | Variable storage
      IVSTR |
      MPR | Size of constrained system
      IMAX | Number of non-zero elements in constraint matrix
      NIJ | Variable storage pointer for vector of coded subscripts of b_{ij} matrix
      NB | Variable storage pointer for constraint matrix
      NC | VSTR pointer for additive constants
      CFLG | Additive constants flag

C. Output
   VSTR | Variable storage
   IVSTR |

D. Error/action messages
SUBROUTINES USED

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
   BNDSM

B. Segment
   NRTPOD

C. Called by subroutine
   FIT

FUNCTION

To obtain a set of bounds corresponding to an MPR x MPR system (constrained).

USAGE

A. Calling sequence
   Call BNDSM

B. Input
   
   1. COMMON
      
      NBNDS VSTR pointer for bounds (unconstrained)
      MPR Size of constrained system
      MBNDS VSTR pointer for bounds corresponding to constrained system
      IMAX Number of non-zero elements in constraint matrix
      NIJ IVSTR pointer for vector of coded subscripts of $b_{ij}$
      NB VSTR pointer for non-zero elements of constraint matrix

C. Output
   
   1. COMMON
      
      VSTR (MBNDS)

   2. Calling sequence

5-21
D. Error/action messages

SUBROUTINES USED

A. Library
   SQRT

B. Program

EQUATIONS

Given the diagonal bounds matrix $\beta$ with positive elements $\beta_k, k = 1, \ldots, NPR$, new bounds are computed. The diagonal bounds matrix of constrained variables has elements $\beta_j, j = 1, \ldots, MPR$ and are computed as shown below:

$$
\beta_j = \frac{1}{\sqrt{\sum_k \left( \frac{b_{kj}}{\beta_k} \right)^2}} \quad j = 1, \ldots, MPR \quad MPR < NPR
$$
SUBROUTINE IDENTIFICATION

A. Title
BODY

B. Segment
NRTPOD

C. Called by subroutine
DAUX

FUNCTION

The function is to compute the perturbative acceleration of a spacecraft due to other bodies in the solar system and to account for these effects in the variational equations.

USAGE

A. Calling sequence
Call BODY

B. Input
1. COMMON
   TLIST Current integration list
   DBASE Days from 1950.0 to midnight day of epoch
   CMU GM of Earth (e.r. \( \frac{3}{\text{min}^2} \))
   CGMR Ratio of moon, sun GM to that of the Earth
   FLVE Flag to skip computation of variational equations
   BFLAGS Flags to indicate whether the accelerations of the moon and sun are to be considered
   NDAYNS NAMELIST input variable denoting the number of days of lunar-solar ephemeris input.
   NDPR Number of CAT1 variables in solution vector

2. Calling sequence
C. Output

1. COMMON

T3PERT  The total acceleration of the vehicle due to all the desired bodies

PMAT    Matrix of the position dependent effects in the variational equations (the body effects are added to this matrix)

XN      Cartesian position of Moon and Sun

2. Calling sequence

SUBROUTINES USED

A. Library

B. Program

EVERT
RADSO
OUTER

EQUATIONS

The position of the Moon and Sun with respect to the Earth, \( x_i \), \( y_i \), \( z_i \), is obtained from the ephemeris cards.

\[
R_i = \left( x_i^2 + y_i^2 + z_i^2 \right)^{1/2}
\]

\[
x_{vi} = x_v - x_i
\]

\[
y_{vi} = y_v - y_i
\]

\[
z_{vi} = z_v - z_i
\]

where \( x_v \), \( y_v \), \( z_v \) is the position of the vehicle with respect to the earth.

\[
R_{vi} = \left( x_{vi}^2 + y_{vi}^2 + z_{vi}^2 \right)^{1/2}
\]

\[
x_{\text{bodies}} = -\sum_{i=1}^{n} \mu_i \left( \frac{(x_v - x_i)}{R_{vi}^3} + \frac{x_i}{R_i^3} \right)
\]
\[ \dot{y}_{\text{bodies}} = - \sum_{i=1}^{n} \mu_i \left[ \frac{(y_v - y_i)}{R_{vi}} + \frac{y_i}{R_i} \right] \]
\[ \dot{z}_{\text{bodies}} = - \sum_{i=1}^{n} \mu_i \left[ \frac{(z_v - z_i)}{R_{vi}} + \frac{z_i}{R_i} \right] \]

\[
\begin{align*}
\text{PMAT} &= \begin{bmatrix}
\sum_{i=1}^{n} \mu_i \left( \frac{3x^2_{vi}}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3x_{vi} y_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3x_{vi} z_{vi}}{R_{vi}^5} \right) \\
\sum_{i=1}^{n} \mu_i \left( \frac{3x_{vi} y_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3y^2_{vi}}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3y_{vi} z_{vi}}{R_{vi}^5} \right) \\
\sum_{i=1}^{n} \mu_i \left( \frac{3x_{vi} z_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3y_{vi} z_{vi}}{R_{vi}^5} \right) & \sum_{i=1}^{n} \mu_i \left( \frac{3z^2_{vi}}{R_{vi}^5} - \frac{1}{R_{vi}^3} \right)
\end{bmatrix}
\end{align*}
\]
Compute
TT - Time in
Days from 1950

Call
EVERT

Compute
\( \ddot{x}_p, \dddot{y}_p, \dddot{z}_p \)
perturbative accelerations of sun + moon

\[ \begin{align*}
PMAT (1) &= PMAT (1) + D \\
PMAT (5) &= PMAT (5) + D \\
PMAT (9) &= PMAT (9) + D \\
\end{align*} \]

Figure 5-2. BODY Flow Diagram
SUBROUTINE IDENTIFICATION

A. Title
   CONST

B. Segment
   NRTPOD

C. Called by subroutine
   INPUT

FUNCTION

Subroutine CONST computes those constants which are functions of other constants that may be altered at input time.

USAGE

A. Calling sequence
   CALL CONST

B. Input
   1. COMMON
      CKMER     Conversion from earth radii to kilometers
      CFTER     Conversion from earth radii to feet
      CFTNM     Conversion from nautical miles to feet
      CELLIP    Ellipticity of the earth
   2. Calling sequence

C. Output
   1. COMMON
      CMTER     Conversion from earth radii to meters
      CNMER     Conversion from earth radii to nautical miles
      CBE       Semi-minor axis of the earth (earth radii)
      TRM1      First term used in the computation of the "radius at sea level" (See equations.)
      TRM2      Second term used in the computation of the "radius at sea level" (See equations.)
2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library
   EXP

B. Program

EQUATIONS

\[
\text{CMTER} = \text{CKMER} \times 1000.
\]
\[
\text{CNMER} = \text{CFTER}/\text{CFTNM}
\]
\[
\text{CBE} = 1.0 - \text{CELLIP}
\]

The distance from the center of the earth to the surface of the ellipsoid at geocentric latitude \( \varphi \), termed "radius at sea level", is given by,

\[
\rho = a_e \left( 1 + k \sin^2 \varphi \right)^{-1/2}
\]

where \( a_e \) is the earth equatorial radius and

\[
k = \frac{2 - \epsilon}{(1 - \epsilon)^2} \epsilon
\]

is a constant derived from ellipticity, \( \epsilon \).

Utilizing a series expansion along with an introduction of a shifted Chebyshev polynomial of order three, an approximation for the radius at sea level is obtained.

\[
\rho \approx a_e \left[ 1 - \left( \frac{k}{2} - \frac{45}{256} k^3 \right) S^2 + \left( \frac{3}{8} k^2 - \frac{15}{32} k^3 \right) S^4 \right]
\]

where

\[
\sin \varphi = S
\]

5-28
This routine computes the constant terms in the approximation for $\rho$

$$TRM1 = \frac{k}{2} - \frac{45}{256} k^3$$

$$TRM2 = \frac{3}{8} k^2 - \frac{15}{32} k^3$$
SUBROUTINE IDENTIFICATION

A. Title
   CTOM

B. Segment
   NRTPOD

C. Called by subroutines
   APPLY
   TPRNT

FUNCTION

To convert a set of Cartesian elements to osculating elements and then to mean elements.

USAGE

A. Calling sequence
   Call CTOM(TNOMX, ADBAR, ITER)

B. Input

1. COMMON

   CJ2      J2 Earth Harmonic
   CMU      μ (Earth Radii, Minutes)
   CPI      π
   C2PI     2π
   KOUT     Output tape number

2. Calling sequence

   TNOMX(1)  x
   TNOMX(2)  y
   TNOMX(3)  z
   TNOMX(4)  x
   TNOMX(5)  y
   TNOMX(6)  z
   ITER     Number of iterations to be used to calculate δ's; see subroutine DLSTV

C. Output

1. COMMON

   None

5-31
2. Calling sequence

ADBAR(1) \( a_{K-25} \) (Earth Radii)
ADBAR(2) \( e \)
ADBAR(3) \( i \)
ADBAR(4) \( \Omega \)
ADBAR(5) \( \omega \) (Radians)
ADBAR(6) \( M \)
ADBAR(7) \( \dot{\omega} \) (Radians/Minute)
ADBAR(8) \( \dot{\Omega} \) (Radians/Minute)

D. Error Messages

If \( E \) (eccentric anomaly) fails to converge after 50 iterations

E FAILED TO CONVERGE

THE VALUE OF \( E \) IS ±, XXXXXE ±XX

The computation proceeds with the last computed value of \( E \).

SUBROUTINES USED

A. Library

ABS
SQRT
ATNQ
SIN
COS

B. Program

PIMOD
To set the principle value of an angle between 0 and \( 2\pi \)

DLSTV
To compute the \( \delta \)'s for conversion from osculating to mean and mean to osculating

EQUATIONS

1. Compute epoch values of

a) magnitude of radius vector

\[ r_o = \sqrt{x^2 + y^2 + z^2} \]

b) Angular momentum

\[ h_o^2 = (y\dot{z} - z\dot{y})^2 + (z\dot{x} - x\dot{z})^2 + (x\dot{y} - y\dot{x})^2 \]
c) Orbital semi-parameter

\[ p_o = \frac{h_o^2}{\mu} \]

2. Compute osculating orbital elements.

a) Semi-major axis

\[ a_{os} = \frac{r_o \mu}{\left[2\mu - r_o (x^2 + y^2 + z^2)\right]} \]

b) Eccentricity

\[ e_{os} = \pm \sqrt{(p_o/r_o - 1)^2 + \frac{p_o}{\mu r_o^2}(x\dot{x} + y\dot{y} + z\dot{z})^2} \]

c) The true anomaly \( \nu_o \)

\[ e_{os} \cos \nu_o = \left(p_o/r_o - 1\right) \]
\[ e_{os} \sin \nu_o = \sqrt{\frac{p_o}{\mu}} \left(x\dot{x} + y\dot{y} + z\dot{z}\right) \]

d) The orbital inclination and the longitude of the ascending node

\[ \sin i_{os} \sin \Omega_{os} = \frac{y\dot{z} - z\dot{y}}{h_o} \]
\[ \sin i_{os} \cos \Omega_{os} = \frac{x\dot{z} - z\dot{x}}{h_o} \]
\[ \cos i_{os} = \frac{x\dot{y} - y\dot{x}}{h_o} \]

e) The argument of latitude, \( u_o \), is determined from

\[ \cos u_o = \frac{x}{r_o} \cos \Omega_{os} + \frac{y}{r_o} \sin \Omega_{os} \]
\[ \sin u_o = \frac{z}{r_o \sin i_{os}} \]

f) The argument of perigee

\[ \omega_{os} = u_o - \nu_o \]
g) The eccentric anomaly

\[ e_{os} \sin E_o = (xx + yy + zz) / \sqrt{\mu a_{os}} \]

\[ e_{os} \cos E_o = (1 - r_o / a_{os}) \]

h) The mean anomaly

\[ M_{os} = E_o - e_{os} \sin E_o \]

Compute the initial K-25 element where \( A_2 = \frac{3}{2} J_2 a_e^2 \)

\[ a_{osK-25} = a_{os} \left[ 1 - \frac{A_2}{p_o^2} \left( 1 - \frac{3}{2} \sin^2 i_{os} \right) \left( \sqrt{1 - e_{os}^2} \right) \right] \]

Compute the initial K-25 p

\[ p_o = a_{osK-25} \left( 1 - e_{os}^2 \right) \]

Then iterate on the following index, k

\[ a_{K-25_k} = a_k \left[ 1 - \frac{A_2}{p_{k-1}^2} \left( 1 - \frac{3}{2} \sin^2 i_k \right) \left( \sqrt{1 - e_k^2} \right) \right] \]

\[ p_k = a_{K-25_k} \left( 1 - e_k^2 \right) \]

Compute \( \delta \)'s using \( (a_{K-25_k}, e_k, i_k, \Omega_k, \omega_k, M_k) \)

Compute

\[ a_k = a_{os} - \delta a_k \]

\[ e_k = e_{os} - \delta e_k \]

\[ i_k = i_{os} - \delta i_k \]

\[ \Omega_k = \Omega_{os} - \delta \Omega_k \]

5-34
\[ \omega_k = \omega_\text{o,k} - \delta \omega_k \]

\[ M_k = M_\text{o,k} - \delta M_k \]

\[ r = r_0. \text{ Iterate Kepler's equations (see MTOC) to find } E \text{ and } \nu \text{ after each iteration. After the last iteration, the mean values are:} \]

\[ a_{K-25_m} = a_{K-25_k} \]

\[ e_m = e_k \]

\[ i_m = i_k \]

\[ \Omega_m = \Omega_k \]

\[ \Omega_m = \Omega_k \]

\[ \omega_m = \omega_k \]

\[ M_m = M_k \]

After iterating, compute the secular rates of \( \omega \) and \( \Omega \)

\[ \dot{\omega}_m = \frac{A_2}{2 \, p_k} \sqrt{\frac{\mu}{3 \, a_{K-25_k}}} \left( 2 - \frac{5}{2} \sin^2 i_k \right) \]

\[ \left\{ 1 - \frac{A_2}{2 \, p_k} \left( 1 - \frac{3}{2} \sin^2 i_k \right) \sqrt{1 - e_k^2} \right\}^{1/2} \]

\[ \dot{\Omega}_m = -\frac{A_2}{2 \, p_k} \sqrt{\frac{\mu}{3 \, a_{K-25_k}}} \cos i_k \]

\[ \left\{ 1 - \frac{A_2}{2 \, p_k} \left( 1 - \frac{3}{2} \sin^2 i_k \right) \sqrt{1 - e_k^2} \right\}^{1/2} \]
SUBROUTINE IDENTIFICATION

A. Title
DAUX

B. Segment
NRTPOD

C. Called by subroutine
TRAJ
SETIC

FUNCTION

The function is to compute the second derivatives in the equations of motion and control the computation of the second derivatives in the variational equations.

USAGE

A. Calling sequence
Call DAUX

B. Input
1. COMMON

TLIST   Numerical integration working storage
SGAMAM  Constant used in calculating radiation pressure effects, $S_yA/m (e.r.3/min^2)$
CDAD2M  Drag parameter $C_D A/2m (ft^2/slug)$
FLVE    Variational equation control flag $\neq 0$ compute variational equations
TBPRT   Acceleration due to bodies $(e.r./min^2)$
TPOT    Acceleration due to aspherical potential $(e.r./min^2)$
TDRAG   Acceleration due to drag $(e.r./min^2)$
TRPRES  Acceleration due to radiation pressure $(e.r./min^2)$
TR      Magnitude of geocentric position vector, $R(e.r.)$
CMU     $GM\ earth\ (e.r.3/min^2)$
NDPR    Total number of CATEGORY 1 variables to solve for
NLAM    Total number of entries in the altitude $C_DA/m$ table

5-37
 Calling sequence

1. COMMON
   TLIST (58-60) Numerical integration working storage containing the total acceleration

2. Calling sequence

SUBROUTINES USED

A. Library

B. Program
   BODY
   DRAG
   POTENT
   RADSQ
   RPRESS
   VAREQ

EQUATIONS

The Cowell formulation of the equations of motion is used:

\[ R = \left( x^2 + y^2 + z^2 \right)^{1/2} \]

\[ \ddot{x} = -\frac{\mu}{R^3} + \ddot{x}_{\text{bodies}} + \ddot{x}_{\text{drag}} + \ddot{x}_{\text{potential}} + \ddot{x}_{\text{radiation pressure}} \]

\[ \ddot{y} = -\frac{\mu}{R^3} + \ddot{y}_{\text{bodies}} + \ddot{y}_{\text{drag}} + \ddot{y}_{\text{potential}} + \ddot{y}_{\text{radiation pressure}} \]

\[ \ddot{z} = -\frac{\mu}{R^3} + \ddot{z}_{\text{bodies}} + \ddot{z}_{\text{drag}} + \ddot{z}_{\text{potential}} + \ddot{z}_{\text{radiation pressure}} \]
where

\[ x_{\text{bodies}} = \text{The perturbation acceleration due to other bodies in the solar system} \]

\[ x_{\text{drag}} = \text{The perturbation acceleration due to atmosphere drag} \]

\[ x_{\text{potential}} = \text{The perturbation acceleration due to the potential field set by the aspherical earth} \]

\[ x_{\text{radiation pressure}} = \text{The perturbation acceleration due to solar radiation pressure} \]

The tests are made to see which of the above perturbation effects are to be included in the evaluation of the equations of motion.
Zero out accelerations

\[
\begin{align*}
TBERT(I) &= 0 \\
TPOT(I) &= 0 \\
TDRAG(I) &= 0 \\
TRPRES(I) &= 0 \\
\end{align*}
\]

I = 1, ... 3

Compute \( R, R^2, R^3, R^5, R^7 \)

Call \texttt{BODY}

\( \& \) \texttt{SGAMAM = 0}

Call \texttt{RPRESS}

Call \texttt{POTENT}

\( \& \) \texttt{CDAD2m = 0}

Call \texttt{DRAG}

Compute 2-body and total acceleration

\( \# \) \texttt{FLVE:0}

Call \texttt{VAREQ}

RETURN

Figure 5-3. DAUX Flow Diagram
DLSTV

SUBROUTINE IDENTIFICATION

A. Title
   DLSTV

B. Segment
   NRTPOD

C. Called by subroutines
   CTOM
   MTOC

FUNCTION

To compute the differentials used to convert from osculating to mean and mean to osculating.

USAGE

A. Calling sequence
   Call DLSTV (STATE, R, V, E, DELTA)

B. Input

   1. COMMON
      CJ2   J2 Earth Harmonic

   2. Calling sequence
      Osulating or Mean
      STATE(1) a_oK-25 a_mK-25 (Earth Radii)
      STATE(2) e_o e_m
      STATE(3) i_o i_m
      STATE(4) \Omega_o \Omega_m (Radians)
      STATE(5) \omega_o \omega_m
      STATE(6) M_o M_m

      R   Magnitude of radius vector (Earth Radii)
      V   True anomaly (Radians)
      E   Eccentric anomaly (Radians)

C. Output

   1. COMMON
      None

5-41
2. Calling sequence

\[
\begin{align*}
\delta_a & = \text{DELTA(1)} \\
\delta_e & = \text{DELTA(2)} \\
\delta_i & = \text{DELTA(3)} \\
\delta_{\Omega} & = \text{DELTA(4)} \\
\delta_\omega & = \text{DELTA(5)} \\
\delta_M & = \text{DELTA(6)}
\end{align*}
\]

D. Error Messages

None

SUBROUTINES USED

A. Library

\[
\begin{align*}
\text{SIN} \\
\text{COS} \\
\text{SQRT}
\end{align*}
\]

B. Program

None

EQUATIONS

Equations (2), (5), and (6) have been formulated to preserve numerical accuracy when eccentricity is near zero, and hence do not appear as in the standard references, Kozai, et al.

\[
\begin{align*}
da &= \frac{A_2}{a} \left[ \frac{2}{3} \left( 1 - \frac{3}{2} \sin^2 i \right) \left( \frac{a}{r} \right)^3 \left( 1 - e^2 \right)^{-3/2} \right] \\
&+ \left( \frac{a}{r} \right)^3 \sin^2 i \cos 2(v + \omega) \\
&\quad + \left( \frac{a}{r} \right)^3 \sin^2 i \cos 2(v + \omega) \\
&+ \left( \frac{a}{r} \right)^3 \sin^2 i \cos 2(v + \omega)
\end{align*}
\]

(1)

\[
\begin{align*}
de &= \frac{A_2}{a^2} \left[ (1 - e^2) \left[ \frac{1}{3} \left( 1 - \frac{3}{2} \sin^2 i \right) (S_5 - S_3) \right] + \frac{\sin^2 i \cos 2u S_6}{2} \\
&- \frac{\sin^2 i \cos (v + 2\omega)}{2(1-e^2)} \left( \cos (v + 2\omega) + \frac{1}{3} \cos (3v + 2\omega) \right) \right]
\end{align*}
\]

(2)
where

\[ S_1 = e/\left[ 1 + (1 - e^2)^{1/2} \right] \]
\[ S_2 = S_1/(1 - S_1 \epsilon) \]
\[ S_3 = 3S_2 + 3S_2^2 \epsilon + S_2^3 \epsilon^2 \]
\[ S_4 = \cos E/(1 - \epsilon \cos E) \]
\[ S_5 = 3S_4 + 3S_4^2 \epsilon + S_4^3 \epsilon^2 \]
\[ S_6 = \frac{S_5 + (1 + S_5 \epsilon) \epsilon^3 - 2\epsilon}{1 - \epsilon^2} \]

\[ d\Omega = -\frac{A_2}{p^2} \cos i \left\{ (v - M) - \frac{1}{2} \sin 2(v + \omega) + \epsilon \sin v \right. \\
- \frac{\epsilon}{2} \sin (v + 2\omega) - \frac{\epsilon}{6} \sin (3v + 2\omega) \right\} \]

\[ d\omega = \frac{A_2}{2p} \left[ A_\omega (\epsilon^{-1}) + B_\omega (\epsilon^0) + C_\omega (\epsilon^1) \right] \]

where

\[ A_\omega (\epsilon^{-1}) = \frac{1}{12\epsilon} \left[ 12 \sin v + \sin^2 i \right] \left[ 7 \sin (3v + 2\omega) - 3 \sin (v + 2\omega) - 18 \sin v \right] \]
\[ B_\omega (e^0) = \frac{1}{8} \left[ 4 \sin 2\nu + 16(\nu - M) - 4 \sin 2(\nu + \omega) \\
+ \sin^2 i \left[ 3 \sin(4 \nu + 2\omega) - 6 \sin 2\nu \\
+ 10 \sin 2(\nu + \omega) - 20(\nu - M) \right] \right] \]

\[ C_\omega (e^1) = \frac{e}{48} \left[ 4 \sin 3\nu + 84 \sin \nu - 24 \sin (\nu + \omega) \\
- 8 \sin (3\nu + 2\omega) + \sin^2 i \left[ 3 \sin (5\nu + 2\omega) \\
+ 3 \sin (\nu - 2\omega) - 6 \sin 3\nu \\
+ 19 \sin (3\nu + 2\omega) + 45 \sin (\nu + 2\omega) \\
- 102 \sin \nu \right] \right] \]

\[ dM = \frac{A_2}{p^2} \sqrt{1 - e^2} \left[ A_M (e^{-1}) + B_M (e^0) + C_M (e^1) \right] \quad (6) \]

where

\[ A_M (e^{-1}) = -\frac{1}{12e} \left[ 12 \sin \nu + \sin^2 i \left[ 7 \sin (3\nu + 2\omega) \\
- 3 \sin (\nu + 2\omega) - 18 \sin \nu \right] \right] \]

\[ B_M (e^0) = -\frac{1}{8} \left[ 4 \sin 2\nu + \sin^2 i \left[ 3 \sin (4\nu + 2\omega) \\
- 6 \sin 2\nu \right] \right] \]

\[ C_M (e^1) = -\frac{e}{48} \left[ 4 \sin 3\nu - 12 \sin \nu \\
+ \sin^2 i \left[ 3 \sin (5\nu + 2\omega) + 3 \sin (\nu - 2\omega) \\
- 6 \sin 3\nu - \sin (3\nu + 2\omega) \\
- 15 \sin (\nu + 2\omega) + 18 \sin \nu \right] \right] \]
SUBROUTINE IDENTIFICATION

A. Title
   DPRLM

B. Segment
   NRTPOD - Input Processor

C. Called by subroutine
   INPUT

FUNCTION

To set up preliminary information for the input processor link. This information concerns epoch time and mode of epoch position and velocity.

USAGE

A. Calling sequence

   Call DPRLM

B. Input

1. COMMON

   CDEG
   CWE
   STVEC
   DTYPE
   DAYINT
   DAYFRC
   TNULL
   SMELM
   CWE
   TEPOCH

   Degrees/radian
   Earth's rotational rate (radians/min)
   Input initial conditions
   Initial conditions type
   Integer portion of Julian date
   Fractional portion of Julian date
   Time to which input elements are to be updated
   21-word vector containing the Smithsonian mean elements and their time derivatives; see Table I in MTOC Subroutine.
   Earth's rotational rate
   Time of epoch, minutes from 0 hours
2. Calling sequence

C. Output

1. COMMON
   TALFAG \( \alpha_g \) for midnight day of epoch
   TEPOCH Epoch time, minutes from midnight
   TNOMX Initial Cartesian coordinates
   TNOMP Initial spherical coordinates

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
   TINIT Sets up initial time, computes \( \alpha_g \) and DBASE
            (days from 1950 to day of epoch)
   PIMOD Takes principle value of angle between 0 and 2\pi
   PTOC Converts from polar coordinates to Cartesian coordinates
   CTOP Converts Cartesian coordinates to polar coordinates
   SETSTR Sets up drag, radiation pressure, potential, parameters
   IPRNT Prints header page
   MTOC Converts mean elements to Cartesian coordinates
SUBROUTINE IDENTIFICATION

A. Title
DPROS

B. Segment
NRTPOD

C. Called by subroutine
INPUT

FUNCTION

To issue calls on the sensor and observation loading routines if required by input.

USAGE

A. Calling sequence
CALL DPROS

B. Input
1. COMMON
   PREFLG
   MT
   NLAM
   ALTS
   CLAMDA

2. Calling sequence

C. Output
1. COMMON

2. Calling sequence

D. Error/action messages
Figure 5-4. DPROS Flow Diagram
SUBROUTINE IDENTIFICATION

A. Title
   DRAG

B. Segment
   NRTPOD

C. Called by subroutine
   DAUX

FUNCTION

Function is to compute the perturbative acceleration of a vehicle due to atmosphere drag and to account for these effects in the variational equations.

USAGE

A. Calling sequence
   Call DRAG

B. Input
   1. COMMON
      FLVE  Variational equation control flag
      TV    Earth-fixed velocity of vehicle
      TVA   Magnitude of TV
      CELLIP Constant = ellipticity of the Earth
      TLIST Numerical integration working storage
      TR2   Square of TR
      TR    Magnitude of the vector from the center of the Earth to the vehicle
      CWE   Constant = rotation rate of the earth (radians/minutes) = \( \omega \)
      CDAD2M Drag parameters \( C_{DA} A/2 m \)
      CFTER Feet per earth radius
      TRHOA Density in slugs/ft\(^3\)
      TALT  Altitude of vehicle in feet

C. Output
   1. COMMON
Perturbative acceleration due to drag

Matrix of velocity dependent terms in the evaluation of the variational equations

Matrix of position dependent terms in the evaluation of the variational equation. (The drag effects are added to the contents of this matrix.)

D. Error/action messages

SUBROUTINES USED

A. Library
SQRT

B. Program
JACHIA
OUTER

EQUATIONS

\[ R_e = \sqrt{\frac{1 - \epsilon}{1 - \epsilon(2 - \epsilon) \left( \frac{x^2 + y^2}{R^2} \right)}}^{1/2} \]

Radius of the Earth

Altitude = \( R - R_e \)

\[ \rho_a = \text{density at the given altitude} \]

\[ v_{ax} = \dot{x} + \omega_e y \]

\[ v_{ay} = \dot{y} - \omega_e x \]

Earth-fixed velocity

\[ v_a = \left( v_{ax}^2 + v_{ay}^2 + v_{az}^2 \right)^{1/2} \]

\[ \lambda = \frac{C_d A}{2m} \]

\[ \dot{x}_{\text{drag}} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ax} \]

\[ \dot{y}_{\text{drag}} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ay} \]

\[ \dot{z}_{\text{drag}} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{az} \]
\[
\text{DRAG}
\]

\[
\begin{align*}
\text{PMAT} &= \text{PMAT} - \lambda \rho_a v_a \left[ \begin{array}{ccc}
0 & \omega & 0 \\
-\omega & 0 & 0 \\
0 & 0 & 0 
\end{array} \right]
- \frac{\lambda V_a \rho'}{R} \left[ \begin{array}{ccc}
0 & \omega & 0 \\
-\omega & 0 & 0 \\
0 & 0 & 0 
\end{array} \right] \\
& \quad - \frac{\lambda \rho_a}{V_a} \left[ \begin{array}{ccc}
\frac{v^2}{\text{ax}} & \text{ax}^\prime y & \text{ax}^\prime z \\
\text{ax}^\prime y & \frac{\text{ax}^\prime}{\text{ay}} & \frac{\text{ax}^\prime}{\text{az}} \\
\text{ax}^\prime z & \frac{\text{ax}^\prime}{\text{ay}} & \frac{\text{ax}^\prime}{\text{az}} 
\end{array} \right] \\
& \quad - \frac{\partial \lambda}{\partial h} \left[ \begin{array}{ccc}
\frac{\rho_a v_a}{R} & \frac{\text{ax}^\prime y}{\text{ay}} & \frac{\text{ax}^\prime z}{\text{az}} \\
\frac{\text{ax}^\prime y}{\text{ay}} & \frac{\text{ax}^\prime}{\text{az}} & \frac{\text{ax}^\prime}{\text{az}} \\
\frac{\text{ax}^\prime z}{\text{az}} & \frac{\text{ax}^\prime}{\text{az}} & \frac{\text{ax}^\prime}{\text{az}} 
\end{array} \right] \\
\text{VMAT} &= \text{VMAT} - \frac{\lambda \rho_a}{V_a} \left[ \begin{array}{ccc}
\frac{v^2}{\text{ax}} & \text{ax}^\prime y & \text{ax}^\prime z \\
\text{ax}^\prime y & \frac{\text{ax}^\prime}{\text{ay}} & \frac{\text{ax}^\prime}{\text{az}} \\
\text{ax}^\prime z & \frac{\text{ax}^\prime}{\text{ay}} & \frac{\text{ax}^\prime}{\text{az}} 
\end{array} \right] \\
& \quad - \frac{\lambda \rho_a v_a}{1} \left[ \begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 
\end{array} \right]
\end{align*}
\]
When multiple $C_D A/m's$ are included in the solution vector, DRAG computes the inhomogeneous terms in the variational equations of drag. These equations are derived in Reference 3. The inhomogeneous term is expressed as:

$$\frac{\partial \lambda}{\partial \lambda_i} \rho \vec{v} \vec{a}$$

Written out in full, this term has the form

$$\frac{\partial \lambda}{\partial \lambda_i} \rho \vec{v} \vec{a} = E_1(h) \rho \vec{v} \vec{a}$$

where

$$E_1(h) = \begin{cases} 
0 & h < h_{i-1} \\
\frac{h - h_{i-1}}{h_i - h_{i-1}} & h_{i-1} < h < h_i \\
\frac{h_{i+1} - h}{h_{i+1} - h_i} & h_i < h < h_{i+1} \\
0 & h_{i+1} < h 
\end{cases}$$
SUBROUTINE IDENTIFICATION

A. Title
   DRDP

B. Segment
   NRTPOD

C. Called by subroutine
   RADR

FUNCTION

Function is to compute the partial of the $M^{th}$ type of observation with respect to the solution vector.

USAGE

A. Calling sequence
   Call DRDP (M)

B. Input
   1. COMMON
      
      NAROW    Starting location where one row of the augmented matrix (A, B) is stored
      NDPR     Number of all differential plus initial parameters to solve for (Category 1)
      PUBS     Current observation buffer; ID, time, R or $\hat{R}$, A, E, $\hat{R}$, type
      PCSA     Cos A
      PRSUBI   $R_1 = VR$
      PSNA     Sin A
      PSNE     Sin E
      PSTAT    Working storage for sensor information
      PUDTI    $\bar{u} = (\dot{u}_1, \dot{u}_2, \dot{u}_3)$
      PUI      $(u_1, u_2, u_3)$
      PWDTPP   $\partial \bar{w}/\partial P_i$

5-53
2w/8P. 3w/3P.  

\[ \bar{u} = (\bar{u}_1, \bar{u}_2, \bar{u}_3) \]

2. Calling sequence

M               Observation type number \( (1, 2, 3, 4, ) \)
1               R or \( \bar{R} \) (range or range acceleration)
2               Azimuth
3               Elevation
4               \( \dot{R} \) (range rate)

C. Output

1. COMMON

\[ \text{VSTR(NAROW)} \Rightarrow \text{VSTR(NAROW + NDPR - 1)} \] contains the partial derivatives of the \( M^{th} \) type observation with respect to the Category 1 variables being solved for

2. Calling sequence

SUBROUTINES USED

A. Library

B. Program

EQUATIONS

Range (type 0 observation, \( M = 1 \))

\[
\frac{\partial R}{\partial p_i} = u_1 \frac{\partial w_1}{\partial p_i} + u_2 \frac{\partial w_2}{\partial p_i} + u_3 \frac{\partial w_3}{\partial p_i} \quad p_i \Rightarrow a, \delta, \beta, A, r, v, \lambda_1, \lambda_2 \ldots \lambda_n
\]

Azimuth (type 0 observation, \( M = 2 \))

\[
\frac{\partial A}{\partial p_i} = \frac{1}{R} \left[ \frac{\partial w_2}{\partial p_i} \cos A - \left( \frac{\partial w_1}{\partial p_i} \sin \phi^y + \frac{\partial w_3}{\partial p_i} \cos \phi^y \right) \sin A \right]
\]

5-54
Elevation (type 0 observation, \( M = 3 \))

\[
\frac{\partial E}{\partial p_i} = \frac{1}{R_1} \left( \frac{\partial w_1}{\partial p_i} \cos \phi^* + \frac{\partial w_3}{\partial p_i} \sin \phi^* - \frac{\partial R}{\partial p_i} \sin E \right)
\]

Range Rate (type 0 observation, \( M = 4 \))

\[
\frac{\partial R}{\partial p_i} = \left( \frac{\partial w}{\partial p_i} \cdot \frac{\dot{u}}{u} \right) + \left( \frac{\dot{u}}{u} \cdot \frac{\partial w}{\partial p_i} \right)
\]

Range Acceleration (type 1 observation, \( M = 1 \))

\[
\frac{\partial \ddot{R}}{\partial p_i} = \frac{\partial}{\partial t} \left( \frac{\partial R}{\partial p_i} \right) = \ddot{\frac{\dot{u}}{u}} \cdot \frac{\partial w}{\partial p_i} + 2 \ddot{\frac{\dot{u}}{u}} \cdot \frac{\partial w}{\partial p_i} + \dddot{u} \cdot \frac{\partial w}{\partial p_i}
\]
SUBROUTINE IDENTIFICATION

A. Title
   ELEM

B. Segment
   NRTPOD

C. Called by subroutine
   NXN

FUNCTION

To access an element of a lower triangular matrix stored by rows.

USAGE

A. Calling sequence

\[ S_{ij} = \text{RLRM}(S, I, J) \]

B. Input

1. Calling sequence
   
   \[ S \quad \text{Location of } S \]
   
   \[ I \quad \text{Row number} \]
   
   \[ J \quad \text{Column number} \]

C. Output

1. COMMON
   
   \[ \text{ELEM} \quad \text{Element } S_{ij} \]

D. Error/action messages

SUBROUTINES USED

A. Library
B. Program

EQUATIONS

\[ k = \frac{i(i - 1)}{2} + j \]

\[ \text{ELEM} = S(k) \]
SUBROUTINE IDENTIFICATION

A. Title
   FALSI

B. Segment
   NRTPOD

C. Called by subroutine
   TRJGEN

FUNCTION

Subroutine FALSI utilizes the method of "false position" (regula falsi), in determining altitude cutoffs as a function of time.

USAGE

A. Calling sequence
   Call FALSI

B. Input

1. COMMON

   TG          Integration time to go ... minutes from 0 hrs
               day of epoch

   TLIST       Integration list (See TRAJ subroutine.)

   TRAJX       Integration coordinates - referenced to time,
               TG

   TCRASH      Impact flag - non-zero if earth impact has
               occurred

   TEPOCH      Time of epoch, minutes from 0 hrs. day of
               epoch

   CHEPS       Tolerance criterion of altitude cutoffs (earth
               radii)

   TMINUS      Flag indicating the direction of integration to
               subroutine SELECT

2. Calling sequence
C. Output

1. COMMON

ALT  Two altitude layers bounding the current region of influence of the drag coefficients (CD/m)

INFG   Flag indicating to subroutine PLAMDA whether an altitude crossing has occurred and which region of drag influence has been entered

INFG = 0 no altitude crossing occurred

   = 1 vehicle has reentered altitude region 1

   = 2 vehicle has left influence of altitude region 1 and crossed into altitude region 2

IFVE(2) Of the two inhomogeneous variational systems inside any one altitude division being integrated, IFVE cells flag which of the drag coefficients in the current region of influence are being solved for

IFVE(I) = 0 Ith inhomogeneous variational system is not being solved for

IFVE(I) ≠ 0 Ith inhomogeneous variational system is being solved for

where

   I = 1, 2

2. Calling sequence

---

D. Error/action messages

---

SUBROUTINES USED

A. Library

SIGN     Sign of arg 2 times |arg 1|

ABS      Absolute value
B. Program

DAUX
Driver controlling routines which compute the accelerations in the equations of motion and accelerations in the variational equations

FVE
Determines the flags which indicate the C_D/m's to be solved in the current altitude region of influence

HEIGHT
Computes altitude of vehicle above the reference ellipsoid

HINT
Computes the current two altitude layers that the vehicle lies between during the trajectory simulation

PLAMDA
Computes the partials of x, y, z, \(\dot{x}, \dot{y}, \dot{z}\), \(\ddot{x}, \ddot{y}, \ddot{z}\) with respect to each C_D/m to be solved for

TRAJ
Integrates the equations of motion and up to 24 variational equations to a specified time

EQUATIONS

In the method of "false position" (regula falsi), the iteration is initiated by finding \(z_0\) and \(z_1\) such that \(f_0\) and \(f_1\) are of opposite signs and by defining the slope of the secant \(P_0P_1\), so that

\[
z_2 = z_1 - \left(\frac{z_1 - z_0}{f_1 - f_0}\right)f_1 = \frac{f_1z_0 - f_0z_1}{f_1 - f_0}
\]

In each following iteration, the slope is taken as the slope of the line joining \(P_k\) and the most recently determined point at which the ordinate differs in sign from that at \(P_k\). See figure 5-5.
Figure 5-5. Iteration Techniques Used in Subroutine FALSI
Figure 5-6. FALSI Flow Diagram
Figure 5-6. FALSI Flow Diagram (Continued)
SUBROUTINE IDENTIFICATION

A. Title
FIT

B. Segment
NRTPOD

C. Called by subroutine
DCITER

FUNCTION
This subroutine monitors the flow of information through the following sequence of events.

a) Determines whether the current iteration is converging or diverging
b) Forming the solution vector of the differential correction and applying it to give new estimates of the parameters being solved for
c) Test for maximum iterations
d) Set the bounds for the next iteration
e) Test whether 4 solutions in a row have failed to converge

USAGE

A. Calling sequence
CALL FIT

B. Input

1. COMMON

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFTEPS</td>
<td>Convergence criterion, (nominally set to 1.0 E-3).</td>
</tr>
<tr>
<td>KOUT</td>
<td>Symbolic output tape (print).</td>
</tr>
<tr>
<td>NDPAR1</td>
<td>Identifier showing the starting location of where the solution vector will be stored in variable storage.</td>
</tr>
<tr>
<td>NPR</td>
<td>Total number of parameters to solve for.</td>
</tr>
<tr>
<td>NITER</td>
<td>Maximum allowable iterations.</td>
</tr>
<tr>
<td>IFIT</td>
<td>Identifies predicted RMS's if bounds are used in forming solutions.</td>
</tr>
<tr>
<td>CFLAG</td>
<td>Suppresses RMS test when impact has occurred.</td>
</tr>
</tbody>
</table>

5-65
FIT

NITCT  Counter on number of iterations.
TSUSB  Best RMS so far.
TSUSP  Predicted RMS for next iteration
TZ     Flag to indicate if the solution was
       affected by the bounds. If the flag is
       non-zero the solution was affected by
       the bounds.
XBSQ   Scale factor for BNDS to cause sub-
       sequent solutions to be affected by
       bounds.
TCRASH Flag to indicate impact, TCRASH
        ≠ 0, indicates impact has occurred.
IFTEX  Indicates mode of exit from FIT.
POBCNT Number of observables actually included
        (after editing, etc.) on any iteration.
TSUS   Current RMS.
MPR    Size of constrained system.
MBNDS  Variable storage pointer for bounds
        corresponding to constrained system.
NST    Variable storage pointer for temporary
        storage used for linear constraints.

2. Calling sequence

- 

C. Output

1. COMMON

VSTR (NBDNS) Array in variable storage containing the
            set of bounds to be used on the next
            iteration.

2. Calling sequence

- 

D. Error/action messages

"*************MAJOR PROGRAM ERROR,...POSSIBLE
          INPUT AND/OR MACHINE ERROR"

This message is printed if FIT is less than or equal to zero.

SUBROUTINES USED

A. Library

SQRT
ABS

5-66
B. Program

BOUND

Least square package, solves $Ax = B$

LEGS2

Applies differential correction solution vector and prints the iteration summary.

APPLY

Scale bounds with a given scale factor
FIT

TEST CURRENT SOS WITH BEST PREVIOUS SOS

CONVERGING?

SET NEW BEST SOS

IS CURRENT SOS MORE THAN 10% LARGER THAN PREVIOUS SOS?

CALL BOUND

CALL LEGSS2

MAXIT (O):

CALL NCKN

COMPUTE NEW PREDICTED SOS

CALL APPLY

CONVERGENCE TEST NETS

MAXITATIONS?

CALL BOUND

MAKE 3 ADDITIONAL SO LUTIONS

RETURN

CALL APPLY

CALL BOUND WITH NEW SCALE

RETURN

HAVE ALL 4 SOLUTIONS BEEN TRIED FOR CONVERGENCE?

Figure 5-7. FIT Flow Diagram

5-68
SUBROUTINE IDENTIFICATION

A. Title
FSIGMA

B. Segment
NRTPOD

C. Called by subroutine
RADR

FUNCTION

Computes the specified functional standard deviation as a function of the nominal standard deviation, the topocentric range, the radar cross section, and other sensor dependent constants.

USAGE

A. Calling sequence
   Call FSIGMA (I)

B. Input

1. COMMON
   KOUT       Output tape number
   CMTER      Meters/e. r.
   CDEG       Degrees/radian
   CKMER      Kilometers/e. r.
   NSTAT      Starting location in VSTR of the master sensor table
   NSSTB      Starting location in VSTR of sensor mean and RMS table
   NMSTAT     Number of cells in VSTR allotted per sensor
   PUBS       Observation vector ID, time, R, A, E, \( \hat{R} \), type or ID, time, \( \hat{R} \), 0, 0, 0, type
   PCMR       Computed slant range of spacecraft relative to the current station in PSTAT
   PUI        The topocentric direction cosines of vehicle position in the equatorial system
PWDTI  Geocentric earth fixed velocity of the vehicle in a station meridian equatorial system.

2. Calling sequence

I  Flag denoting observation tape
I = 1  denotes range or range acceleration observation type
I = 2  denotes azimuth observation type
I = 3  denotes elevation observation type
I = 4  denotes range rate observation type

C. Output

1. COMMON
   PSIG - Functional standard deviation

2. Calling sequence

D. Error/action messages

If the station ID is not found in the master sensor table, an error comment "ROUTINE FSIGMA CANNOT FIND STATION XXX, INPUT ERROR" is printed off line and the program halts.

SUBROUTINES USED

A. Library
   ACOS
   EXIT
   SQRT

B. Program
   DOT  Dot product routine
   RADSQ  Magnitude of a vector
   FTHET  Computes \( f(\theta) \) which is the radar cross section.
The specified functional standard deviation, $\sigma_f$, is as follows:

$$\sigma_f = \sqrt{A_i^2 + B_i R^4/f(\theta)}$$

$A_i =$ Nominal standard deviation

$B_i =$ Sensor dependent constant ($i = 1, \ldots, 5$) denoting $A, E, R, R, \bar{R}$

$R =$ Topocentric range

$f(\theta) =$ Radar cross section

The function $f(\theta)$ is obtained by linear interpolation from the input table $[\theta_j, f(\theta)_j]$, $j = 1, \ldots, n$ where $n \leq 7$, each table being sensor dependent. The argument $\theta$ is the angle between the drag velocity vector and the topocentric range vector. This angle is sensor dependent and its formulation is derived in Reference 4.
SUBROUTINE IDENTIFICATION

A. Title
FTHET

B. Segment
NRTPOD

C. Called by subroutine
FSIGMA

FUNCTION

FTHET computes \( f(\theta) \), the radar cross section, by linearly interpolating a sensor specific input table \( \{ \theta_j, f(\theta)_j \} \) \( j = 1, \ldots, n \), where \( n \leq 7 \). The angle between the drag velocity vector and the topocentric range vector, \( \theta \), is given as an argument to FTHET.

USAGE

A. Calling sequence

\[ F = \text{FTHET}(J, \text{THETA}) \] (FUNCTION SUBPROGRAM)

B. Input

1. COMMON

VSTR - Variable storage

2. Calling sequence

THETA \hspace{1cm} \text{angle between the drag velocity vector and the topocentric range vector. Using THETA as an argument, } f(\theta) \text{ is obtained by linear interpolation from the input table } \{ \theta_j, f(\theta)_j \} \hspace{1cm} \text{\( j = 1, \ldots, n \); where } n \leq 7

J \hspace{1cm} \text{Integer identifying the position in the master sensor table of the current station ID}

C. Output

1. COMMON

5-73
2. Calling sequence (FUNCTION SUBPROGRAM)

F - Interpolated value of the radar cross section, \( f(\theta) \)

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

EQUATIONS

Given the argument \( \theta \) where \( \theta_j \leq \theta \leq \theta_{j+1} \), the corresponding value of \( f(\theta) \) is obtained by linearly interpolating between \( \theta_j, f(\theta_j) \) and \( \theta_{j+1}, f(\theta_{j+1}) \).

The function, \( f(\theta) \), is assumed constant between 0 and \( \theta_1 \), and between \( \theta_n \) and \( \pi \). In other words,

\[
\text{if } 0 \leq \theta \leq \theta_1, \quad f(\theta) = f(\theta_1)
\]

\[
\text{if } \theta_n \leq \theta \leq \pi, \quad f(\theta) = f(\theta_n)
\]

The two following equations will apply when there is either one entry or none in the \([\theta, f(\theta)]\) table.

\[
\text{if } n = 1, \quad f(\theta) = f(\theta_1) \quad 0 \leq \theta \leq \pi
\]

\[
\text{if } n = 0, \quad f(\theta) = 1 \quad 0 \leq \theta \leq \pi
\]
SUBROUTINE IDENTIFICATION

A. Title
   FVE

B. Segment
   NRTPOD

C. Called by subroutines
   FALSI
   SETIC

FUNCTION

Determines the flags which indicate the $C_D$m's to be solved while in a particular altitude region.

USAGE

A. Calling sequence
   Call FVE

B. Input
   1. COMMON
      
      VSTR         Variable storage
      NLAMS        Number of $C_D$m's in the solution vector
      NH           Pointer to location in variable storage where the altitude - $C_D$m table is stored
      NLID         Pointer to location in variable storage where the identifiers for the $C_D$m's appearing in the solution vector are stored
      NICPR        Number of ADBARV variables in the solution vector
      NH1          Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence
      NH2          Pointer to location in variable storage of 2nd altitude layer bounding the current region of influence

2. Calling sequence
C. Output

1. COMMON

   IFVE(1) Flag indicating whether the 1st $C_D A/m$ of a particular region is in the solution vector

   IFVE(2) Flag indicating whether the 2nd $C_D A/m$ of a particular region is in the solution vector

   IFVE = 0 the $C_D A/m$ of a particular region is not in the solution vector.

   ≠ 0 the $C_D A/m$ of a particular region is in the solution vector.

   NDPRT Number of CAT1 variables plus number of $(C_D A/m)$ drag parameters being integrated at any one time (either 6 or 8).

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

5-76
SUBROUTINE IDENTIFICATION

A. Title
   HEIGHT

B. Segment
   NRTPOD

C. Called by subroutines
   DRAG
   FALSI

FUNCTION

Computes the altitude of a reference vehicle above the earth where the earth equatorial radius is computed as a function of the geocentric latitude.

USAGE

A. Calling sequence

   \[ H = \text{HEIGHT}(X) \]  \text{(FUNCTION SUBPROGRAM)}

B. Input

1. COMMON
   \[ \text{TRM1} \quad \text{See CONST subroutine} \]
   \[ \text{TRM2} \quad \text{See CONST subroutine} \]

2. Calling sequence

   \[ X \quad \text{- Beginning address of the position vector (earth radii)} \]

C. Output

1. COMMON

2. Calling sequence \text{(FUNCTION SUBPROGRAM)}

   \[ H \quad \text{- Altitude above earth (earth radii)} \]

D. Error/action messages

---

5-77
SUBROUTINES USED

A. Library
   SQRT
   EXP

B. Program

EQUATIONS

The altitude (h) above the earth is computed as follows:

\[ h = r - \rho \]

where \( r \) is the radius magnitude of the vehicle and \( \rho \) is the radius of the earth at sea level.

\[ \rho \approx a_e \left[ 1 - \left( \frac{k}{2} - \frac{45}{256} k^3 \right) s^2 + \left( \frac{3}{8} k^2 - \frac{15}{32} k^3 \right) s^4 \right] \]

where

\[ s = \sin \phi \quad \phi = \text{geocentric latitude} \]
\[ k = \frac{2 - \epsilon}{(1 - \epsilon)^{2/3}} \quad \epsilon \text{ is the ellipticity of the earth} \]
\[ a_e = \text{the earth equatorial radius} \]
SUBROUTINE IDENTIFICATION

A. Title
HINT

B. Segment
NRTPOD

C. Called by subroutines
SETIC
FALSI

FUNCTION

Computes the current two altitude layers that the vehicle lies between during the trajectory simulation.

USAGE

A. Calling sequence
Call HINT (H, ALT)

B. Input

1. COMMON

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSTR</td>
<td>Variable storage</td>
</tr>
<tr>
<td>NH</td>
<td>Pointer to location in variable storage where the altitude - $C_D A/m$ table is stored</td>
</tr>
<tr>
<td>NLAM</td>
<td>Total number of entries in the altitude ($C_D A/m$) table</td>
</tr>
<tr>
<td>NPXLM</td>
<td>Pointer to location in variable storage where the $\frac{\partial (C_D A)}{m}$ matrix is stored</td>
</tr>
</tbody>
</table>

$$i = 1, \ldots, NLAM$$

5-79
NPALM Pointer to location in variable storage where the \(b_i\)'s are stored.

\[
b_i = \left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial(v, \dot{v}, \beta, A, r, v)}_{t=0} \right]^{-1} \left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial \left( \frac{C_D A}{m} \right)}_{i} \right]
\]

6 x 6 6 x 1

\(i = 1, \ldots, NLAM\)

KOUT Output tape number

INFG Flag indicating whether an altitude crossing has occurred and which region of drag influence has been entered

\(\text{INFG} = 0\) no altitude crossing occurred

\(= 1\) vehicle has reentered altitude region 1

\(= 2\) vehicle has left influence of altitude region 1 and crossed into altitude region 2

2. Calling sequence

\(H\) - current altitude of vehicle (earth radii)

C. Output

1. COMMON

\(\text{NH1}\) Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence

\(\text{NH2}\) Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

\(\text{ALT}(I)\) Two altitude layers bounding the current region of influence \(i = 1, 2\)

5-80
D. Error/action messages

If the current altitude is not bounded by the input altitudes the following error message appears off-line: "ALTITUDE = XXXXXX NOT BOUNDED BY INPUT ALTITUDES" and the program halts.

SUBROUTINES USED

A. Library
   EXIT

B. Program
HINT

NO ALTITUDE CROSSING
SEARCH TABLE OF ALTITUDES
FOR BOUNDING ALTITUDE LAYERS

ALTITUDE LAYER 1 HAS BEEN REENTERED.
COMPUTE NEW ALTITUDE LAYERS BASED ON REENTRY STATUS OF VEHICLE
ALT (1) = ALT (1)
ALT (2) = NEW ALTITUDE

ALTITUDE LAYER 2 HAS BEEN ENTERED.
COMPUTE NEW ALTITUDE LAYERS
ALT (1) = ALT (2)
ALT (2) = NEW ALTITUDE

RETURN

Figure 5-8. HINT Flow Diagram
SUBROUTINE IDENTIFICATION

A. Title
   INPUT

B. Segment
   NRTPOD - Input processor

C. Called by subroutines
   NRTPOD (DRIVER)

FUNCTION

INPUT's function is to serve as a main driver for the Input Processor Link. It utilizes routines to initialize COMMON storage, process NAME- LIST input, assign variable storage, and process sensor information and observations.

USAGE

A. Calling sequence
   CALL INPUT

B. Input
   1. //COMMON
      -
   2. Calling sequence
      -

C. Output
   1. //COMMON
      -
   2. Calling sequence
      -

D. Error/action messages
   -
E. Internal storage

1. //COMMON

   NDAYS  Number of days of ephemeris data (positions of the moon and sun) accepted on input
   DVEHN  Array of 3 BCD words identifying the vehicle number and name
           (Input to columns 4-17 on JDC card)
   DHEAD  2 BCD words containing arbitrary header information.  (Input to columns 18-29 on JDC card)
   PREFLG NRTPOD control flags
           (columns 31-40 on JDC card)
   DCFLG  NRTPOD control flags
           (columns 41-50 on JDC card)
   PSTFLG NRTPOD control flags
           (columns 51-60 on JDC card)
   KIN    Symbolic input tape number
   KOUT   Symbolic output tape number
   COMLST Contains size of variable storage

2. Labeled COMMON

   /VSTR/  Variable storage array
   VSTR
   /INPP/  Temporary cells containing sensor information used by the Input Processor Link
   DTMP
   DATA   Temporary cells used only by the Input Processor Link
   /EPHCOM/  Array of storage containing the position ephemerides of the moon and sun (Input to NRTPOD)
   ECOM

SUBROUTINES USED

A. Library

B. Program

   SETCON  Sets up program constants.
   RDDATA  Routine to read NAMELIST input and ephemeris data.
   ASSIGN  Establishes storage assignments for VSTR (variable storage) arrays.
Sets up VSTR (NIDP), VSTR (NPRCD), VSTR (NBPIS), VSTR (NSCALE), VSTR (NBDNS), and DTMP tables.

Moves observation deletion numbers from DATA storage to VSTR (NIDLED).

Convert the upper triangular S matrix in DATA storage from human units to machine units and then transfer to VSTR (NATA).

Move the initial update matrix from DATA storage to VSTR (NR) and convert from human units to machine units.

Sets up preliminary information for the input processor. This information concerns epoch time and mode of epoch position and velocity.

Precess ephemeris data from mean equator and equinox of 1950.0 to the equator and true equinox of date.

Issue calls on the sensor and observation loading routines if required.

Prints program constants, input data, variable storage pointers, and working storage cells.

Processes linear constraint data.
Figure 5-9. Input Flow Diagram
Figure 5-9. Input Flow Diagram (Continued)

5-87
SUBROUTINE IDENTIFICATION

A. Title
   JDCPRT

B. Segment
   NRTPOD

C. Called by subroutine
   INPUT

FUNCTION

Prints and describes the Job Description Card options which have been requested on a particular NRTPOD case.

USAGE

A. Calling sequence
   Call JDCPRT

B. Input
   1. COMMON
      PREFLG   Columns 31-40 of the JDC
      DCFLG    Columns 41-50 of the JDC
      PSTFLG   Columns 51-60 of the JDC
      KOUT     Output tape number
   2. Calling sequence

C. Output
   1. COMMON

2. Calling sequence

3. Other

   Printed description of the options requested on the JDC
D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
LEGGS2

B. Segment
NRTPOD

C. Called by subroutine
FIT
PRAUPD

FUNCTIONS

a) To solve an overdetermined linear system of equations $Ax = b$

b) To compute the inverse of $A^T A$

c) After solving for $x$, to compute $\|Ax - b\|^2$

USAGE

A. Calling sequence
Call LEGGS2 (NDPAR, Z, SUSP, I1, I2, I4)

B. Input

1. COMMON

   NATA Identifies the starting location of where the upper triangular $A^T A$ is stored

   NBDNS Identifies the starting location for the bounds used by LEGGS2

   NPR Number of all parameters to solve for

   NR Identifies the starting location of where the inverse $A^T A$ (in triangular form) is stored

2. Calling sequence

   NPAR The index for variable storage where the solution vector $x$ is to be stored

   I1
   I2
   I4 Option control flags

   NEQ Size of system

   NBD Pointer in VSTR for system bounds
C. Output

1. COMMON
   VSTR (NDPAR)  Start of the array containing the solution vector \( x \)
   VSTR (NR)    Start of an array containing \((A^T A)^{-1}\) as a lower triangular matrix

2. Calling sequence
   Z
   Flag to indicate if the solution was affected by the bounds. If the flag is non-zero the solution was affected by the bounds
   B
   Predicted SOS for the next iteration

SUBROUTINES USED

A. Library

B. Programs

EQUATIONS

To solve for differential corrections, find \( x \) so that \( \|Ax - b\| ^2 \) is minimum under the side condition that

\[
\sum_{i} \left( \frac{x_i}{B_i} \right)^2 \leq 1 \quad \text{for} \quad B_1', B_2', \ldots, = \text{bounds}
\]

The side condition may be described as

\[
x^T B^{-2} x \leq 1
\]

where

\[
\begin{bmatrix}
B_1^{-2} & 0 & \cdots \\
0 & B_2^{-2} & \cdots \\
\vdots & \vdots & \ddots
\end{bmatrix} = B^{-2}
\]

\( B^{-2} \) is a diagonal matrix
Bounds

Define \( x(z) \) as the solution of the linear system

\[
(A^T A + zB^{-2}) \cdot X = A^T b
\]

where \( B^{-1} \) is the diagonal matrix with the \((i, i)\) diagonal element being \( B_i^{-1} \) if \( B_i > 0 \) and \( B_i < 0 \). If \( B_i = 0 \), the \( i^{th} \) row and column of the augmented normal matrix is ignored and \( x_i \) is set to zero.

a) The routine finds \( x = x(0) \). If \( (B^{-2}, x, x) \leq 1 + \epsilon \), the solution is obtained. Otherwise

b) Define \( y(z) = [B^{-2}x(z), x(z)] \). Now \( y(0) > 1 + \epsilon \). Compare \( y(h), y(100h), y(1000h), \ldots \), until a value of \( z \) is found with

\[ 1 - \epsilon_2 \leq y(z) \leq 1 + \epsilon_1, \]

in which case \( x(z) \) is the solution or until two values of \( z \) are found with \( y(z_1) > 1 + \epsilon_1 \) and \( y(z_2) < 1 - \epsilon_2 \). The required value of \( z \) is now bracketed. Then

c) Choose a value \( z_3 \) between \( z_1 \) and \( z_2 \). If \( 1 - \epsilon_2 \leq y(z_3) \leq 1 + \epsilon_2 \), then \( y(z_3) \) is the solution. Otherwise

d) Use inverse quadratic interpolation (to zero) to obtain a new guess \( z_4 \). If \( 1 - \epsilon_2 \leq y(z_4) \leq 1 + \epsilon_1 \), then \( x(z_4) \) is the solution. Otherwise

e) Select from the set \( z_1, z_2, z_3, z_4 \) the two values of \( z \) which bracket the solution most tightly. Use these values as \( z_1 \) and \( z_2 \) and go back to 3.

The iterative process will stop if the number of solutions of the linear system reaches 20.

Linear System

Let \( C = A^T A + zB^{-2} \). The routine finds a matrix \( S \) with \( SCST = D \). \( S \) is lower triangular with (-1) on the diagonal. It is easy to find \( S \) and \( D \) for a 1 x 1 matrix \( C \). Assume \( S \) and \( D \) have been found for a \( k \times k \) matrix \( C \). Now augment \( C \) by another row and column

\[
\begin{pmatrix}
C & d \\
(d^T & a)
\end{pmatrix}
\]
A vector \( \omega \) and a scalar \( \beta \) are now desired such that

\[
\begin{pmatrix}
S & 0 \\
\omega^T & -1
\end{pmatrix}
\begin{pmatrix}
C & d \\
d^T & a
\end{pmatrix}
\begin{pmatrix}
S^T \\
0 & -1
\end{pmatrix} =
\begin{pmatrix}
D & 0 \\
0 & \beta
\end{pmatrix}
\]

The requirements are satisfied by

\[
\omega = S^T D^{-1} S \delta
\]

\[
\beta = a - \omega^T d
\]

The routine builds the matrix \( S \) by the above process with \( k = 2, 3, \ldots, N \).

The final result is a decomposition of the augmented matrix

\[
\begin{pmatrix}
S & 0 \\
\omega^T & -1
\end{pmatrix}
\begin{pmatrix}
A^T A + z B^{-2} & A^T b \\
b^T A & b^T b
\end{pmatrix}
\begin{pmatrix}
S^T \\
0 & -1
\end{pmatrix} =
\begin{pmatrix}
D & 0 \\
0 & a
\end{pmatrix}
\]

and the \( N \)-dimensional vector \( \omega \) which appears above is the solution vector.

\textbf{Predicted RMS for Next Iteration}

Given \( b^T b, A^T A, A^T b, X, n \) = total number of observations

\[
\text{Predicted RMS} = \frac{1}{\sqrt{n}} \sqrt{b^T b - 2 x^T (A^T b) + x^T (A^T A x)}
\]
SUBROUTINE IDENTIFICATION

A. Title
LESK

B. Segment
NRTPOD

C. Called by subroutine
PLAMDA

FUNCTION

This subroutine solves the matrix equation AX = B. A is an N x N coefficient matrix, B is an N x M matrix, and X is the N x M solution matrix. Gaussian elimination is used with row interchange taking place to position maximum pivot elements after the rows are initially scaled.

USAGE

A. Calling sequence
Call LESK (C, X, S, N1, M1, N1X, DET, LA)

B. Input

1. COMMON

2. Calling sequence

C The augmented matrix [A, B] of maximal dimension N1X x (N1X + M1X)

N1 The actual order of A

M1 The actual number of columns in B

N1X The maximal order of A

DET Both the determinant flag and determinant value if desired

on input DET = 0. No determinant

on output DET = 1. Determinant

on input

5-95
C. Output

1. COMMON

2. Calling sequence

\[ X \] The solution matrix of maximal dimension \( N_1 \times M_1 \)

\[ S \] A vector of dimension \( N_1 \) used to store the row scale factors

\[ LA \] An error return

- \( LA = +1 \) during scaling a zero row was found
- \( LA = 0 \) normal return
- \( LA = -1 \) \( |A(J, J)| < 10^{-10} \) for some \( J \)

D. Error/action messages

---

SUBROUTINES USED

A. Library

DABS Takes absolute value

B. Program

---

5-96
SUBROUTINE IDENTIFICATION

A. Title
   MATMLT

B. Segment
   NRTPOD

C. Called by subroutine
   PLAMDA

FUNCTION

Forms the matrix product \([A][B]\) and stores the result in \([C]\). \([A]\), \([B]\), and \([C]\) are all singly subscripted arrays stored by rows.

USAGE

A. Calling sequence
   Call MATMLT (A, B, C, I, J, K)

B. Input
   1. COMMON
      
   2. Calling sequence
      A - Beginning location of matrix \([A]\)
      B - Beginning location of matrix \([B]\)
      I  Dimensions of matrix \([A]\) is I rows by J
      J  columns. Dimensions of matrix \([B]\) is J rows
      K  by K columns. Dimensions of matrix \([C]\) is I rows
          by K columns.

C. Output
   1. COMMON
      
   2. Calling sequence
      C - Beginning location of where matrix product \([A][B]\) is
          stored
3. Internal Storage

TEMP  Temporary storage used for accumulating the row by row product. TEMP is currently dimensioned (10) in this subroutine which restricts the maximum size of the matrix product to a 10 x 10. This maximum may be modified by merely altering the dimension size of TEMP.

D. Error/action messages

---

SUBROUTINES USED

A. Library

---

B. Program

---

5-98
MTOC

SUBROUTINE IDENTIFICATION

A. Title
MTOC

B. Segment
NRTPOD

C. Called by subroutine
DPRLM

FUNCTION

To update a set of Smithsonian mean elements, convert to osculating and then to Cartesian. It also calls JTOC to convert the Julian date to calendar date.

USAGE

A. Calling sequence
Call MTOC (TNOMX, SMELM, DELT)

B. Input
1. COMMON

   DAYINT    Integer portion of Julian date
   DAYFRC    Fractional portion of Julian date
   CJ2       J2 earth harmonic
   C2PI      $2\pi$ radians
   CPI       $\pi$ radians
   KOUT      Output tape unit
   CMU       $\mu$ ER$^3$/min$^2$
   CKMER     Conversion from kilometers to earth radii

2. Calling sequence

   SMELM     21-word vector containing the Smithsonian mean elements and their time derivatives for updating and conversion to osculating. See Table I.
Table 1

<table>
<thead>
<tr>
<th>Location</th>
<th>Element</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMELM (1)</td>
<td>a</td>
<td>earth radii</td>
</tr>
<tr>
<td>(2)</td>
<td>e</td>
<td>--</td>
</tr>
<tr>
<td>(3)</td>
<td>i</td>
<td>radians</td>
</tr>
<tr>
<td>(4)</td>
<td>Ω</td>
<td>radians</td>
</tr>
<tr>
<td>(5)</td>
<td>ω</td>
<td>radians</td>
</tr>
<tr>
<td>(6)</td>
<td>M</td>
<td>radians</td>
</tr>
<tr>
<td>(7)</td>
<td>\dot{a}</td>
<td>er/day</td>
</tr>
<tr>
<td>(8)</td>
<td>\dot{e}</td>
<td>--/day</td>
</tr>
<tr>
<td>(9)</td>
<td>\dot{i}</td>
<td>rad/day</td>
</tr>
<tr>
<td>(10)</td>
<td>\dot{Ω}</td>
<td>rad/day</td>
</tr>
<tr>
<td>(11)</td>
<td>\dot{ω}</td>
<td>rad/day</td>
</tr>
<tr>
<td>(12)</td>
<td>n</td>
<td>rad/day</td>
</tr>
<tr>
<td>(13)</td>
<td>\dot{a}/2</td>
<td>er/day²</td>
</tr>
<tr>
<td>(14)</td>
<td>\dot{e}/2</td>
<td>--/day²</td>
</tr>
<tr>
<td>(15)</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>(16)</td>
<td>\dot{Ω}/2</td>
<td>rad/day²</td>
</tr>
<tr>
<td>(17)</td>
<td>\dot{ω}/2</td>
<td>rad/day²</td>
</tr>
<tr>
<td>(18)</td>
<td>\dot{n}/2</td>
<td>rad/day²</td>
</tr>
<tr>
<td>(19)</td>
<td>\dot{n}/6</td>
<td>rad/day³</td>
</tr>
<tr>
<td>(20)</td>
<td>\ddot{n}/24</td>
<td>rad/day⁴</td>
</tr>
</tbody>
</table>

DELT
Time to epoch in days, should be greater than 10⁻⁸ or else set to zero identically

C. Output

1. COMMON

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DYEAR</td>
<td>Calendar year - 1900</td>
</tr>
<tr>
<td>DMNTH</td>
<td>Calendar month</td>
</tr>
</tbody>
</table>

5-100
2. Calling sequence

TNOMX 6-word vector containing x, y, z, x, y, z in kilometers and kilometers/second
TNOMX(1) x kilometers
TNOMX(2) y kilometers
TNOMX(3) z kilometers
TNOMX(4) x kilometers/second
TNOMX(5) y kilometers/second
TNOMX(6) z kilometers/second

D. Error/action messages

E FAILED TO CONVERGE
THE VALUE OF E IS ______ E ___, THE FLAG IS _______

This message occurs if the iteration for E has failed to converge after 50 iterations. The flag = 0 indicates the iteration failed for conversion to osculating of the mean elements. The flag = 1 indicates the iteration failed for conversion to Cartesian of the osculating. The program proceeds normally, using the last value of E computed.

SUBROUTINES USED

A. Library
ABS
SIN
COS
ATNQ
SQRT

B. Program
PIMOD Takes principal value of angle between 0 and 2π
MTOC

DLSTV Computes the differentials used in converting from mean to osculating and osculating to mean

JTOC Converts Julian date to calendar date

EQUATIONS

Given $m_K^{-25}$, $e_m$, $i_m$, $\Omega_m$, $\omega_m$, $M_m$

1. Compute $E$ using

$$E_1 = \pi$$

$$E_{n+1} = E_n + \frac{M_m - E_n + e_m \sin E_n}{1 - e_m \cos E_n}$$

2. Compute true anomaly, $v$

$$\cos v = \frac{\cos E - e_m}{1 - e_m \cos E}$$

$$\sin v = \sqrt{1 - e_m^2} \frac{\sin E}{1 - e_m \cos E}$$

3. Compute radius vector

$$r = m_K^{-25} a_m^{-1} (1 - e_m \cos E)$$

4. Compute orbital semi-parameter

$$p_m = m_K^{-25} a_m^{-1} (1 - e_m^2)$$

5. Obtain $\delta$'s from DLSTV

5-102
6. Compute \( a_m \)

\[
a_m = \frac{a_{mK-25}}{\sqrt{1 - \frac{A^2}{2} \left(1 - \frac{3}{2} \sin^2 i_m\right) \sqrt{1 - e_m^2}}}
\]

7. Compute osculating elements

\[
a_{os} = a_m + \delta a_{mK-25} \quad \text{(a)} \quad e_{os} = e_m + \delta e_{mK-25} \quad \text{(e)} \quad i_{os} = i_m + \delta i_{mK-25} \quad \text{(i)} \quad \Omega_{os} = \Omega_m + \delta \Omega_{mK-25} \quad \text{(\Omega)} \quad \omega_{os} = \omega_m + \delta \omega_{mK-25} \quad \text{and} \quad M_{os} = M_m + \delta M_{mK-25}
\]

8. Convert to Cartesian

a. Obtain \( E \) and \( v \) as above

\[
u = v + \omega
\]

\[
l = u + \Omega_{os}
\]

\[
l_r = u - \Omega_{os}
\]
b. \[ U_x = \frac{1}{2} \left[ (1 + \cos i_{\text{os}}) \cos \ell + (1 - \cos i_{\text{os}}) \cos \ell \right] \]

\[ U_y = \frac{1}{2} \left[ (1 + \cos i_{\text{os}}) \sin \ell - (1 - \cos i_{\text{os}}) \sin \ell \right] \]

\[ U_z = \sin u \sin i_{\text{os}} \]

\[ V_x = -\frac{1}{2} \left[ (1 + \cos i_{\text{os}}) \sin \ell + (1 - \cos i_{\text{os}}) \sin \ell \right] \]

\[ V_y = \frac{1}{2} \left[ (1 + \cos i_{\text{os}}) \cos \ell - (1 - \cos i_{\text{os}}) \cos \ell \right] \]

\[ V_z = \cos u \sin i_{\text{os}} \]

c. \[ r = a_{\text{os}} (1 - e_{\text{os}} \cos E) \]

\[ \dot{r} = \frac{\sqrt{\mu a_{\text{os}}}}{r} e_{\text{os}} \sin E \]

\[ r \dot{\nu} = \frac{\sqrt{\mu a_{\text{os}}}}{r} \left( 1 - \frac{e_{\text{os}}^2}{r} \right) \]

d. \[ x = r U_x \]

\[ y = r U_y \]

\[ z = r U_z \]

\[ \dot{x} = \dot{r} U_x + r \dot{\nu} V_x \]

\[ \dot{y} = \dot{r} U_y + r \dot{\nu} V_y \]

\[ \dot{z} = \dot{r} U_z + r \dot{\nu} V_z \]

5-104
SUBROUTINE IDENTIFICATION

A. Title
   NTOM

B. Segment
   NRTPOD

C. Called by subroutine
   RADR

FUNCTION

To reduce a row of the unconstrained observational equations to the constrained system.

USAGE

A. Calling sequence
   Call NTOM

B. Input
   1. COMMON
      NPR       Size of unconstrained system
      NAROW     NSTR pointer for row of observational equations
      NR        VSTR pointer for ATA (constrained)
      MPR       Size of constrained system
      IMAX      Number of non-zero elements of constraint matrix
      NIJ       IVSTR pointer for coded subscripts of constraint matrix
      NB        VSTR pointer for non-zero elements of constraint matrix
      NC        VSTR pointer for additive constants
      CFLG      Flag for additive constants

2. Calling sequence

5-105
NTOM

C. Output
   1. COMMON
   
   2. Calling sequence
   
D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

EQUATIONS

A_o = AB where A_o is the constrained set of observational equations, A the unconstrained set, and B the constraint matrix.
SUBROUTINE IDENTIFICATION

A. Title
NXN

B. Segment
NRTPOD

C. Called by subroutine
FIT

FUNCTION

NXN expands the constrained inverse and the solution vector to the unconstrained system.

USAGE

A. Calling sequence
Call NXN (NDPAR, S, RI)

B. Input

1. COMMON

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPR</td>
<td>Size of unconstrained system</td>
</tr>
<tr>
<td>NR</td>
<td>VSTR pointer for ATA (constrained)</td>
</tr>
<tr>
<td>MPR</td>
<td>Size of constrained system</td>
</tr>
<tr>
<td>IMAX</td>
<td>Number of non-zero elements in constraint matrix</td>
</tr>
<tr>
<td>NIJ</td>
<td>IVSTR pointer for coded subscripts of constraint matrix</td>
</tr>
<tr>
<td>NST</td>
<td>Temporary storage used in VSTR for constraining the size of the system from NPR to MPR</td>
</tr>
<tr>
<td>NB</td>
<td>VSTR pointer for non-zero elements of constraint matrix</td>
</tr>
<tr>
<td>NC</td>
<td>VSTR pointer for additive constants</td>
</tr>
<tr>
<td>CFLG</td>
<td>Flag for additive constants</td>
</tr>
<tr>
<td>VSTR</td>
<td>Variable storage</td>
</tr>
<tr>
<td>IVSTR</td>
<td></td>
</tr>
</tbody>
</table>

5-107
2. Calling sequence

NDPAR VSTR pointer by solution vector
S Block of temporary storage \[ MPR (MPR + 1)/2 \]
RI Block of temporary storage MPR

C. Output

1. COMMON

VSTR (NDPAR) Constrained solution vector
VSTR (NR) Constrained \( A^T A \)

2. Calling sequence

SUBROUTINES USED

A. Library
B. Program

ELEM

EQUATIONS

\[
x = By + c
\]
\[
(A^T A)^{-1} = B \left( A^T_o A_o \right)^{-1} B^T
\]

where

- \( x \) = unconstrained solution vector
- \( y \) = constrained solution vector
- \( B \) = constraint matrix
- \( \left( A^T_o A_o \right)^{-1} = \) inverse of constrained normal matrix
SUBROUTINE IDENTIFICATION

A. Title
OBSIN

B. Segment
NRTPOD - Input Processor

C. Called by subroutine
LODOBS

FUNCTION

Function is to apply sensor biases, if any, scale observation data and weights ($\sigma$'s) to internal units, and move this data from temporary storage to permanent storage ($Z$). This routine overrides the weights input on sensor cards by the weights, if any, input on the observation cards.

USAGE

A. Calling sequence
Call OBSIN (Z, ISTART, NOB)

B. Input

1. Blank COMMON

   CKMER (km/e.r.)
   CDEG (Deg/radian)
   KOUT Output tape number
   NSSTB VSTR pointer for station means and RMS information
   NSTAT VSTR pointer for master sensor table
   Julian date of midnight, epoch day

2. Labeled COMMON

   /TEMP/
   TEMP (1) Station ID
   TEMP (2-7) Time of observation in year, month, day, hour, minute, second
   TEMP (8) type
   TEMP (9) $R$, range (e.r.) or $\dot{R}$, range acceleration (e.r./min$^2$)
   TEMP (10) $A$, azimuth (rad)
   TEMP (11) $E$, elevation (rad)
   TEMP (12) $\ddot{R}$, range rate (e.r./min)
   TEMP (13) $\sigma_R$, standard deviation in range (e.r.) or $\sigma_{\ddot{R}}$, standard deviation in range acceleration (e.r./min$^2$)
   TEMP (14) $\sigma_A$, standard deviation in azimuth (rad)
   TEMP (15) $\sigma_E$, standard deviation in elevation (rad)
   TEMP (16) $\sigma_R$, standard deviation in range rate (e.r./min)

5-109
3. Calling sequence

ISTART  Starting location of Z

C. Output

1. COMMON

2. Calling sequence

Z (ISTART)  STATION ID
Z (ISTART +1)  Time from epoch (min)
Z (ISTART +2)  R, range (e.r.) or
Z (ISTART +3)  A, azimuth (rad)
Z (ISTART +4)  E, elevation (rad)
Z (ISTART +5)  R, range rate (e.r./min)
Z (ISTART +6)  Type
Z (ISTART +7)  \( \sigma_R \), standard deviation in range (e.r.) or
Z (ISTART +8)  \( \sigma_A \), standard deviation in azimuth (rad)
Z (ISTART +9)  \( \sigma_E \), standard deviation in elevation (rad)
Z (ISTART +10)  \( \sigma_{R'} \), standard deviation in range rate

Note: Whenever observations are processed, the rest of the Z buffer contains zeros.

NOB  Flag to indicate error in observation ID.

\( = 0 \) ID found in master sensor table.
\( \neq 0 \) ID not found in master sensor table.

D. Error/action messages

1. Off-line comment

"ERROR IN OBSERVATION ID _____"

2. On-line comment

3. Action

Set NOB flag, return to calling program.
SUBROUTINES USED

A. Library

B. Program

TIME - Computes Julian date and minutes from midnight of epoch day
Figure 5-10. OBSIN Flow Diagram
SUBROUTINE IDENTIFICATION

A. Title
   PAGE2

B. Segment
   NRTPOD

C. Called by subroutine
   LINES

FUNCTION

Prints out the functional standard deviations on option (JDC column 43).

USAGE

A. Calling sequence
   Call PAGE2

B. Input
   1. COMMON
      KOUT                  Output tape number
   2. Calling sequence

C. Output
   1. COMMON
      SBUF
      Buffer array containing sets of functional standard deviations for each residual to make up one page of functional sigma output.
      SBUF(1) - Station ID
      (2) - Residual type = 0(R, A, E, R, ) = 1(R)
      (3) - Time of residual (min from epoch)
      (4) - Residual count
      (5) - $\sigma_R$ (kilometers) or $\sigma_{\dot{R}}$ ($\frac{\text{meters}}{\text{sec}^2}$)
      (6) - $\sigma_A$ (deg.)
      (7) - $\sigma_E$ (deg.)

5-113
2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

(8) - $\sigma_R$ (km/sec)

(9) - Station ID

...

(16) - $\sigma_R$ (km/sec)

(17) - Station ID

...

(24) - $\sigma_R$ (km/sec)

etc.

Eight cells per residual until SBUF is filled with an output page of functional standard deviations.
SUBROUTINE IDENTIFICATION

A. Title
   PLAMDA

B. Segment
   NRTPOD

C. Called by subroutines
   FALSIN
   TRJGEN

FUNCTION

Subroutine PLAMDA computes the partials of \( x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}, \) with respect to each \( \lambda_i \), where the \( \lambda_i \) are the drag parameters \( \frac{C_D A}{m_i} \) in the solution vector.

USAGE

A. Calling sequence
   Call PLAMDA (IFLAG)

B. Input

1. COMMON

<table>
<thead>
<tr>
<th>VSTR</th>
<th>Variable storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR2DPI(1-3)</td>
<td>( \frac{\partial \dot{x}}{\partial \alpha}, \frac{\partial \dot{y}}{\partial \alpha}, \frac{\partial \dot{z}}{\partial \alpha} )</td>
</tr>
<tr>
<td>(4-6)</td>
<td>( \frac{\partial \dot{x}}{\partial \delta}, \frac{\partial \dot{y}}{\partial \delta}, \frac{\partial \dot{z}}{\partial \delta} )</td>
</tr>
<tr>
<td></td>
<td>( \vdots )</td>
</tr>
<tr>
<td>(16-18)</td>
<td>( \frac{\partial \ddot{x}}{\partial V}, \frac{\partial \ddot{y}}{\partial V}, \frac{\partial \ddot{z}}{\partial V} )</td>
</tr>
<tr>
<td>(19-21)</td>
<td>( \frac{\partial \dot{x}}{\partial \lambda_j}, \frac{\partial \dot{y}}{\partial \lambda_j}, \frac{\partial \dot{z}}{\partial \lambda_j} )</td>
</tr>
<tr>
<td>(22-24)</td>
<td>( \frac{\partial \ddot{x}}{\partial \lambda_{j+1}}, \frac{\partial \ddot{y}}{\partial \lambda_{j+1}}, \frac{\partial \ddot{z}}{\partial \lambda_{j+1}} )</td>
</tr>
</tbody>
</table>

Where \( \lambda_j, \lambda_{j+1} \) are the current \( \frac{C_D A}{m} \) drag parameters in the region of influence.
KOUT Output tape number

NPALM Pointer to variable storage location where the \( b_i \)'s are stored

\[
b_i = \left[ \frac{\partial \left( x, y, z, \dot{x}, \dot{y}, \dot{z} \right)}{\partial \left( \alpha, \delta, \beta, A, r, v \right)} \right]_{t=t_i}^{-1} \left[ \frac{\partial \left( x, y, z, \dot{x}, \dot{y}, \dot{z} \right)}{\partial \lambda_i} \right]_{t=t_i}
\]
\[6 \times 6\]
\[6 \times 1\]
\[i = 1, \ldots, NLAM\]

\( \lambda_i = \left( \frac{C_{DA}}{m} \right)_i \)

NICPR Number of ADBARV variables in the solution vector

NPXLM Pointer to location in variable storage where the

\[
\left[ \frac{\partial \left( x, y, z, \dot{x}, \dot{y}, \dot{z} \right)}{\partial \left( \frac{C_{DA}}{m} \right)_i} \right]
\]
are stored

\[i = 1, \ldots, NLAM\]

NLAM Total number of entries in the altitude \( \left( \frac{C_{DA}}{m} \right) \) table

NPXDLM Pointer to location in variable storage where the

\[
\left[ \frac{\partial \left( \ddot{x}, \ddot{y}, \ddot{z} \right)}{\partial \left( \frac{C_{DA}}{m} \right)_i} \right]
\]
are stored

\[i = 1, \ldots, NLAM\]

NH Pointer to location in variable storage where the altitude - \( C_{DA} / m \) table is stored

IFVE Flags indicating whether the \( C_{DA} / m \) in the region of influence is in the solution vector or not

5-116
IFVE = 0  the $C_D A/m$ of a particular region is not in the solution vector

≠ 0  the $C_D A/m$ of a particular region is in the solution vector

INFG  Flag indicating whether an altitude crossing has occurred and which region of drag influence has been entered.

INFG = 0  no altitude crossing has occurred

= 1  vehicle has reentered altitude region 1

= 2  vehicle has left influence of altitude region 1 and crossed into altitude region 2

NDPRT  Number of CAT1 variables plus number of $C_D A/m$ drag parameters being integrated at any one time (either 6 or 8)

NH1  Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence

NH2  Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

IFLAG  Flag indicating to subroutine PLAMDA whether time is at an observation point or an altitude cutoff point

IFLAG = 0 signifies time at an observation time

= 1 signifies time at altitude cutoff

C. Output

1. COMMON

$\frac{\partial x}{\partial \lambda_i}$, $\frac{\partial y}{\partial \lambda_i}$, $\frac{\partial z}{\partial \lambda_i}$, $\frac{\partial \dot{x}}{\partial \lambda_i}$, $\frac{\partial \dot{y}}{\partial \lambda_i}$, $\frac{\partial \dot{z}}{\partial \lambda_i}$

5-117
where $\lambda_i$ and $\lambda_{i+1}$ indicate the two \((CpA/m)\) drag parameters in the region of influence.

2. Calling sequence

D. Error/action messages

If difficulty is encountered when computing

$$\begin{bmatrix}
\frac{\partial (x, y, z, \dot{x}, \dot{y}, \dot{z})_{t=t_i}}{\partial (\alpha, \delta, \beta, A, r, v)_{t=0}} \end{bmatrix}^{-1}
$$

where $t_i$ is the time at the \(i\)th altitude layer,

Subroutine LESK returns a non-zero error flag. If this occurs an error comment, "LINEAR EQUATION SOLVER - LESK ERROR RETURN, LA = ____", is printed and the program halts.

SUBROUTINES USED

A. Library
   EXIT

B. Program
   LESK
   MATMLT
   TRJOUT
Figure 5-11. PLAMDA Flow Diagram
CALL TRJOUT
SET UP TRAJECTORY TAPE PARAMETERS AND WRITE TRAJECTORY TAPE

< INFG: 1 >=

SET UP INTEGRATION LIST WITH PROPER PARTIALS \[ \frac{\partial x}{\partial \lambda} \] BASED ON ALTITUDE LAYER RE-ENTRY

RETURN

Figure 5-11. PLAMDA Flow Diagram (Continued)
SUBROUTINE IDENTIFICATION

A. Title
POPPC

B. Segment
NRTPOD

C. Called by subroutine
PRAUPD

FUNCTION

The function is to compute the matrix which takes a Cartesian covariance matrix into an ECI orbit plane matrix up, down, cross. The dimension of the matrix U is the total number of Category 1 variables and drag variables in the solution vector.

USAGE

A. Calling sequence
Call POPPC

B. Input

1. COMMON

   NDPR        Total number of Category 1 variables to solve for
   TEMP        Temporary storage
   TRAJX(1)    x (e. r.)
   TRAJX(2)    y (e. r.)
   TRAJX(3)    z (e. r.)
   TRAJX(4)    ẋ (e. r. /min)
   TRAJX(5)    ẏ (e. r. /min)
   TRAJX(6)    ẑ (e. r. /min)

2. Calling sequence
C. Output

1. COMMON

TDPDX

Contains matrices of partials for covariance matrix update

D. Error/action messages

SUBROUTINES USED

A. Library

SQRT

B. Program

EQUATIONS

\[ r = \sqrt{x^2 + y^2 + z^2} \]

\[ v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \]

\[ r \cdot v = x\dot{x} + y\dot{y} + z\dot{z} \]

\[ |J| = \sqrt{r^2v^2 - (r \cdot v)^2} \]

\[ |J \times r| = \sqrt{r^2(v^2 - (r \cdot v)^2)} \]

\[ \xi_x = \frac{x}{r} \quad \eta_x = \frac{r^2 \dot{x} - (r \cdot v)x}{|J \times r|} \]

\[ \xi_y = \frac{y}{r} \quad \eta_y = \frac{r^2 \dot{y} - (r \cdot v)y}{|J \times r|} \]

\[ \xi_z = \frac{z}{r} \quad \eta_z = \frac{r^2 \dot{z} - (r \cdot v)z}{|J \times r|} \]

\[ \xi_x = \frac{y\dot{z} - z\dot{y}}{|J|} \quad \xi_y = \frac{z\dot{x} - x\dot{z}}{|J|} \]

\[ \xi_z = \frac{xy - yx}{|J|} \]
\[ A = \begin{bmatrix}
\xi_x & \xi_y & \xi_z \\
\eta_x & \eta_y & \eta_z \\
\zeta_x & \zeta_y & \zeta_z
\end{bmatrix} \]

For NDPR = 6

\[ \text{TDPDX} = U = \begin{bmatrix}
A & 0 \\
0 & A
\end{bmatrix} \]

For NDPR > 6

\[ \text{TDPDX} = U = \begin{bmatrix}
A & 0 & 0 \\
0 & A & 0 \\
0 & 0 & I
\end{bmatrix}_{NDPR-6} \]

where \( I \) is the identity matrix

5-123
SUBROUTINE IDENTIFICATION

A. Title
   PPLPC

B. Segment
   NRTPOD

C. Called by subroutine
   PRAUPD

FUNCTION

Function is to compute the partial of polar coordinates with respect to Cartesian coordinates and to set up a matrix U necessary to perform the update $V = U \Sigma_x U^T$. The dimension of the matrix U is the total number of Category 1 variables and $C_D A/m$ drag parameters being solved for.

USAGE

A. Calling sequence
   Call PPLPC

B. Input
   1. COMMON
      NDPR          Total number of CAT 1 variables and $C_D A/m$ drag parameters being solved for.
      TRAJX         Position, velocity, and acceleration vectors of the vehicle.
      Also the TRAJX array contains the variational matrix computed from the integration of the variational equations.

C. Output
   1. COMMON
      TDPDX        Contains the matrices of partials for covariance matrix update (See equations.)

D. Error/action messages

5-125
SUBROUTINES USED

A. Library
   COS
   SQRT
   Cosine function
   Square root function

B. Program
   ATNQ
   Arc tangent function

EQUATIONS

\[ r^2 = x^2 + y^2 + z^2 \]
\[ v = \sqrt{x^2 + y^2 + z^2} \]
\[ r\hat{r} = x\hat{x} + y\hat{y} + z\hat{z} \]
\[ \hat{r} = \frac{r\hat{r}}{r} \]
\[ q = \frac{1}{r} \frac{\sqrt{v^2 - \hat{r}^2}}{v} \]
\[ A = \tan^{-1}\left( \frac{xy - y\hat{x}}{rz - z\hat{r}} \right) \]
\[ W = \frac{\cos^2 A}{rz - z\hat{r}} \]

\[ \frac{\partial}{\partial x} \theta = \frac{-y}{\sqrt{x^2 + y^2}} \frac{\partial}{\partial y} \theta = \frac{x}{\sqrt{x^2 + y^2}} \frac{\partial}{\partial z} \theta = \frac{\partial}{\partial x} \theta = \frac{\partial}{\partial y} \theta = \frac{\partial}{\partial z} \theta = 0 \]

\[ \frac{\partial}{\partial x} q = q \left( \frac{x\hat{r}}{r} - x \right) \frac{\partial}{\partial y} q = q \left( \frac{y\hat{r}}{r} - y \right) \frac{\partial}{\partial z} q = q \left( \frac{z\hat{r}}{r} - z \right) \]

\[ \frac{\partial}{\partial x} \beta = q \left( \frac{xy\hat{r}}{v^2} - x \right) \frac{\partial}{\partial y} \beta = q \left( \frac{y\hat{r}v^2}{v^2} - y \right) \frac{\partial}{\partial z} \beta = q \left( \frac{z\hat{r}v^2}{v^2} - z \right) \]
\[
\begin{align*}
\frac{\partial A}{\partial x} &= W \left[ \dot{y} - \tan A \left( \frac{\dot{x} - \ddot{x} z + \frac{z \dot{r}}{r}}{r} \right) \right] \\
\frac{\partial A}{\partial y} &= W \left[ -\dot{x} - \tan A \left( \frac{\dot{y} - \ddot{y} z + \frac{z \dot{r}}{r}}{r} \right) \right] \\
\frac{\partial A}{\partial z} &= \dot{r} W \tan A \left( 1 - \frac{z^2}{r^2} \right) \\
\frac{\partial A}{\partial \dot{x}} &= W \left( -\gamma + \frac{\dot{x} z}{r} \tan A \right) \\
\frac{\partial A}{\partial \dot{y}} &= W \left( x + \frac{\dot{y} z}{r} \tan A \right) \\
\frac{\partial A}{\partial \dot{z}} &= -r W \tan A \left( 1 - \frac{z^2}{r^2} \right)
\end{align*}
\]

\[
\begin{align*}
\frac{\partial r}{\partial x} &= \frac{\partial r}{\partial y} = \frac{\partial r}{\partial z} = \frac{\partial r}{\partial \dot{x}} = \frac{\partial r}{\partial \dot{y}} = \frac{\partial r}{\partial \dot{z}} = 0
\end{align*}
\]

\[
\begin{align*}
\frac{\partial v}{\partial x} &= \frac{\partial v}{\partial y} = \frac{\partial v}{\partial z} = 0 \\
\frac{\partial v}{\partial \dot{x}} &= \frac{\dot{x}}{v} \\
\frac{\partial v}{\partial \dot{y}} &= \frac{\dot{y}}{v} \\
\frac{\partial v}{\partial \dot{z}} &= \frac{\dot{z}}{v}
\end{align*}
\]

\[
[A] = \begin{bmatrix}
\frac{\partial A}{\partial x} & \frac{\partial A}{\partial y} & \frac{\partial A}{\partial z} & \frac{\partial A}{\partial \dot{x}} & \frac{\partial A}{\partial \dot{y}} & \frac{\partial A}{\partial \dot{z}} \\
\frac{\partial \delta}{\partial x} & \frac{\partial \delta}{\partial y} & \frac{\partial \delta}{\partial z} & \frac{\partial \delta}{\partial \dot{x}} & \frac{\partial \delta}{\partial \dot{y}} & \frac{\partial \delta}{\partial \dot{z}} \\
\frac{\partial \beta}{\partial x} & \frac{\partial \beta}{\partial y} & \frac{\partial \beta}{\partial z} & \frac{\partial \beta}{\partial \dot{x}} & \frac{\partial \beta}{\partial \dot{y}} & \frac{\partial \beta}{\partial \dot{z}} \\
\frac{\partial A}{\partial x} & \frac{\partial A}{\partial y} & \frac{\partial A}{\partial z} & \frac{\partial A}{\partial \dot{x}} & \frac{\partial A}{\partial \dot{y}} & \frac{\partial A}{\partial \dot{z}} \\
\frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial \dot{y}} & \frac{\partial r}{\partial \dot{z}} \\
\frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} & \frac{\partial v}{\partial \dot{x}} & \frac{\partial v}{\partial \dot{y}} & \frac{\partial v}{\partial \dot{z}}
\end{bmatrix}
\]
For \( \text{NDPR} = 6 \)

\[
\text{TDPDX} = \text{U} = \begin{bmatrix} \text{A} \end{bmatrix}
\]

For \( \text{NDPR} > 6 \)

\[
\text{TDPDX} = \text{U} = \begin{bmatrix} \text{A} & 0 \\ 0 & \text{I} \end{bmatrix}
\]

where I is the identity
A. Title
   PRAUPD

B. Segment
   NRTPOD

C. Called by subroutine
   PRUDRV

FUNCTION

Function is to update a given covariance matrix to a specified time $t$, and to print the resulting matrices. The covariance matrix to be updated can either be a $6 \times 6 (a, \delta, \beta, A, R, v)$ or an $n \times n (a, \delta, B, A, R, v, C_{D}A/m'S)$. The updated normal matrix (polar spherical coordinates) and a correlation matrix is printed. Eigenvalues and eigenvectors of the UVW covariance matrix are output along with determinants of the phase space Cartesian covariance matrix.

USAGE

A. Calling sequence
   CALL PRAUPD

B. Input
   1. COMMON
      KOUT   Symbolic output tape (print)
      NPR    Number of all parameters to solve for
      NDPR   Number of Category 1 parameters to solve for
      NATA   Starting location of where the triangular $A^T A$ is stored [VSTR(NATA)]
      NR     Starting location of where the inverse $A^T A$ is stored [VSTR(NR)]
      NSCALE Starting location of the list of conversion factors which convert from machine to output units and vice versa
      NDPAR1 Starting location where the solution vector will be stored
      NRTMP   Starting location of temporary storage for special handling of the $R$ matrix
      NBDNS  Starting location for the bounds used by LEGS2

5-129
PRAUPD

TEMP  Temporary storage
VSTR   Variable storage. VSTR(NR), VSTR(NRTMP), VSTR(NBDNS) etc.
TRAJX  Contains the position, velocity and acceleration vectors of the vehicle
        The variational equations may also be present in TRAJX
TZ     Indicates if the solution was affected by bounds
CKMER  Km/e.r.
CDEG   Deg/radian
KOUT   Output tape number
ITRJTP  Trajectory tape number
DYEAR  Epoch year
DMNTH  Epoch month
DDAY   Epoch day
TG     Current integration time
TCRASH Impact flag
        =0 no impact
        ≠0 impact

2. Calling sequence

C. Output
   Off-line print
      Sigma and Rho matrix (polar spherical coordinates)
      Normal matrix (polar spherical coordinates)
      Covariance matrix output (see description of output)
      This routine outputs the eigenvalues and eigenvectors along with yaw-roll-pitch rotations aligning the UVW system with the principal axes of error ellipsoid. The determinants mentioned in the output description section are also output by this routine.

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
   MATPT
   HUMAH
   PPLPC
   CORMAT
   MABAT
   LEGS2
   POPPC
   PRAXIS
   UPPER

5-130
PRAXIS

SUBROUTINE IDENTIFICATION

A. Title
   PRAXIS

B. Segment
   NRTPOD

C. Called by subroutine
   UPDATE

FUNCTION

The functions of this subroutine are described below:

a) To compute the eigenvalues and eigenvectors of a real symmetric 3 x 3 matrix, A (stored as a lower triangular matrix). The eigenvectors for the columns of a matrix U and are ordered as column vectors in such a way that the sum of the diagonal elements of the U matrix is maximized.

b) These eigenvectors are then used to compute the three angles $\phi_1$, $\phi_2$, $\phi_3$ which will resolve the matrix A into a diagonal matrix with the eigenvalues of A as the diagonal elements.

USAGE

A. Calling sequence
   Call PRAXIS (A, I, B, J)

B. Input
   1. COMMON

2. Calling sequence

   L(A) Address of an array A where the matrix is stored

   I Index to indicate just where in the above array the first element of the matrix is [i.e., A(I) is the first element of the matrix].

   L(B) Address of an array B where the results of PRAXIS are to be stored

   J Index to indicate just where in the above array the first element of the results are to be stored (See Output for arrangement of results in array B).
C. Output

1. COMMON

2. Calling sequence

\[
\begin{align*}
B(J) &\ - \lambda_1 \\
B(J+1) &\ - \lambda_2 \\
B(J+2) &\ - \lambda_3 \\
B(J+3) &\ - U_{11} \\
B(J+4) &\ - U_{12} \\
B(J+5) &\ - U_{13} \\
B(J+6) &\ - U_{21} \\
B(J+7) &\ - U_{22} \\
B(J+8) &\ - U_{23} \\
B(J+9) &\ - U_{31} \\
B(J+10) &\ - U_{32} \\
B(J+11) &\ - U_{33} \\
B(J+12) &\ - \phi_1 \\
B(J+13) &\ - \phi_2 \\
B(J+14) &\ - \phi_3 \\
B(J+15) &\ - \sqrt[5]{\lambda_1} \\
B(J+16) &\ - \sqrt[5]{\lambda_2} \\
B(J+17) &\ - \sqrt[5]{\lambda_3}
\end{align*}
\]

\text{eigenvalues of } A

\text{first eigenvector}

\text{second eigenvector}

\text{third eigenvector}

\text{rotational angles (rad)}

\text{square roots of the three eigenvalues}

D. Error/action messages

SUBROUTINES USED

A. Library

SQRT
COS
SIN

5-132
B. Program

<table>
<thead>
<tr>
<th>ATNQ</th>
<th>Arctangent routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCROSS</td>
<td>Cross product routine</td>
</tr>
</tbody>
</table>

**EQUATIONS**

Compute the eigenvalues of $A$

$$m = \frac{1}{3} \text{tr} (A) \text{ where } \text{tr}(A) = \sum_{i=1}^{3} a_{ii}$$

$$q = \frac{1}{2} \det(A - ml)$$

$6p = \text{sum of the squares of the elements of } (A - ml)$. From "Cardano's" trigonometric solution of $\det [(A - ml) - \mu I]$ as a cubic in $\mu$, the eigenvalues of $A$ are

$$\lambda_1 = m + 2 \sqrt{p} \cos \phi$$

$$\lambda_2 = m - \sqrt{p} (\cos \phi + \sqrt{3} \sin \phi)$$

$$\lambda_3 = m - \sqrt{p} (\cos \phi - \sqrt{3} \sin \phi)$$

where

$$\phi = \frac{1}{3} \tan^{-1} \frac{\sqrt{p^3 - q^2}}{q} \quad 0 \leq \phi \leq \frac{\pi}{3}$$

Compute the eigenvectors. Let $\lambda$ represent one of the three eigenvalues $\lambda_1$, $\lambda_2$, $\lambda_3$. 

$$\vec{C}_1 = \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix} \times \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix}$$


\[
\bar{C}_2 = \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix} \times \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix}
\]

\[
\bar{C}_3 = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix} \times \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix}
\]

If \( \bar{C}_1 \cdot \bar{C}_2 < 0 \); set \( \bar{C}_2 = -\bar{C}_2 \). If \( \bar{C}_1 \cdot \bar{C}_3 < 0 \); set \( \bar{C}_3 = -\bar{C}_3 \).

\( \bar{u} = 1/3(\bar{C}_1 + \bar{C}_2 + \bar{C}_3) \). \( \bar{u} \) is the eigenvector corresponding to \( \lambda \).

Letting the three eigenvectors form the columns of the matrix \( U \), the following diagram shows the logic used to maximize the sum of the diagonal elements of \( U \).
Finally, compute $\phi_1, \phi_2, \phi_3$

$$\phi_1 = \tan^{-1} \left[ -\frac{u_{32}}{u_{33}} \right]$$

$$\phi_2 = \sin^{-1} \left[ -\frac{u_{31}}{u_{31}} \right]$$

$$\phi_3 = \tan^{-1} \left[ -\frac{u_{21}}{u_{11}} \right]$$
SUBROUTINE IDENTIFICATION

A. Title
   PRELIM

B. Segment
   NRTPOD

C. Called by subroutine
   RADR

FUNCTION

The function is to calculate preliminary quantities for the formulation of residuals and partial derivatives of observation with respect to solution parameters.

USAGE

A. Calling sequence
   Call PRELIM

B. Input
   1. COMMON
      a. PSTAT(4)   Cos φ
                     PSTAT(5)   Sin φ
                     PSTAT(6)   αθ₀ + λ (rad)
                     PSTAT(7)   w₁ (e.r.)
                     PSTAT(8)   w₃ (e.r.)
      b. PUBS(1)    T (min)
                     PUBS(6)    R (e.r./min)
      c. TRAJX(1)   x
                     TRAJX(2)   y
                     TRAJX(3)   z
                     TRAJX(4)   ¨x
                     TRAJX(5)   ¨y
                     TRAJX(6)   ˙z
                     TRAJX(7)   ˙z
                     TRAJX(8)   ˙y
                     TRAJX(9)   ˙x
                     TRAJAX (> 9) partials of TRAJX (1-9) with respect to Pᵢ, i = 1, NDPR
                     P = parameters in the solution vector

5-137
d. **NDPR**

   Number of all differential plus initial parameters to solve for (Category 1) including drag parameters \( C_D A/m's \).

e. **TEMP**

   Temporary storage

f. **CWE**

   Earth's rotational rate

2. **Calling sequence**

C. **Output**

1. **COMMON**

   **PCMR**
   
   **PCSA**
   
   **PCSA LF**
   
   **PCSE**
   
   **PRSUBL**
   
   **PSNA**
   
   **PSNALF**
   
   **PSNE**
   
   **PUDDTI**

   **PUI**

   **PV**
   
   **PVI**

   **PWDTI**
   
   **PWDTPP**
   
   **PWI**

   **PWPP**
   
   **PR2DQ**
    
   **PWDT2P**
   
   **PUDDTI**
   
   **PWDT2**

2. **Calling sequence**

D. **Error/action messages**

---

5-138
SUBROUTINES USED

A. Library
   COS
   SIN
   SQRT

B. Program

EQUATIONS

The computed orbit positions \((x, y, z)\) and station positions \((\phi^*, \lambda, h)\) are processed to produce geocentric and topocentric coordinates of the vehicle in an Earth-fixed coordinate system. Right ascensions of the station for times of observations \(t_i\) are

\[
\alpha_i = (\alpha_0 + \lambda) + \omega_e (t_i - t_0)
\]

Geocentric position and velocity of the vehicle in Earth-fixed coordinates are

\[
\begin{bmatrix}
  \dot{w}_1 \\
  \dot{w}_2 \\
  \dot{w}_3
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha_i & \sin \alpha_i & 0 \\
  -\sin \alpha_i & \cos \alpha_i & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
\]

\[
\begin{bmatrix}
  \ddot{w}_1 \\
  \ddot{w}_2 \\
  \ddot{w}_3
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha_i & \sin \alpha_i & 0 \\
  -\sin \alpha_i & \cos \alpha_i & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  \dot{x} + \omega_e \dot{y} \\
  -\omega_e \dot{x} - \dot{\omega} \\
  \ddot{z}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  \dddot{w}_1 \\
  \dddot{w}_2 \\
  \dddot{w}_3
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha_i & \sin \alpha_i & 0 \\
  -\sin \alpha_i & \cos \alpha_i & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  \ddot{x} + 2\omega_e \dot{\dot{y}} - \omega_e^2 x \\
  -2\omega_e \dot{x} - \dddot{\omega} \\
  \ddot{\dot{z}}
\end{bmatrix}
\]

5-139
The station position in meridian coordinates is provided by the preprocessor module where it is computed from geodetic latitude, $\phi^*$, and altitude, $h$, as follows.

$$
A_s = \cos^2 \phi^* + b_e^2 \sin^2 \phi^* \\
B_s = \left( \sin^2 \phi^* + \frac{1}{b_e^2} \cos^2 \phi^* \right)^{-1/2}
$$

$$
w_1^s = (a_e A_s + h) \cos \phi^* \\
w_3^s = (b_e B_s + h) \sin \phi^*
$$

where $a_e$ and $b_e$ are the equatorial and polar axes of the reference spheroid respectively.

Topocentric coordinates, direction cosines, and related quantities for the vehicle in meridian plane coordinates system are then

$$
q_1 = w_1 - w_1^s \quad \text{(Topocentric position in equatorial coordinate system)}
$$

$$
q_2 = w_2
$$

$$
q_3 = w_3 - w_3^s
$$

$$
R = \sqrt{q_1^2 + q_2^2 + q_3^2}
$$
\[ \bar{u} = \begin{cases} u_1 = q_1/r \\ u_2 = q_2/r \\ u_3 = q_3/r \end{cases} \]

\[ \ddot{\bar{u}} = \begin{cases} \dot{u}_1 = (\dot{w}_1 - Ku_1)/r \\ \dot{u}_2 = (\dot{w}_2 - Ku_2)/r \\ \dot{u}_3 = (\dot{w}_3 - Ku_3)/r \end{cases} \]

\[ K = u_1 \dot{w}_1 + u_2 \dot{w}_2 + u_3 \dot{w}_3 \]

\[ \bar{v} = \begin{cases} v_1 = u_2 \\ v_2 = -u_1 \sin \phi^* + u_3 \cos \phi^* \\ v_3 = u_1 \cos \phi^* + u_3 \sin \phi^* \end{cases} \]

\[ V = \sqrt{v_1^2 + v_2^2} \]

\[ R_1 = VR \]

\[ \sin E = v_3 \]

\[ \cos E = V \]

\[ \cos A = v_2/V \]

\[ \sin A = v_1/V \]
If range rate observations are used (PUBS \neq 0), then variational equations in velocity are rotated as follows:

\[
\begin{bmatrix}
\frac{\partial w_1}{\partial p_i} \\
\frac{\partial w_2}{\partial p_i} \\
\frac{\partial w_3}{\partial p_i}
\end{bmatrix} =
\begin{bmatrix}
cos \alpha & \sin \alpha & 0 \\
-sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{\partial x}{\partial p_i} \\
\frac{\partial y}{\partial p_i} \\
\frac{\partial z}{\partial p_i}
\end{bmatrix}
\]

where the parameters \( p_i \) are the ADBARV conditions at epoch \( (\alpha_0, \delta_0, \beta_0, \lambda_0, r_0, v_0) \), and drag parameters \( (C_D A/m) \).

If range acceleration observations are used, the variational equations are rotated as follows:

\[
\begin{bmatrix}
\frac{\partial \dot{w}_1}{\partial p_i} \\
\frac{\partial \dot{w}_2}{\partial p_i} \\
\frac{\partial \dot{w}_3}{\partial p_i}
\end{bmatrix} =
\begin{bmatrix}
cos \alpha & \sin \alpha & 0 \\
-sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{\partial \dot{x}}{\partial p_i} + 2\omega \frac{\partial y}{\partial p_i} - \omega \frac{\partial x}{\partial p_i} \\
\frac{\partial \dot{y}}{\partial p_i} - 2\omega \frac{\partial x}{\partial p_i} + \omega \frac{\partial y}{\partial p_i} \\
\frac{\partial \dot{z}}{\partial p_i}
\end{bmatrix}
\]
SUBROUTINE IDENTIFICATION

A. Title
   PRTATA

B. Segment
   NRTPOD

C. Called by subroutine
   APPLY

FUNCTION

The functions are to move the de-augmented $A^T A$ and store by rows as a lower triangular matrix in VSTR(NRTMP), to scale as an input $A^T A$ inverse, and to print the $A^T A$ by MATPT.

USAGE

A. Calling sequence
   Call PRTATA

B. Input

1. COMMON
   NPR
   NATA
   NRTMP
   NSCALE
   KOUT
   MPR
   Total number of parameters to solve for
   Starting location of where the triangular
   $A^T A$ is stored
   Starting location of temporary storage for
   special handling of the R matrix
   Starting location of list of conversion
   factors which convert from machine to
   output units and vice-versa
   Symbolic output tape
   Number of parameters in solution vector
   (constrained system)

2. Calling sequence
C. Output

1. COMMON
   VSTR (NRTMP) Contains the scaled $A^T A$ normal matrix which is output off-line

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
   HUMAH
   MATPT
SUBROUTINE IDENTIFICATION

A. Title
   PRUDRV

B. Segment
   NRTPOD

C. Called by Subroutine
   TRJPRO

FUNCTION

Function is to control the post-processing capability of NRTPOD. The trajectory propagation and covariance matrix update is performed in this post-processing link.

USAGE

A. Calling sequence
   Call PRUDRV

B. Input
   1. COMMON
      ITRJTP     Trajectory Tape
      PSTFLG    Columns 51-60 on JDC card
      TEMP      Array of temporary storage
      TRAJX     Array containing position, velocity accelerations and partials of position, velocity, and acceleration with respect to ADBARV and the two drag layers of current influence
      CDAD2M    $C_D A/2m$ — (ft$^2$/slug)
      TRHOO     Density of air at TALT (slugs/ft$^3$)
   2. Calling sequence

C. Output
   1. COMMON
      TG        Time to integrate to (min)
      TCRASH    Impact flag
2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
   TPRNT          Routine to print trajectory block
   PRAUPD        Prints and updates covariance and correlation matrices
SUBROUTINE IDENTIFICATION

A. Title
   PUPB

B. Segment
   NRTPOD

C. Called by subroutine
   RADR

FUNCTION

The function of this subroutine is to evaluate the partials of observations with respect to biases of time, sensor latitude, sensor longitude, and sensor altitude. The observation type and the bias type are given in the calling sequence.

USAGE

A. Calling sequence
   PUPB (I, J)

B. Input
   1. COMMON
      COUNT       Number of lines
      PCMR        R = computed slant range
      PCSA        Cos A_c
      PCSALF      Cos (α_g)
      PCSE        Cos E_c
      PRSUB1      R_1 = VR
      PSNA        Sin A_c
      PSNALF      Sin (α_g)
      PSNE        Sin E_c
      PSTAT       Working storage for sensor information
      PUDTI       Vector (u_1, u_2, u_3)
      PUI         Vector (u_1, u_2, u_3)
      PUBS        Current observation buffer; ID, time, R or ̇R, A, E, ̇R, type
                   (If type (PUBS(7)) = 1, observation is ̇R)
PV $\sqrt{v_1^2 + v_2^2}$

PVI Vector $(v_1, v_2, v_3)$

PWDTI Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$

PWI Vector $(w_1, w_2, w_3)$

PUDDTI Vector $(\ddot{u}_1, \ddot{u}_2, \ddot{u}_3)$

PWDT2 Vector $(\ddot{w}_1, \ddot{w}_2, \ddot{w}_3)$

TRAJX $x, y, z, x, y, z \cdots$

CWE Earth's rotational rate (rad/min)

KOUT Output tape number

2. Calling sequence

$J = 1$ for $R$ or $\dot{R}$

$= 2$ for $A$

$= 3$ for $E$

$= 4$ for $\ddot{R}$

$I = 7$ for $t_b$

$= 8$ for $\phi_b$

$= 9$ for $f_b$

$= 10$ for $h_b$

C. Output

1. COMMON

2. Calling sequence

\[ A\text{ register} = \frac{\partial (\text{variable } J)}{\partial (\text{variable } I)} \]

D. Error/action messages

PARTIAL ( ) WITH RESPECT ( ) ASKED FOR

Given off-line when $I$ and $J$ exceed current program limits ($I = 10$, $J = 6$)
SUBROUTINES USED

A. Library

B. Program

LINES Line counter

EQUATIONS

Range (J = 1 and type = 0)

\[ \frac{\partial R}{\partial \phi^x} = u_1 w_3^s - u_3 w_1^s \text{ (type 8 bias)} \]

\[ \frac{\partial R}{\partial \lambda} = u_1 w_2 - u_2 w_1 \text{ (type 9 bias)} \]

\[ \frac{\partial R}{\partial h} = -u_1 \cos \phi^* - u_3 \sin \phi^* \text{ (type 10 bias)} \]

\[ \frac{\partial R}{\partial t} = u_1 \dot{w}_1 + u_2 \dot{w}_2 + u_3 \dot{w}_3 \text{ (type 7 bias)} \]

Azimuth (J = 2)

\[ \frac{\partial A}{\partial \phi^x} = \frac{\sin A}{R_1} (w_1 \cos \phi^* + w_3 \sin \phi^*) \text{ (type 8 bias)} \]

\[ \frac{\partial A}{\partial \lambda} = \frac{-w_1 \cos A + w_2 \sin \phi^* \sin A}{R_1} \text{ (type 9 bias)} \]

\[ \frac{\partial A}{\partial h} = 0 \text{ (type 10 bias)} \quad R_1 = VR \]

\[ \frac{\partial A}{\partial t} = \frac{1}{v^2} (v_2 \dot{v}_1 - v_1 \dot{v}_2) \text{ (type 7 bias)} \]

Elevation (J = 3)

\[ \frac{\partial E}{\partial \phi^x} = \frac{1}{R_1} \left(w_3 \cos \phi^* - w_1 \sin \phi^* - \frac{\partial R}{\partial \phi^x} \sin E\right) \text{ (type 8 bias)} \]

\[ \frac{\partial E}{\partial \lambda} = \frac{1}{R_1} \left(w_2 \cos \phi^* - \frac{\partial R}{\partial \lambda} \sin E\right) \text{ (type 9 bias)} \]
\[ \frac{\partial E}{\partial h} = - \frac{1}{R_1} \left( 1 + \frac{\partial R}{\partial h} \sin E \right) \text{(type 10 bias)} \]
\[ \frac{\partial E}{\partial t} = \frac{\dot{u}_1 \cos \phi^* + \dot{u}_3 \sin \phi^*}{\cos E} \text{ (type 7 bias)} \]

Range Rate \((J = 4)\)

\[ \frac{\partial \dot{R}}{\partial \phi^*} = w_3 \dot{s}_1 - w_1 \dot{s}_3 \text{ (type 8 bias)} \]
\[ \frac{\partial \dot{R}}{\partial \lambda} = (\dot{w}_2 - w_1 \dot{u}_2) + (\dot{w}_2 - w_1 \dot{u}_2) \text{ (type 9 bias)} \]
\[ \frac{\partial \ddot{R}}{\partial h} = -\dot{u}_1 \cos \phi^* - \dot{u}_3 \sin \phi^* \text{ (type 10 bias)} \]
\[ \frac{\partial \ddot{R}}{\partial \lambda} = \ddot{R} = \ddot{u} \cdot \ddot{W} + \ddot{u} \cdot \ddot{W} \text{ (type 10 bias)} \]

where
\[
\ddot{W} = \begin{cases} 
\dot{w}_1 &= -\omega_2^2 \dot{w}_1 + 2 \omega_e (-\dot{x} \sin \alpha + \dot{y} \cos \alpha) + (\dddot{x} \cos \alpha + \dddot{y} \sin \alpha) \\
\dot{w}_2 &= -\omega_2^2 \dot{w}_2 + 2 \omega_e (-\dot{x} \cos \alpha - \dot{y} \sin \alpha) + (-\dddot{x} \sin \alpha + \dddot{y} \cos \alpha) \\
\dot{w}_3 &= \dddot{z}
\end{cases}
\]

Range Acceleration \([J = 1 \text{ and PUBS}(7) = 1]\)

\[ \frac{\partial \dddot{R}}{\partial \phi^*} = w_3 \dot{s}_1 - w_1 \dot{s}_3 \text{ (type 8 bias, I = 8)} \]
\[ \frac{\partial \dddot{R}}{\partial \lambda} = (\dddot{u}_1 w_2 - \dddot{u}_2 w_1) + 2(\dot{u}_1 \dddot{w}_2 - \dot{u}_2 \dddot{w}_1) + (\dddot{w}_2 u_1 - \dddot{w}_1 u_2) \text{ (type 9 bias, I = 9)} \]
\[ \frac{\partial \dddot{R}}{\partial h} = -\dot{u}_1 \cos \phi^* - \dot{u}_3 \sin \phi^* \text{ (type 10 bias, I = 10)} \]
SUBROUTINE IDENTIFICATION

A. Title
   RADR

B. Segment
   NRTPOD

C. Called by subroutine
   DCITER

FUNCTION

Function is to control region for the formulation of the system of equations to be solved (Ax = B). A is the matrix of partial derivatives of observations with respect to solution variables and B is the vector of observation residuals. RADR also drives those routines which, given A, B, form A^T A, A^T B, and B^T B. It also drives the residuals print routines.

USAGE

A. Calling sequence
   Call RADR

B. Input
   1. COMMON
      IPFRST 0 to indicate first time in RADR
      NAROW Starting location where one row of the augmented matrix (A, B) is stored
      NPR Number of all parameters to solve for
      PCMR Computed slant range
      POBCNT Total number of accepted observations
      PRESD Residuals
      PSIG Sigma list
      PUBS Sensor number, time, R, A, E, Ṛ, Ė, table
      PUI Vector (u_1, u_2, u_3)
      NDPR Number of CAT 1 variables in the solution vector
      NPBIS Pointer to starting location in VSTR of biases being solved for

5-151
### RADR

**NPRCD**
Starting location of code words denoting biases to be solved for

**NSSTB**
Starting location of the mean and SOS table of residuals by type and station

**MPR**
Number of parameters in solution vector
(constrained vector)

**PDELFG**
Array containing editing symbols
(*, N, K, S or blank)

**PSTAT**
Current working sensor array

**PR2DOT**
Computed range acceleration

**PVI**
Vector \( (v_1, v_2, v_3) \)

**PWDTI**
Vector \( (\dot{w}_1, \dot{w}_2, \dot{w}_3) \)

**TSUS**
Current total SOS

**VSTR**
Floating point variable storage

**CPI**
\( \pi \)

**C2PI**
\( 2\pi \)

**PCSE**
\( \cos E_c \)

#### Calling sequence

1. **COMMON**
   - The array VSTR (NATA) contains the total \( A^T A, A^T B, B^T B \).

2. **Calling sequence**

#### Error/action messages

#### SUBROUTINES USED

**A. Library**

**B. Program**
- **ASIN**
  - Arc sine routine
- **ATNQ**
  - Arc tangent routine
- **DRDP**
  - Partials of observations w. r. t.
  - Category 1 variables
LEGSI  Forms $A^T A$ and $A^T B$ given $A$ and $B$

PIMOD  Principal value of angle between 0 and $2\pi$

PRELIM  Preliminary calculations

PAGE1  Accumulates residuals and prints

LINES  Counts output lines per page

FSIGMA  Functional sigma subroutine

PAROUT  Up, down, cross coordinate system routine

SSTB  Computes mean, SOS of residuals by type and station

NTOM  Linear constraint subroutine

REJECT  Editing of residuals processor

PUPB  Computes partials of observations w.r.t. biases of time, sensor latitude, longitude, and altitude

**EQUATIONS**

**Computation of Observables from Fitted Orbit**

The fitted orbit is used to produce computed "observables" for comparison with observations.

\[
R = \sqrt{q_1^2 + q_2^2 + q_3^2} \quad \text{(range)}
\]

\[
A = \tan^{-1} \frac{v_1}{v_2} \quad \text{(azimuth)}
\]

\[
E = \sin^{-1} \frac{v_3}{v_3} = \cos^{-1} V \quad \text{(elevation)}
\]

\[
\dot{R} = \overline{u} \cdot \overline{W} \quad \text{(range rate)}
\]

\[
\ddot{R} = \ddot{u} \cdot \overline{W} + \overline{u} \cdot \overline{W} \quad \text{(range acceleration)}
\]
Figure 5-12. RADR Flow Diagram

5-154
Figure 5-12. RADR Flow Diagram (Continued)
SUBROUTINE IDENTIFICATION

A. Title
RDDATA

B. Program
NRTPOD - Input processor

C. Called by subroutine
INPUT

FUNCTION

To read off-line into core storage all NAMELIST input and the lunar-solar ephemeris data

USAGE

A. Calling sequence
CALL RDDATA

B. Input

1. Blank COMMON

   KIN    Symbolic input tape
   KOUT   Symbolic output tape (print)
   COM    Variables in BLK1 blank COMMON (See NAMELIST input and Layout of COMMON Storage sections)
   DTMAX  A provision for editing residuals by input (See NAMELIST input section)
   NDFAYS Number of days of lunar-solar ephemeris input (See NAMELIST input section)
   CNSIG  N for N (σ) deletion, a provision for editing residuals by input. (NAMELIST Input)
   TIME   A 6-cell array containing epoch time in year, month, day, hour, minutes, seconds (NAMELIST Input)
   DELTT  Sets of Δt (See NAMELIST Input section)
NITER
Number of iterations desired in curve fit, nominally = 1 (NAMELIST Input)

TYPE
Indicates type of initial conditions (position and velocity) input to NRTPOD (NAMELIST Input)

BFLAGS
Flags indicating whether the sun and moon are to be included in the trajectory simulation (See NAMELIST Input)

CKRMS
A provision for editing residuals by input (See NAMELIST Input section)

2. Labeled COMMON

/INPP/
DRAG
\[ \frac{C_D A}{m \text{ meter}^2 \text{ kilogram}} \] (NAMELIST Input)

DRAGCD
Coefficient of drag in DRAG (NAMELIST Input)

DRAGA
Area in DRAG term (meters^2) (NAMELIST Input)

DRAGM
Mass in DRAG term (kilogram) (NAMELIST Input)

STVEC
Array identifying the initial position and velocity. (See NAMELIST Input section)

CAT1
The CAT1 array indicates to the program the Category 1 variables to be solved for (See NAMELIST Input section)

CAT2
The CAT2 array indicates to the program the Category 2 variables to be solved for (See NAMELIST Input section)

BISES
Bias estimates: (See NAMELIST Input section)

SMAT
A priori normal matrix (See NAMELIST Input)

DELET
Input provided to edit residuals (See NAMELIST Input)

BNDS
Bounds specified to control convergence for each CAT1 or CAT2 variable selected for solution (NAMELIST Input)

ZONAL
Array of flags for callouts of the coefficients of the zonal harmonics - J_2, ..., J_12 (NAMELIST Input)

SECT
Array of flags for callouts of the sectorial harmonics (See NAMELIST Input Section)

5-158
<table>
<thead>
<tr>
<th>RDDATA</th>
<th>RDDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESS</td>
<td>Array of code words for selection of tesseral harmonics (See NAMELIST Input section)</td>
</tr>
<tr>
<td>RADPR</td>
<td>Radiation pressure parameter, ( \frac{YA}{m^2} ) (meter(^2)) (See NAMELIST Input)</td>
</tr>
<tr>
<td>RPGAM</td>
<td>Radiation pressure parameter, ( Y ), reflectivity constant (NAMELIST Input)</td>
</tr>
<tr>
<td>RPA</td>
<td>Radiation pressure parameter, ( A ), effective area of vehicle in square meters (NAMELIST Input)</td>
</tr>
<tr>
<td>RPM</td>
<td>Radiation pressure parameter, ( m ), mass of the vehicle in kilograms (NAMELIST Input)</td>
</tr>
<tr>
<td>CJ</td>
<td>Zonal harmonics, ( J_2, \ldots, J_{12} ). May be altered on input (NAMELIST Input)</td>
</tr>
<tr>
<td>CJNM</td>
<td>Coefficients of the sectorial and tesseral harmonics and their associated angles (See NAMELIST Input)</td>
</tr>
<tr>
<td>CLAMNN</td>
<td>Array containing values of the angles associated with the coefficients of the tesseral harmonics; ( \lambda_2, \lambda_3, \ldots, \lambda_6 ) (See NAMELIST Input section)</td>
</tr>
<tr>
<td>UPMAT</td>
<td>A priori covariance matrix (See NAMELIST Input)</td>
</tr>
<tr>
<td>TPOS</td>
<td>A 60-cell vector containing the position of the moon and sun for NDAYS days. TPOS array order is</td>
</tr>
<tr>
<td></td>
<td>( x_{a1}, y_{a1}, z_{a1}, x_{o1}, y_{o1}, z_{o1}, \ldots ) *</td>
</tr>
<tr>
<td></td>
<td>( \ldots, x_{a\text{NDAYS}}, y_{a\text{NDAYS}}, z_{a\text{NDAYS}} )</td>
</tr>
<tr>
<td></td>
<td>( x_{o\text{NDAYS}}, y_{o\text{NDAYS}}, z_{o\text{NDAYS}} )</td>
</tr>
<tr>
<td></td>
<td>Units of earth radii - mean of 1950</td>
</tr>
<tr>
<td>TDEL2</td>
<td>A 60-cell vector containing the second central differences of the position ephemeris of the moon and sun for NDAYS days. TDEL2 array order is</td>
</tr>
<tr>
<td></td>
<td>( \delta^2 x_{a1}, \delta^2 y_{a1}, \delta^2 z_{a1}, \delta^2 x_{o1}, \delta^2 y_{o1}, \ldots )</td>
</tr>
</tbody>
</table>

* a - moon

* o - sun

5-159
\[ \ldots, \delta^2 x_{\text{NDAYS}}', \delta^2 y_{\text{NDAYS}}', \delta^2 z_{\text{NDAYS}}' \]

Units of earth radii - mean of 1950.

**TDEL4**

A 60-cell vector containing the fourth central differences of the position ephemeris of the moon and sun for NDAYS days. TDEL4 array order is

\[ \delta^4 x_{a_1}, \delta^4 y_{a_1}, \delta^4 z_{a_1}, \delta^4 x_{\odot 1}, \delta^4 y_{\odot 1} \]

\[ \delta^4 z_{\odot 1}, \ldots, \delta^4 x_{\text{NDAYS}}', \delta^4 y_{\text{NDAYS}}' \]

Units of earth radii—mean of 1950.

**/EPHCOM/**

**XJD**

A 10-cell vector containing NDAYS Julian dates. Each Julian date is input mod 2,430,000.0. XJD array order is

\[ \text{JD}_1, \text{JD}_2, \text{JD}_3, \ldots, \text{JD}_{\text{NDAYS}} \]

**BIJ**

Non-zero element of constraint matrix

**XIJ**

Subscripts for non-zero elements of constraint matrix, \( B_{ij} \).

**CI**

Additive constants for constraint problem

**ALTS**

Altitude table for multiple drag (kilometers).

**CLAMDA**

\( C_p A/m \) table corresponding to ALTS table (m²/kg).

**CATLM**

Array which flags the drag solution variables.

**CHEPS**

Tolerance criterion of altitude cut-offs (earth radii).

**ECRIT**

Minimum elevation to allow steering ephemeris print.
RDATA

DAYINT  Integer portion of Julian date on mean elements card.
DAYFRC  Fractional portion of Julian date on mean elements card.
RDFLG   Flag which changes output units of \( \mathbf{R} \) and \( \dot{\mathbf{R}} \).
SMELM   21-word vector containing the Smithsonian mean elements and their time derivatives.
TNULL   Time to which input elements are to be updated.

3. Calling sequence

C. Output

1. Blank COMMON
2. Labeled COMMON
3. Calling sequence

D. Error/action messages

1. Off-line comments

"NO. OF EPHEMERIS DAYS LESS THAN 4, TURN BODIES OFF"

2. On line comment

3. Action

If the number of lunar-solar ephemeris days (NDAYS) is greater than 0 and less than 4, the off-line comment is printed and NDAYS is set equal to 0, which in effect turns off computation of perturbative accelerations due to the moon and sun and/or radiation pressure.
SUBROUTINES USED

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
   SELECT

B. Program
   NRTPOD

C. Called by Subroutines
   TRJGEN

FUNCTION

To select the next output time for the trajectory package. This routine is used to select the next observation time during the curve fit portion of NRTPOD, and the next DELTT time for the print-update option.

USAGE

A. Calling sequence
   Call SELECT

B. Input
   1. COMMON
      TEPOCH       Epoch time, minutes from 0 hours
      DELTT        8 sets of Δt, T
      KONTRL       = 1 if curve fit in progress, = 2 if trajectory print-update
      TLIST        Integration list
      NDTCT        Counter for DELTT array to indicate next set to be processed
      NLAMS        Number of drag parameters in the solution vector
      NPXDLM       Starting location in VSTR of $d_i$ vectors, where
                     
                     \[ d_i = \frac{\partial \begin{pmatrix} x, \ y, \ z \end{pmatrix}}{\partial \begin{pmatrix} \frac{C_D A}{m} \end{pmatrix}} \text{ i = 1, ..., NLAM} \]
                      \[ i \]
                     
                     NLAM        Total number of entries in the altitude $C_D A/m$ table

5-163
NPALM
Starting location in VSTR of \( b_i \) vectors, where

\[
b_i = \left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(ADBARV)} \right]_{t_0}^{-1}
\]

\[
\left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(\frac{C_D A}{m})} \right]_i
\]

2. Calling sequence

C. Output

TG  The time of the next output, minutes from 0
     hours day of epoch

PUBS The next observation (if KONTRL = 1)

TUBSEF Non-zero if the end of the observation tape
       has been sensed (if KONTRL = 1)

D. Error/action messages

E. Internal Storage

1. COMMON

TMINUS This flag is used when there are pre-
     epoch times to be processed. When
     the first pre-epoch time is encoun-
     tered this flag is set to 1 and the
     integration is initialized in the back-
     ward time direction. When the first
     post-epoch time is encountered, re-
     initialization of the integration at
     epoch will take place if TMINUS is
     set to 1. Initially, TMINUS is
     assumed 0.

NDTCT Incremented internally
SUBROUTINES USED

A. Library

B. Program
   UBSGET
   SETIC

Processes observation tape
Initializes integration list at epoch
SUBROUTINE IDENTIFICATION

A. Title
SENRD

B. Segment
NRTPOD - Input processor

C. Called by subroutine
LODSEN

FUNCTION

Function is to read the sensor cards (6 types) and to build a temporary buffer (DTMP) for biases and weights by station.

USAGE

A. Calling sequence
CALL SENRD (SEOF)

B. Input
1. Blank COMMON
   KIN
   KOUT
   PREFLG  Symbolic input tape
   Symbolic output tape
   NRTPOD control flags (col 31 - 40 on JDC)

2. Labeled COMMON
   /TEMP/
   TEMP  Internal temporary storage

3. Calling sequence

C. Output
1. Labeled COMMON
   /INPP/
   NDTMP  Counter on DTMP buffer for biases and weights by station.
   DTMP (51)  Station ID
   (52)  Azimuth bias (deg)
   (53)  Elevation bias (deg)
   (54)  Range bias (km)
   (55)  Range bias (km/sec)
   (56)  Range acceleration bias (m/sec^2)
2. Calling sequence

**SEOF**

Flag indicating whether all sensor cards have been read.

= -1. More sensors to be read

= +1. END sensor card has been detected. No more sensor cards to be read.

D. Error/action messages

1. Off-line comment

"NO. OF SENSORS GREATER THAN MAX ALLOW. --- IGNORE."

2. On-line comment

3. Action

Ignores processing of previous sensor data, and proceeds to the next sensor card.

**SUBROUTINES USED**

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
SETCON

B. Segment
NRTPOD - Input Processor

C. Called by subroutine
INPUT

FUNCTION

To set up nominal values of program control constants, potential model constants, scale factors, and symbolic tape assignments.

USAGE

A. Calling sequence
CALL SETCON

B. Input

1. COMMON

2. Calling sequence

C. Output

1. Blank COMMON

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE</td>
<td>Earth's rotational rate (radians/min)</td>
</tr>
<tr>
<td>CELLIP</td>
<td>Ellipticity of the earth</td>
</tr>
<tr>
<td>CMU</td>
<td>GM of the earth (e.r.$^3$/min$^2$)</td>
</tr>
<tr>
<td>CGMR</td>
<td>GM ratios (MOON GM/EARTH GM, SUN GM/EARTH GM)</td>
</tr>
<tr>
<td>CFTER</td>
<td>ft/e.r.</td>
</tr>
<tr>
<td>CKMFT</td>
<td>km/ft</td>
</tr>
<tr>
<td>CKMER</td>
<td>km/e.r.</td>
</tr>
<tr>
<td>CMTER</td>
<td>meters/e.r.</td>
</tr>
<tr>
<td>CDEG</td>
<td>degrees/radian</td>
</tr>
<tr>
<td>CFTNM</td>
<td>ft/n mi</td>
</tr>
<tr>
<td>CNMER</td>
<td>n mi/earth radii</td>
</tr>
<tr>
<td>CDAYMN</td>
<td>12-cell array denoting the number of days in each month</td>
</tr>
<tr>
<td>CPI</td>
<td>$\pi$</td>
</tr>
<tr>
<td>C2PI</td>
<td>$2\pi$</td>
</tr>
</tbody>
</table>
SETCON

KOUT  Output tape number (print)
KIN   Input tape number
MT    Observations tape number
NOUT  Scratch tape not used at present by NRTPOD
ITRJTP Trajectory ephemeris tape number
CHMAX Maximum integration step size
CHMIN Minimum integration step size
CYMIN Parameter for variable step integration
CER   Parameter for variable step integration
CBE   \( b_e = 1.0 - \text{CELLIP} \)
CRASHE Impact flags used by subroutine TRAJ
CRASHM
CJD50  Julian date Jan 0, 1950
COMLST Size of variable storage
CFTEPS RMS convergence criterion
DTMAX  Editing parameter - maximum allowable observation time from epoch (days)
TSTEP  Initial integration step size (min)
BFLAGS Flags indicating bodies (moon and sun) to be considered
SKIP  If 0, always set FLVE = 0, if non-zero, set FLVE accordingly
CKRMS  A provision for editing residuals by input
CNSIG  N for N \( \cdot \sigma \) deletion
NRRR   Ratio of Runge-Kutta step to Cowell step
FLVE   If non-zero, skip VAREQ
CHEPS  Criterion on altitude cut-offs

2. Labeled COMMON

/INPP/

SECT  Array of cells used for callouts of the sectorial harmonics, non-zero to include the desired harmonic
CJ    Values of the coefficients of the zonal harmonics \( J_{2^*}, \ldots, J_{12^*} \)
ZONAL Array of cells used for callouts of the coefficients of the zonal harmonics
CLAMNN Array containing the angles associated with the coefficients of the tesseral harmonics
CJNM  6 x 6 array containing the coefficients of the sectorial and tesseral harmonics and their associated angles

3. Calling sequence

D. Error/action messages

5-170
SETCON

SUBROUTINES USED

A. Library
   -

B. Program
   -
SUBROUTINE IDENTIFICATION

A. Title
   SETIC

B. Segment
   NRTPOD

C. Called by Subroutines
   SELECT
   TRJGEN

FUNCTION

The function is to initialize the integration list and other parameters which must be re-initialized each time the integration is re-started.

USAGE

A. Calling sequence
   Call SETIC

B. Input
   1. COMMON
      NDPR
      TEPOCH
      TSTEP
      TICRT
      NLAM
      ALT
      
      Total number of Category 1 variables
      Minutes from midnight day of epoch
      Starting step size for the numerical integration in minutes
      x, y, z, x, y, z of the vehicle at epoch in earth radii and e.r./min
      Total number of entries in the altitude CoA/m table
      Two altitude layers bounding the current region of influence (e.r.)

C. Output
   1. COMMON
      TG
      TCRASH
      
      Time to integrate to (min)
      Impact flag
      = 0 not impacted
      ≠ 0 impact

5-173
TLIST        Numerical integration working storage
TMINUS       Flag indicating backward integration
PMAT}       Arrays used in variational equation
VMAT}       formulation, initialized at 0
FLVE         Flag for variational equations compata-

2. Calling sequence

D. Error/action messages

SUBROUTINES USED
A. Library

B. Program
DAUX
FVE
HEIGHT
HINT
VPERT
SUBROUTINE IDENTIFICATION

A. Title
   STEER

B. Segment
   NRTPOD

C. Called by subroutine
   TPRNT

FUNCTION

Function is to compute radar steering ephemeris for NRTPOD and print summary values.

USAGE

A. Calling sequence
   Call STEER

B. Input
   1. COMMON
      CBE   Semi-minor axis of earth, $b_e$
      CELLIP Ellipticity of the earth
      CKMER Value of km per earth radii
      CKMFT Conversion from ft to km
      CDEG Conversion from radians to degrees
      KOUT Current output unit
      NDPR Total number of CATI variables to solve for
      NSTAT Pointer to first station of master sensor table in VSTR
      NMSTAT Number of entries per station of master sensor table
      CDAD2M Drag parameter
      ECRIT Minimum elevation to allow printing
RDFLG

Flag which changes output units of RDOT, R2DOT (see equations) from \((\text{km/min})\) and \((\text{km/min}^2)\) to \((\text{km/sec})\) and \((\text{mt/sec}^2)\).

When RDFLG = 0 The output units of RDOT and R2DOT are \((\text{km/min})\) and \((\text{km/min}^2)\) respectively.

RDFLG ≠ 0 The output units of RDOT and R2DOT are \((\text{km/sec})\) and \((\text{mt/sec}^2)\) respectively.

TG

Time to integrate to

TRAJX

Output from TRAJ: \(x, y, z, \dot{x}, \dot{y}, \dot{z}\), etc.

TRHOA

Density, \(\text{kg/m}^3\)

PUBS

Observation vector ID, time, \(R, A, E, R\), type or ID, time, \(\dot{R}, 0, 0, 0\), type

PCMR

\(R = \text{computed slant range (c.r.)}\)

PSTAT

Working storage for current station in master sensor table

PUI

Vector \((u_1, u_2, u_3)\)

PVI

Vector \((v_1, v_2, v_3)\)

PWDTI

Vector \((\psi_1, \psi_2, \dot{\psi}_3)\)

PR2DOT

Second time derivative of range

2. Calling sequence

C. Output

1. COMMON

2. Calling sequence

D. Error/action messages

5-176
SUBROUTINES USED

A. Library

ASIN  Arcsin
ATNQ  Double argument arctan
SQRT  Square root

B. Program

DOT      Dot product
PIMOD    Angle moded 0 to 2π
PRELIM  Quantities of residuals and partial derivatives of observations
RADSQ   Find the square of the radius

EQUATIONS

Altitude  =  R - R_e = h

\[ h = R - \sqrt{b_e \left[ 1 - \epsilon(2 - \epsilon) \left( \frac{x^2 + y^2}{R^2} \right) \right]^{1/2}} \]

\[ R = \sqrt{q_1^2 + q_2^2 + q_3^2} \]  

(range)

\[ A = \tan^{-1} \frac{v_1}{v_2} \]  

(azimuth)

\[ E = \sin^{-1} v_3 \]  

(elevation)

\[ \dot{R} = \vec{u} \cdot \vec{W} \]  

(range rate)

\[ \ddot{R} = \vec{u} \cdot \vec{W} + \ddot{u} \cdot \vec{W} \]  

(range acceleration)

The vectors and vector components which are used to calculate the radar observables above are defined in subroutine PRELIM.
Subroutine STEER is called from TPRNT to calculate and print the radar steering ephemeris. The values printed are either obtained through COMMON or are calculated on a call to PRELIM. All the values are converted to the proper output units. The outputs are:

- **h**: Height of vehicle above the ellipsoid (km)
- **C_D A/m**: The ballistic coefficient (m^2/kg)
- **\( \rho \)**: Atmospheric density (kg/m^3)
- **R**: Slant range to vehicle (km)
- **AZ**: Azimuth of vehicle (degrees)
- **E**: Elevation of vehicle (degrees)
- **\( \dot{R} \)**: Time rate of change of range (km/min)
- **\( \ddot{R} \)**: Range acceleration of vehicle (km/min^2)

Range rate and range acceleration may be output in units of (km/sec) and (mt/sec^2) respectively if RDFLG is set non-zero.
SUBROUTINE IDENTIFICATION

A. Title
   TRAJ

B. Program
   NRTPOD

C. Called by Subroutine
   TRJGEN

FUNCTION

Integrate the equations of motion and up to 24 variational equations to a specified time. The routine uses Runge-Kutta as a starter to build eighth order difference tables for a Cowell method of numerical integration. The routine will automatically exit with a flag set to indicate earth impact.

USAGE

A. Calling sequence
   Call TRAJ(TN)

B. Input
   COMMON
      1. HMAX       Maximum allowable step size
      HMIN       Minimum allowable step size
      ER }       Step size test parameters;
      YMIN}       See method
   TLIST       Input and storage, at output values consistent with T
      CMU       GM of earth (Earth radii and minutes)
      CRASHB     Ellipticity of earth
      CRASHE     1 x 10^-8
      CRASHM     Altitude below which impact test will be made (earth radii)
      NDPR       The number of variational parameters in the integration list
      NRRR       Non-zero if fixed step Runge-Kutta desired
      SKIP       If 0, evaluate variational equations only on "predictor" steps
2. Calling sequence
TN Time to integrate to (Minutes from epoch)

C. Output
1. COMMON
   TRAJX(1-3) x, y, z Output ... consistent with
               TN or impact time
   TRAJX(4-6) \dot{x}, \dot{y}, \dot{z}
   TRAJX(7-9) \dddot{x}, \dddot{y}, \dddot{z}
   TRAJX(10-15) \ddot{x}, \ddot{y}, \ddot{z}, \dot{x}, \dot{y}, \dot{z} first
                 variation
   TRAJX(16-21) \dddot{x}, \dddot{y}, \dddot{z}, \ddot{x}, \ddot{y}, \ddot{z} second
                 variation
   ...
   TRAJX(52-57) \ddots \dddot{x}, \ddots \ddot{y}, \ddots \ddot{z}, \ddots \dot{x}, \ddots \dot{y}, \ddots \dot{z} eighth
                 variation
   TCRASH Set non-zero if impact occurs
   FLVE Non-zero to indicate predictor steps
   PR2DPI 8 x 3 array of partial derivatives of
           accelerations with respect to ADBARV
           and the two drag layers of current
           influence (\lambda_i and \lambda_i + 1)

2. Calling sequence

SUBROUTINES USED
   A. Program
      DAUX

COMMENTS
The integration list must be initialized before calling TRAJ. If
impact occurs, the output is at the impact time, not TN. The initializa-
tion flag set non-zero externally, is returned zero by TRAJ.

5-180
# COMMON (TLIST) Storage

<table>
<thead>
<tr>
<th>TLIST</th>
<th>Program Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FLAG</td>
<td>Initialization parameter—initialize when nonzero</td>
</tr>
<tr>
<td>2</td>
<td>T</td>
<td>Current time</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>Current step size</td>
</tr>
<tr>
<td>4-30</td>
<td>Y(1-27)</td>
<td>$y_1, y_2, \ldots, y_n$</td>
</tr>
<tr>
<td>31-57</td>
<td>YP(1-27)</td>
<td>$\dot{y}_1, \dot{y}_2, \ldots, \dot{y}_n$</td>
</tr>
<tr>
<td>58-84</td>
<td>YPP(1-27)</td>
<td>$\ddot{y}_1, \ddot{y}_2, \ldots, \ddot{y}_n$ DAUX stores</td>
</tr>
<tr>
<td>193-489</td>
<td>DIF</td>
<td>Difference table During Runge-Kutta phase</td>
</tr>
</tbody>
</table>

- $(1, 1-27)$ $V^8_{f_1}$ See method for description of this table. $\dddot{y}_{i4}$ These values are saved during 8NR Runge-Kutta steps. |
- $(2, 1-27)$ $V^7_{f_1}$ as $I = 1, N$ $\dddot{y}_{i5}$ as $I = 1, N$ |
- $(3, 1-27)$ $V^6_{f_1}$ $\dddot{y}_{i2}$ |
- $(4, 1-27)$ $V^5_{f_1}$ $\dddot{y}_{i3}$ |
- $(5, 1-27)$ $V^4_{f_1}$ $\dddot{y}_{i4}$ |
- $(6, 1-27)$ $V^3_{f_1}$ $\dddot{y}_{i5}$ |
- $(7, 1-27)$ $V^2_{f_1}$ $\dddot{y}_{i6}$ |
- $(8, 1-27)$ $V^1_{f_1}$ $\dddot{y}_{i7}$ |
- $(9, 1-27)$ $f_i = \dot{y}$ $\dddot{y}_{i8}$ |
- $(10, 1-27)$ $'F_i$ $\dddot{y}_{i4}$ |
- $(11, 1-27)$ $''F_i$ $\dddot{y}_{i4}$ |
INITIALIZE
N = 3(NDPR + 1)
BSO = (CRA5HC)²
TDP = T
HDP = H
KC = 1
FLAG = 0

STORE INITIAL POSITION, VELOCITY, AND ACCELERATION
YDP(I) = Y(I)
YPDP(I) = YP(I)
DIF(1,I) = YPP(I)
I = 1, N

SELECT INITIAL STEP SIZE USING RK
FLVE = 1.0

IF
|DPA| : |HDP|

SET HALVING AND DOUBLING PARAMETERS
VMAX = ER x 10⁴/1.5
VMIN = VMAX x 10⁻⁴

Figure 5-13. TRAJ Flow Diagram

5-182
TRAJ

Figure 5-13. TRAJ Flow Diagram (Continued)

5-183
\[ A = \max (Y(1), \ Y_{\min}) \]
\[ V = \max \left( \sum_{i=1}^{7} \frac{f_i}{A}, \sum_{i=1}^{8} \frac{f_i}{A} \right) \]
\[ W = \max \left( \sum_{i=1}^{7} \frac{f_i}{A}, \sum_{i=1}^{8} \frac{f_i}{A} \right) \]

\( i = 1, 3 \)
\[ \text{DIF} (1, 1) = \sum_{i=1}^{7} f_i \]
\[ \text{DIF} (2, 1) = \sum_{i=1}^{8} f_i \]

\( \text{KS} = \frac{\text{KS}}{2} \)
\( \text{DPA} = \frac{\text{HDP}}{2} \)

\( \text{KS} = 4 \)

\( \text{HDP} = \text{DPA} \)

Figure 5-13. TRAJ Flow Diagram (Continued)
TRAJ

SAVE PRESENT VALUES
TS = TDP
HS = HDP
TR(I,1) = Y(I)
TR(I,1) = YP(I)
I = 1, N

KRTN = 2
HDP = DPA
H = DPA

FORM RATIO OF OUTPUT STEP TO INTEGRATION STEP
A = HDP/HS

COMPUTE \( \sigma' \) AND \( Y' \) AS FUNCTIONS OF A

COMPUTE \( Y_{i-1}(o) \) AS A FUNCTION OF A AND THE DIFFERENCE TABLE
\( YPP(I) = Y_{i-1}(o) \)
I = 1, N

EVALUATE PREDICTOR FORMULAS USING \( \sigma' \) AND \( Y' \)
\( Y(I) = Y(I)(o) \)
\( YP(I) = Y(I)(p) \)
\( YP(I) = Y(I)(p) \)
I = 1, N

EXIT

Figure 5-13. TRAJ Flow Diagram (Continued)

5-185
Figure 5-13. TRAJ Flow Diagram (Continued)
SUBROUTINE IDENTIFICATION

A. Title
   TRJGEN

B. Segment
   NRTPOD

C. Called by Subroutine
   TRJGEN

FUNCTION

Driver for the trajectory link. Controls the logic associated with the trajectory integration and the generation of the trajectory tape.

USAGE

A. Calling sequence
   Call TRJGEN

B. Input

1. COMMON

   KOUT       Output tape (print)
   TEPOCH     Time of epoch, minutes from 0 hours
   DTMAX      Maximum allowable time interval for an observation - in days since epoch
   TUBSEF     Flag denoting when the last observation has been processed from tape.
               Set ≠ 0 when "end of file" encountered.
   PLSTSN     Station ID for previous observation
   TG         Integration time to go. Minutes from 0 hours, day of epoch
   TCRASH     Flag indicating earth impact. Non-zero if impact has occurred
   KONTRL     Flag indicating mode of NRTPOD
               KONTRL = 1   Execute TRJGEN for curve fit and trajectory
               KONTRL = 2   Execute TRJGEN for trajectory only

2. Calling sequence
   

5-187
C. Output
1. COMMON

2. Calling sequence

D. Error/action messages
1. Action messages

"START TRAJECTORY"

and

"END TRAJECTORY"

Occur when the program begins executing the trajectory link
and when execution of the trajectory link terminates

SUBROUTINES USED
A. Library

B. Program

SETIC     Initializes integration lists
SELECT    Selects next observation
PARSET    Sets up the PSTAT sensor information
           array from master sensor table for
current observation
TRAJ      Integration subroutine
TRJTAP    Writes trajectory tape
FALSI     Determines altitude cut-offs
PLAMDA    Computes the required partial derivatives
           of position, velocity, and acceleration
           with respect to drag
TRJOUT    Prepares a variable length trajectory
           record to be written on the trajectory tape
TRJGET

SUBROUTINE IDENTIFICATION

A. Title
   TRJGET

B. Segment
   NRTPOD

C. Called by subroutine
   DCITER

FUNCTION

TRJGET reads one trajectory record from the trajectory tape and, if necessary, sets the impact control flag.

USAGE

A. Calling sequence
   Call TRJGET (TG)

B. Input
   1. COMMON
      ITRJTP     Trajectory tape number
      TEPOCH    Minutes from midnight day of epoch to epoch
   2. Calling sequence
      TG        Observation time for which a corresponding trajectory record is to be read
   3. Tape input
      The trajectory tape generated by the trajectory segment

C. Output
   1. COMMON
      A(I)     Variable length trajectory array
                 I = 1, ..., N

where

A (1-9)     x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}
(10 - 15) \[ \frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha} \]

. . . .

. . . .

. . . .

(40 - 45) \[ \frac{\partial x}{\partial \nu}, \frac{\partial y}{\partial \nu}, \frac{\partial z}{\partial \nu}, \frac{\partial x}{\partial \nu}, \frac{\partial y}{\partial \nu}, \frac{\partial z}{\partial \nu} \]

\[ \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1} \]

(46 to 6*NDPR + 9)

\[ \frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}} \]

\[ \frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}} \]

(6* NDPR + 10 to 9*NDPR + 9)

\[ \frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha} \]

. . .

. . .

. . .

\[ \frac{\partial x}{\partial \nu}, \frac{\partial y}{\partial \nu}, \frac{\partial z}{\partial \nu} \]

\[ \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1} \]

. . .

. . .

. . .

\[ \frac{\partial x}{\partial \lambda_{NLAMS}}, \frac{\partial y}{\partial \lambda_{NLAMS}}, \frac{\partial z}{\partial \lambda_{NLAMS}} \]

and NLAMS is the total number of \( \frac{C_D A}{m} \)'s appearing in the solution vector.
TRJGET

= -1, if impact is pre-epoch

TCRASH

= 1, if impact is post-epoch

= 0, if no impact

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library
   .FBLT.
   .FRDB.
   .FVIO.
SUBROUTINE IDENTIFICATION

A. Title
   TRJOUT

B. Segment
   NRTPOD

C. Called by subroutines
   PLAMDA
   TRJGEN

FUNCTION

Prepares a variable length trajectory tape record to be written on the trajectory tape at each observation and altitude cut-off point.

USAGE

A. Calling sequence
   Call TRJOUT

B. Input

1. COMMON

   VSTR

   Variable storage

   PR2DPI (1-3) $\frac{\partial \xi}{\partial \alpha}$, $\frac{\partial \eta}{\partial \alpha}$, $\frac{\partial \zeta}{\partial \alpha}$

   (4-6) $\frac{\partial \xi}{\partial \delta}$, $\frac{\partial \eta}{\partial \delta}$, $\frac{\partial \zeta}{\partial \delta}$

   .

   (16-18) $\frac{\partial \xi}{\partial \nu}$, $\frac{\partial \eta}{\partial \nu}$, $\frac{\partial \zeta}{\partial \nu}$

   (19-21) $\frac{\partial \xi}{\partial \lambda_j}$, $\frac{\partial \eta}{\partial \lambda_j}$, $\frac{\partial \zeta}{\partial \lambda_j}$

   (22-24) $\frac{\partial \xi}{\partial \lambda_{j+1}}$, $\frac{\partial \eta}{\partial \lambda_{j+1}}$, $\frac{\partial \zeta}{\partial \lambda_{j+1}}$

   Where $\lambda_j$, $\lambda_{j+1}$ are the current (CDA/m) drag parameters in the region of influence
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLAMS</td>
<td>Number of $C_{DA}/m$ drag parameters in the solution vector</td>
</tr>
<tr>
<td>NLID</td>
<td>Pointer to location in variable storage where the identifiers for the $(C_{DA}/m)$ drag parameters appearing in the solution vector are stored</td>
</tr>
<tr>
<td>NPXLM</td>
<td>Pointer to location in variable storage where the $\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial \left( \frac{C_{DA}}{m} \right)_{i}}$ are stored if $i = 1, \ldots, NLAM$</td>
</tr>
<tr>
<td>NLAM</td>
<td>Total number of entries in the altitude $C_{DA}/m$ table</td>
</tr>
<tr>
<td>NPXDLM</td>
<td>Pointer to location in variable storage where the $\frac{\partial(\dot{x}, \dot{y}, \dot{z})}{\partial \left( \frac{C_{DA}}{m} \right)_{i}}$ are stored if $i = 1, \ldots, NLAM$</td>
</tr>
<tr>
<td>NDPR</td>
<td>Total number of CAT1 variables ($\alpha \delta \beta$ARV and $C_{DA}/m$'s) in the solution vector</td>
</tr>
<tr>
<td>TRAJX</td>
<td>Integration coordinates referenced to some time, $t$.</td>
</tr>
<tr>
<td></td>
<td>(1-9) $x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}$</td>
</tr>
<tr>
<td></td>
<td>(10-15) $\frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial \dot{x}}{\partial \alpha}, \frac{\partial \dot{y}}{\partial \alpha}, \frac{\partial \dot{z}}{\partial \alpha}, \frac{\partial \ddot{x}}{\partial \alpha}, \frac{\partial \ddot{y}}{\partial \alpha}, \frac{\partial \ddot{z}}{\partial \alpha}$</td>
</tr>
<tr>
<td></td>
<td>(40-45) $\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}, \frac{\partial \dot{x}}{\partial v}, \frac{\partial \dot{y}}{\partial v}, \frac{\partial \dot{z}}{\partial v}, \frac{\partial \ddot{x}}{\partial v}, \frac{\partial \ddot{y}}{\partial v}, \frac{\partial \ddot{z}}{\partial v}$</td>
</tr>
<tr>
<td>CFTER</td>
<td>Conversion constant (feet/earth radii)</td>
</tr>
<tr>
<td>TLIST</td>
<td>Trajectory integration list (See TRAJ subroutine)</td>
</tr>
</tbody>
</table>

5-194
NH1   Pointer to location in variable storage of the 1st altitude layer bounding the current region of influence

NH2   Pointer to location in variable storage of the 2nd altitude layer bounding the current region of influence

2. Calling sequence

C. Output

1. COMMON

A   Variable length trajectory array to be written on trajectory tape
A(1-9)  x, y, z, x, y, z, x, y, z
A(10-15)  \( \frac{\partial x}{\partial \alpha} \), \( \frac{\partial y}{\partial \alpha} \), \( \frac{\partial z}{\partial \alpha} \), \( \frac{\partial x}{\partial \alpha'} \), \( \frac{\partial y}{\partial \alpha'} \), \( \frac{\partial z}{\partial \alpha'} \)
A(40-45)  \( \frac{\partial x}{\partial \nu} \), \( \frac{\partial y}{\partial \nu} \), \( \frac{\partial z}{\partial \nu} \)
A(46-51)  \( \frac{\partial x}{\partial \lambda_{i}} \), \( \frac{\partial y}{\partial \lambda_{i}} \), \( \frac{\partial z}{\partial \lambda_{i}} \), \( \frac{\partial x}{\partial \lambda_{i}} \), \( \frac{\partial y}{\partial \lambda_{i}} \), \( \frac{\partial z}{\partial \lambda_{i}} \)
A(6*NDPR+4 to 6*NDPR+9)  Present only if \( \lambda \)'s are present in the solution vector

\( \frac{\partial x}{\partial \lambda_{NLAMS}} \), \( \frac{\partial y}{\partial \lambda_{NLAMS}} \), \( \frac{\partial z}{\partial \lambda_{NLAMS}} \)
\( \frac{\partial x}{\partial \lambda_{NLAMS}} \), \( \frac{\partial y}{\partial \lambda_{NLAMS}} \), \( \frac{\partial z}{\partial \lambda_{NLAMS}} \)
\( \frac{\partial x}{\partial \alpha} \), \( \frac{\partial y}{\partial \alpha} \), \( \frac{\partial z}{\partial \alpha} \)
\( \frac{\partial x}{\partial \nu} \), \( \frac{\partial y}{\partial \nu} \), \( \frac{\partial z}{\partial \nu} \)
If no drag parameters $CDA/m$ are to be solved for, the partials of position, velocity, and acceleration wrt $\lambda$ are not placed into the $A$ array.

$CDAD2M$  
Current $\left(\frac{CDA}{2m}\right)$ (ft$^2$/slug) in internal units

$TALT$  
Current altitude of vehicle (ft.)

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

$TRJTAP$  
Writes one record of binary trajectory information on the trajectory tape
SUBROUTINE IDENTIFICATION

A. Title
   TRJPRO

B. Segment
   NRTPOD

C. Called by Subroutine
   NRTPOD

FUNCTION

Main driver controlling the coordination of all activities involving the trajectory segment, curve fit segment, and the trajectory print and update segment.

USAGE

A. Calling sequence
   Call TRJPRO

B. Input

1. COMMON
   KONTRL
   DCFLG
   IFTEX
   PSTFLG
   PR2DPI

   Flag indicating mode of NRTPOD
   KONTRL = 1 Curve fit and trajectory
         = 2 Trajectory only
   JDC card options (card column 41-50)
   Exit flag from subroutine FIT
   IFTEX = 1 Solution has converged
           = 2 Maximum iterations exceeded and converging
           = 3 Failed K BOUNDS/8
           = 4 Normal return
           = 5 Maximum iterations exceeded and converging
   JDC options (card columns 51-60)
   8 x 3 array of partial derivatives of accelerations with respect to ADBARV
   and the two drag layers of current influence
C. Output
   1. COMMON
      
      2. Calling sequence
         
D. Error/action messages

**SUBROUTINES USED**

<table>
<thead>
<tr>
<th>A. Library</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPRLM</td>
<td>Performs necessary initialization prior to a differential correction pass</td>
</tr>
<tr>
<td>TRJGEN</td>
<td>Driver for the trajectory segment; generates the trajectory tape</td>
</tr>
<tr>
<td>DCITER</td>
<td>Driver for the curve fit segment</td>
</tr>
<tr>
<td>PRUDRV</td>
<td>Trajectory print and update driver</td>
</tr>
</tbody>
</table>
SUBROUTINE IDENTIFICATION

A. Title
   TRJTAP

B. Segment
   NRTPOD

C. Called by subroutine
   TRJGEN

FUNCTION

Function is to write the trajectory tape used by the curve fit and trajectory print and update segments.

USAGE

A. Calling sequence
   Call TRJTAP (IOPT)

B. Input
   1. COMMON
      ITRJTP   Trajectory tape number
      TG      Integration time to go ... minutes from 0 hours day of epoch
      TRAJX   Integration coordinates at time TG: position, velocity, acceleration, partials of position and velocity w. r. t. the category 1 variables
      N      Total number of words in the A array to be written
      TG      Trajectory time that the trajectory tape record is referenced to
      TCRASH  Impact flag
               = 0   vehicle has not impacted
               ≠ 0   vehicle has impacted
2. Calling sequence

IOPT

Flag indicating type of trajectory record written on trajectory tape

IOPT = 1  Writes a standard data record

= 2  Writes a pseudo "end of file" record

3. Trajectory tape

A(I)

Variable length trajectory array

I = 1, ..., N

where

A (1-9) x, y, z, x, y, z, x, y, z

(10-15) \frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}, \frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}

\vdots

\vdots

(40-45) \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}, \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}
\[
\frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1},
\]
\[
\vdots \\
\vdots \\
\vdots \\
\]
\[
\frac{\partial x}{\partial \text{NLAMS}}, \frac{\partial y}{\partial \text{NLAMS}}, \frac{\partial z}{\partial \text{NLAMS}},
\]
\[
(46 \text{ to } 6 \times \text{NDPR}+9)
\]
\[
\frac{\partial x}{\partial \text{NLAMS}}, \frac{\partial y}{\partial \text{NLAMS}}, \frac{\partial z}{\partial \text{NLAMS}}
\]
\[
(6 \times \text{NDPR}+10 \text{ to } 9 \times \text{NDPR}+9)
\]
\[
\frac{\partial x}{\partial \alpha}, \frac{\partial y}{\partial \alpha}, \frac{\partial z}{\partial \alpha}
\]
\[
\vdots \\
\vdots \\
\vdots \\
\]
\[
\frac{\partial x}{\partial \nu}, \frac{\partial y}{\partial \nu}, \frac{\partial z}{\partial \nu}
\]
\[
\frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}
\]
\[
\vdots \\
\vdots \\
\vdots \\
\]
\[
\frac{\partial x}{\partial \text{NLAMS}}, \frac{\partial y}{\partial \text{NLAMS}}, \frac{\partial z}{\partial \text{NLAMS}}
\]
and NLAMS is the total number of \(CpA/m\)'s appearing in the solution vector.

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program

5-201
Figure 5-14. TRJTAP Flow Diagram
SUBROUTINE IDENTIFICATION

A. Title
UPPER

B. Segment
NRTPOD

C. Called by subroutine
PRAUPD

FUNCTION

UPPER converts an N x N matrix stored lower triangular by rows into an N x N matrix stored upper triangular by rows with an augmented column. (This matrix with the augmented column is required as an input to subroutine LEGS2.)

USAGE

A. Calling sequence
Call UPPER (B, N)

B. Input
1. COMMON
   NATA Starting location in variable storage of the accumulated normal matrix.
2. Calling sequence
   B Input lower triangular matrix stored by rows
   N Dimension of matrix [B] is N x N

C. Output
1. COMMON
   VSTR(NATA) Output upper triangular matrix with an augmented column stored by rows
2. Calling sequence

D. Error/action messages

5-203
SUBROUTINES USED

A. Library

B. Program
6. NRTPD2 STORAGE MAP

6.1 NEW/REVISED COMMON ARRAYS

The following labeled COMMON blocks have been introduced or revised.

1) COMMON/SIGBUF/SBUF(400)

This COMMON array is used in the differential correction link as a buffer for the computed functional standard deviations. These standard deviations are computed in parallel with the observation residuals, are buffered in the SBUF array, and printed a page at a time.

2) COMMON/PLSS/PWPP(63), PWDTPP (63), PWDT2P (63)

The PWPP, PWDTPP, PWDT2P arrays have been taken out of PLS labeled COMMON and placed into PLSS labeled COMMON. These arrays have been extended from 24 cells to 63 cells respectively. (See Reference 2, Page 2-45.)

\[ PWPP(1 - 3) = \frac{\partial w_1}{\partial p_1}, \frac{\partial w_2}{\partial p_1}, \frac{\partial w_3}{\partial p_1} \]

\[ (4 - 6) = \frac{\partial w_1}{\partial p_2}, \frac{\partial w_2}{\partial p_2}, \frac{\partial w_3}{\partial p_2} \]

\[ \ldots \]

\[ PWPP[3 + 3(n - 1)] = \frac{\partial w_1}{\partial p_n}, \frac{\partial w_2}{\partial p_n}, \frac{\partial w_3}{\partial p_n} \]

where

\[ PWPP[3 + 3(n - 1)] = \frac{\partial w_3}{\partial p_n} \]

and \( n \) is the number of parameters \( p \) to be solved for from the list \((\phi_0, \delta_0, B_0, A_0, R_0, V_0, \lambda_1, \lambda_2, \ldots, \lambda_15)\), and \( \mathbf{W} = (w_1, w_2, w_3) \) is the geocentric position vector of the vehicle in a station meridian equatorial system.
\[
\begin{align*}
\text{PWDTPP}(1, 2, 3) &= \frac{\partial \dot{w}_1}{\partial p_1}, \frac{\partial \dot{w}_2}{\partial p_1}, \frac{\partial \dot{w}_3}{\partial p_1} \\
&\quad \vdots \quad \vdots \\
&\quad \vdots \\
&\quad \vdots \\
\text{PWDTPP}[3 + 3(n - 1)] &= \frac{\partial \dot{w}_1}{\partial p_n}, \frac{\partial \dot{w}_2}{\partial p_n}, \frac{\partial \dot{w}_3}{\partial p_n}
\end{align*}
\]

where \( n \) is defined as above and \( \ddot{\mathbf{w}} = (\ddot{w}_1, \ddot{w}_2, \ddot{w}_3) \) is the geocentric earth fixed velocity vector of the vehicle in a station meridian equatorial system.

\[
\begin{align*}
\text{PWDT2P}(1, 2, 3) &= \frac{\partial \ddot{w}_1}{\partial p_1}, \frac{\partial \ddot{w}_2}{\partial p_1}, \frac{\partial \ddot{w}_3}{\partial p_1} \\
&\quad \vdots \quad \vdots \\
&\quad \vdots \\
&\quad \vdots \\
\text{PWDT2P}[3 + 3(n - 1)] &= \frac{\partial \ddot{w}_1}{\partial p_n}, \frac{\partial \ddot{w}_2}{\partial p_n}, \frac{\partial \ddot{w}_3}{\partial p_n}
\end{align*}
\]

where \( n \) is defined as above and \( \dddot{\mathbf{w}} = (\dddot{w}_1, \dddot{w}_2, \dddot{w}_3) \) is the geocentric earth fixed acceleration of the vehicle in a station meridian equatorial system.

3) COMMON/PLS/PLS(150)

Although the PLS labeled COMMON region has been increased in size from 125 to 150, no new cells were introduced leaving cells PLS(117) to PLS(150) inclusive, unused in the differential correction link.

4) COMMON/SMATRIX/SMAT(630)

The SMAT array has been removed from DATA COMMON storage (see Page 2-48, Reference 2), and placed into a labeled COMMON block, SMATRIX. This array now allows storage for a 35 x 35 upper triangular by rows a priori normal matrix (A^T A). This storage was extended from a maximum allowable 20 x 20 to accommodate up to 15 drag layers in the solution vector.
5) **COMMON/UMATRIX/UPMAT(231)**

The UPMAT array has been removed from DATA storage (see Page 2-50, Reference 2), and placed into labeled COMMON block, UMATRX. This array now allows storage for up to a 21 x 21 lower triangular covariance matrix stored by rows. This storage was extended from a maximum allowable 7 x 7 to accommodate up to 15 drag layers plus 6 initial conditions in the a priori covariance matrix.

6) **COMMON/INPP/DTMP(690), DATA(1250)**

Array DTMP has increased from 300 to 690 to accommodate up to 20 sensors of information. DTMP cells (1-50) contain the same information as before (see Page 2-46, Reference 2). DTMP cells (51-690) contain the following information per each sensor:

<table>
<thead>
<tr>
<th>DTMP (51)</th>
<th>Station ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>(52)</td>
<td>Range bias (km)</td>
</tr>
<tr>
<td>(53)</td>
<td>Azimuth bias (deg)</td>
</tr>
<tr>
<td>(54)</td>
<td>Elevation bias (deg)</td>
</tr>
<tr>
<td>(55)</td>
<td>Range rate bias (km/sec)</td>
</tr>
<tr>
<td>(56)</td>
<td>Range acceleration bias (m/sec^2)</td>
</tr>
<tr>
<td>(57)</td>
<td>Time bias (sec)</td>
</tr>
<tr>
<td>(58)</td>
<td>( \sigma_R ) standard deviation on range (km)</td>
</tr>
<tr>
<td>(59)</td>
<td>( \sigma_A ) standard deviation on azimuth (deg)</td>
</tr>
<tr>
<td>(60)</td>
<td>( \sigma_E ) standard deviation on elevation (deg)</td>
</tr>
<tr>
<td>(61)</td>
<td>( \sigma_R^2 ) standard deviation range rate (km/sec)</td>
</tr>
<tr>
<td>(62)</td>
<td>( \sigma_R^2 ) standard deviation range acceleration (m/sec^2)</td>
</tr>
<tr>
<td>(63) [ \text{Sensor data input associated with the functional standard deviation option. See Section 2.2.1} ]</td>
<td></td>
</tr>
<tr>
<td>(64)</td>
<td>( B_R )</td>
</tr>
<tr>
<td>(65)</td>
<td>( B_A )</td>
</tr>
<tr>
<td>(66)</td>
<td>( B_E )</td>
</tr>
<tr>
<td>(67)</td>
<td>( B_R )</td>
</tr>
<tr>
<td>(68)</td>
<td>( \theta_1 )</td>
</tr>
<tr>
<td>(69)</td>
<td>( \theta_2 )</td>
</tr>
<tr>
<td>(70)</td>
<td>( \theta_3 )</td>
</tr>
<tr>
<td>(71)</td>
<td>( \theta_4 )</td>
</tr>
<tr>
<td>(72)</td>
<td>( \theta_5 )</td>
</tr>
<tr>
<td>(73)</td>
<td>( \theta_6 )</td>
</tr>
<tr>
<td>(74)</td>
<td>( \theta_7 )</td>
</tr>
<tr>
<td>(75)</td>
<td>( f(\theta_1) )</td>
</tr>
<tr>
<td>(76)</td>
<td>( f(\theta_2) )</td>
</tr>
<tr>
<td>(77)</td>
<td>( f(\theta_3) )</td>
</tr>
<tr>
<td>(78)</td>
<td>( f(\theta_4) )</td>
</tr>
<tr>
<td>(79)</td>
<td>( f(\theta_5) )</td>
</tr>
<tr>
<td>(80)</td>
<td>( f(\theta_6) )</td>
</tr>
<tr>
<td>(81)</td>
<td>( f(\theta_7) )</td>
</tr>
<tr>
<td>(82)</td>
<td>blank</td>
</tr>
</tbody>
</table>
The above information associated with each sensor follows in the DTMP array. Since 20 sensors are allowed and 32 cells are reserved in the DTMP array for each sensor, 690 cells have been allotted to DTMP.

The new cells added to DATA storage are in Section 6.2.

7) COMMON/PRDD/PR2DPI(24)

The PR2DPI array has been introduced to contain the

$$\frac{\partial r}{\partial (CAT 1)}$$

partial derivatives. These partial derivatives are computed at each integration step and are stored in the following manner

$$\text{PR2DPI}(1 - 3) = \frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial \theta}, \frac{\partial z}{\partial \theta}$$

$$(4 - 6) = \frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial \theta}, \frac{\partial z}{\partial \theta}$$

$$\ldots \ldots \ldots$$

$$(16 - 18) = \frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial \theta}, \frac{\partial z}{\partial \theta}$$

where $\lambda_i, \lambda_{i+1}$ are the current (CDA/m) drag parameters bounding the region of influence

$$(19 - 21) = \frac{\partial x}{\partial \lambda_i}, \frac{\partial y}{\partial \lambda_i}, \frac{\partial z}{\partial \lambda_i}$$

$$(22 - 24) = \frac{\partial x}{\partial \lambda_{i+1}}, \frac{\partial y}{\partial \lambda_{i+1}}, \frac{\partial z}{\partial \lambda_{i+1}}$$

8) COMMON/TRJX/TRAJX(198) or A(198)

The arrays TRAJX and A above are identical arrays with different names. The A array is used in the subroutines associated with the writing of the trajectory tape and the TRAJX array is used elsewhere. (See subroutine TRJOUT, Section 5.4 for a detailed description of this array).
This labeled COMMON block was added to accommodate the extended partial derivative arrays (see subroutine TRJOUT) associated with the additional 15 drag layers which may appear in the solution vector.

9) COMMON/TDPD/TDPDX(441), FINK(441), DINK(441), RINK(441)

This labeled COMMON block was introduced in the "UPDATE" link to be used as temporary working storage. The arrays defined by TDPD contain at different times during the "Updating," the partials of polar coordinates with respect to cartesian coordinates; the sigma and rho matrix (polar); the normal matrix (polar); the cartesian covariance matrix; etc.

6.2 NEW CELLS ADDED TO DATA STORAGE

<table>
<thead>
<tr>
<th>Name</th>
<th>Equivalence</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIJ</td>
<td>954</td>
<td>100</td>
<td>bij constraint matrix</td>
</tr>
<tr>
<td>XIJ</td>
<td>1054</td>
<td>100</td>
<td>i* 100 + j for bij</td>
</tr>
<tr>
<td>CI</td>
<td>1154</td>
<td>30</td>
<td>additive constants for linear constraints</td>
</tr>
<tr>
<td>NMDTMP</td>
<td>1184</td>
<td>1</td>
<td>Number of cells per sensor in DTMP list.</td>
</tr>
<tr>
<td>ALTS</td>
<td>1185</td>
<td>15</td>
<td>Altitude table for multiple drag (kilometers)</td>
</tr>
<tr>
<td>CLAMDA</td>
<td>1200</td>
<td>15</td>
<td>C_D/A/M table corresponding to ALTS table (meters^2/kilogram).</td>
</tr>
<tr>
<td>CATLM</td>
<td>1215</td>
<td>15</td>
<td>The CATLM array indicates to the program the CLAMDA variables to be solved for. This array must contain either &quot;ones&quot; or &quot;zeros,&quot; a 1 indicating the corresponding variable is to be solved for. For example, to solve for the second, fourth, and fifth CLAMDA, the ALTS-CLAMDA arrays must contain at least 5 entries, and the CATLM array must be input: CATLM = 0, 1, 0, 1, 1,</td>
</tr>
</tbody>
</table>

See Section 4, Operational Considerations, in Reference 8.
6.3 BLANK COMMON

1) Array BLK 1 additions:

<table>
<thead>
<tr>
<th>Name</th>
<th>BLK1( )</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHEPS  | BLK1(40)| 1         | Epsilon on altitude cut-offs. An altitude layer is accepted as a cut-off point when the computed cut-off altitude, $H_c$, is within CHEPS of the input cut-off altitude, $H_l$. Or the computed cut-off altitude, $H_c$, is accepted when the following condition exists:  

$$|H_c - H_l| \leq \text{CHEPS}$$

CHEPS is nominally set at $10^{-6}$ earth radii.

2) Array BLK2 has been increased from 30 to 50 cells. The following variables have been added:

<table>
<thead>
<tr>
<th>Name</th>
<th>BLK2( )</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPR</td>
<td>23</td>
<td>1</td>
<td>Size of the constrained system (when using linear constraints).</td>
</tr>
<tr>
<td>MBNDS</td>
<td>24</td>
<td>1</td>
<td>Starting location in VSTR of the bounds vector for the constrained system.</td>
</tr>
<tr>
<td>IMAX</td>
<td>25</td>
<td>1</td>
<td>The number of non-zero elements of $b_{ij}$ (linear constraints).</td>
</tr>
<tr>
<td>NIJ</td>
<td>26</td>
<td>1</td>
<td>Starting location in VSTR of the vector containing the $i \times 100 + j$ where $i$, $j$ refer to the elements of $b_{ij}$.</td>
</tr>
<tr>
<td>NST</td>
<td>27</td>
<td>1</td>
<td>Temporary storage used in VSTR for constraining the size of the system from $N$ to $M$.</td>
</tr>
<tr>
<td>NB</td>
<td>28</td>
<td>1</td>
<td>VSTR pointer for non-zero elements of the constraint matrix.</td>
</tr>
<tr>
<td>NC</td>
<td>29</td>
<td>1</td>
<td>Starting location in VSTR for additive constants used in linear constraints.</td>
</tr>
<tr>
<td>NMSTAT</td>
<td>30</td>
<td>1</td>
<td>Number of cells allotted per station in the master sensor table.</td>
</tr>
<tr>
<td>Name</td>
<td>BLK2( )</td>
<td>Dimension</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NLAM</td>
<td>31</td>
<td>1</td>
<td>Total number of entries in the altitude $C_{D}A/m$ table.</td>
</tr>
<tr>
<td>NLAMS</td>
<td>32</td>
<td>1</td>
<td>Number of drag parameters $C_{D}A/m$ in the solution vector.</td>
</tr>
<tr>
<td>NH</td>
<td>33</td>
<td></td>
<td>Starting location in VSTR of the altitude $C_{D}A/m$ table.</td>
</tr>
<tr>
<td>NLID</td>
<td>34</td>
<td></td>
<td>Starting location in VSTR of the identifiers for the drag parameters $C_{D}A/m$ appearing in the solution vector.</td>
</tr>
</tbody>
</table>
| NPALM   | 35      | 1         | Starting location in VSTR of the $b_{1}$ vectors where \[
\begin{align*}
    b_{1} &= \left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t}}{\partial(\alpha, \delta, \beta, A, R, v)_{t_{0}}} \right]^{-1} \\
    &\times \left[ \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})_{t}}{\partial \left( \frac{C_{D}A}{m} \right)_{i}} \right] \\
    &= \left[ \frac{\partial(\alpha, \delta, \beta, A, R, v)_{t_{0}}}{\partial \left( \frac{C_{D}A}{m} \right)_{i}} \right]
\end{align*}
\]
i=1, \ldots, NLAM

This array is NLAM x 6 cells long and is stored in the following manner:

\[
\begin{align*}
VSTR(NPALM to NPALM+5) &= \frac{\partial \alpha}{\partial \lambda^{'}_{1}}, \frac{\partial \delta}{\partial \lambda^{'}_{1}}, \frac{\partial \beta}{\partial \lambda^{'}_{1}}, \frac{\partial A}{\partial \lambda^{'}_{1}}, \frac{\partial R}{\partial \lambda^{'}_{1}}, \frac{\partial v}{\partial \lambda^{'}_{1}} \\
&\quad \vdots \frac{\partial \alpha}{\partial \lambda^{'}_{2}}, \frac{\partial \delta}{\partial \lambda^{'}_{2}}, \frac{\partial \beta}{\partial \lambda^{'}_{2}}, \frac{\partial A}{\partial \lambda^{'}_{2}}, \frac{\partial R}{\partial \lambda^{'}_{2}}, \frac{\partial v}{\partial \lambda^{'}_{2}} \\
&\quad \vdots \frac{\partial \alpha}{\partial \lambda^{'}_{NLAM}}, \frac{\partial \delta}{\partial \lambda^{'}_{NLAM}}, \frac{\partial \beta}{\partial \lambda^{'}_{NLAM}}, \frac{\partial A}{\partial \lambda^{'}_{NLAM}}, \frac{\partial R}{\partial \lambda^{'}_{NLAM}}, \frac{\partial v}{\partial \lambda^{'}_{NLAM}} \\
VSTR(NPALM+6*NLAM-1) &= \frac{\partial v}{\partial \lambda^{'}_{NLAM}}
\end{align*}
\]

6-7
Name    | BLK2( ) | Dimension | Description
---|---|---|---
NPXLM   | 36   | 1   | Starting location in VSTR of the $c_i$ vectors, where

$$
\lambda_i = \left( \frac{C_{DA}}{m} \right)_i
$$

This array is NLAM x 6 cells long and is stored in the following manner:

$$
\begin{align*}
VSTR(\text{NPXLM to NPXLM+5}) &= \frac{\partial x}{\partial \lambda_1}, \frac{\partial y}{\partial \lambda_1}, \frac{\partial z}{\partial \lambda_1}, \frac{\partial x}{\partial \lambda_2}, \frac{\partial y}{\partial \lambda_2}, \frac{\partial z}{\partial \lambda_2} \\
(NPXLM+6 \text{ to } NPXLM+11) &= \frac{\partial x}{\partial \lambda_2}, \ldots \quad \ldots \quad \ldots \quad \frac{\partial z}{\partial \lambda_2} \\
&\quad \ldots \quad \ldots \quad \ldots \\
VSTR(\text{NPXLM+6*NLAM-1}) &= \ldots \quad \ldots \quad \ldots \quad \frac{\partial z}{\partial \lambda_{NLAM}}
\end{align*}
$$

where

$$
\lambda_i = \left( \frac{C_{DA}}{m} \right)_i
$$

NPXDLM  | 37   | 1   | Starting location in VSTR of the $d_i$ vectors, where

$$
d_i = \left[ \frac{\partial (\tilde{x}, \tilde{y}, \tilde{z})}{\partial \left( \frac{C_{DA}}{m} \right)_i} \right]_t
$$

$$
i = 1, \ldots, \text{NLAM}
$$
This array is NLAM x 3 cells in length and is stored in the following manner:

\[ VSTR(NPXDLM \text{ to } NPXDLM+2) = \frac{\partial \hat{x}}{\partial \lambda_1}, \frac{\partial \hat{y}}{\partial \lambda_1}, \frac{\partial \hat{z}}{\partial \lambda_1} \]

\[ VSTR(NPXDLM+3 \text{ to } NPXDLM+5) = \frac{\partial \hat{x}}{\partial \lambda_2}, \frac{\partial \hat{y}}{\partial \lambda_2}, \frac{\partial \hat{z}}{\partial \lambda_2} \]

\[ \vdots \]

\[ VSTR(NPXDLM+3*NLAM-1) = \frac{\partial \hat{z}}{\partial \lambda_{NLAM}} \]

where

\[ \lambda_i = \left( \frac{CDA}{m} \right)_i \]

3) Array BLK3

<table>
<thead>
<tr>
<th>Name</th>
<th>BLK3(_)</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECRIT</td>
<td>90</td>
<td>1</td>
<td>Minimum elevation at which steering ephemeris is allowed. Nominally set at -5°. (See Page 3-5, Steering Ephemeris.)</td>
</tr>
<tr>
<td>RDFLG</td>
<td>91</td>
<td>1</td>
<td>Flag which controls the output units of range rate and range acceleration in the steering ephemeris. Nominally set = 0. (See Page 3-5, Steering Ephemeris for further description.)</td>
</tr>
</tbody>
</table>

4) Array BLK 4 has been increased from 400 to 450 cells. The following variables have been added:

<table>
<thead>
<tr>
<th>Name</th>
<th>BLK4(_)</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRMS</td>
<td>392</td>
<td>5</td>
<td>The RMS from the previous iteration of the range, azimuth, elevation, range rate, and range acceleration residuals.</td>
</tr>
<tr>
<td>Name</td>
<td>BLK4( )</td>
<td>Dimension</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>CFLAG</td>
<td>397</td>
<td>1</td>
<td>Linear constraints flag for additive constants.</td>
</tr>
<tr>
<td>IFIT</td>
<td>398</td>
<td>1</td>
<td>(See page 2-60 of Reference 1.)</td>
</tr>
<tr>
<td>CFLG</td>
<td>399</td>
<td>1</td>
<td>Linear constraints flag for additive constants.</td>
</tr>
<tr>
<td>IFVE</td>
<td>400</td>
<td>2</td>
<td>Two flags indicating whether the respective ( C_{DA}/m ) in the region of influence is in the solution vector or not.</td>
</tr>
<tr>
<td>ALT</td>
<td>402</td>
<td>2</td>
<td>Two altitude layers bounding the current region of influence. (e. r.)</td>
</tr>
<tr>
<td>INFG</td>
<td>404</td>
<td>1</td>
<td>Flag indicating to subroutine PLAMDA whether an altitude crossing has occurred and which region of drag influence has been entered.</td>
</tr>
<tr>
<td>NDPRT</td>
<td>405</td>
<td>1</td>
<td>Number of CAT1 variables plus number of ( C_{DA}/m ) drag parameters being integrated at any one instance (either 6 or 8).</td>
</tr>
<tr>
<td>NH1</td>
<td>406</td>
<td>1</td>
<td>Location in VSTR of the 1st altitude layer bounding the current region of influence.</td>
</tr>
<tr>
<td>NH2</td>
<td>407</td>
<td>1</td>
<td>Location in VSTR of the 2nd altitude layer bounding the current region of influence.</td>
</tr>
</tbody>
</table>
7. PREPCD

PREPOD is a preliminary orbit determination program designed to be used in conjunction with the NRTPOD program. A preliminary estimate of the position and velocity of a satellite is derived by fitting an orthogonal polynomial of degree $\leq 4$ in the least squares sense to the components of $n$ position fixes (topocentric range, elevation, azimuth) from an observing sensor. The velocity components are obtained by differentiating the position polynomials ($R, A, E$) with respect to time.

The observations, station biases, if any, and station coordinates are input on cards in the NRTPOD format. See Reference 1 for a description of these input cards. The degree of the polynomial fit is nominally set to four and is automatically adjusted downward if there are fewer than five observations. Also, the degree may be selected by the analyst, provided that the number of observations exceeds the degree by at least one.

The epoch may be selected at any time past (and including) the time of the first observation. If epoch is not specified or if epoch is specified prior to the time of the first observation, the time of the first observation is automatically selected as epoch.

The evaluated polynomials of the topocentric quantities ($R, A, E, \dot{R}, \dot{A}, \dot{E}$) are converted to Cartesian Earth-centered inertial coordinates. For output purposes, the geocentric Cartesian coordinates are transformed to polar spherical (ADBARV) coordinates, if desired.

The output consists of the geocentric inertial state vector and the associated epoch time; it is printed and punched on cards which are suitable for input to NRTPOD.

The flow diagram on the following page is a general computer program logic flow of PREPOD. The following sections describe the input/output, COMMON storage, and subroutines. A description of the mathematical techniques used in PREPOD to fit an orthogonal polynomial to $n$ observations in a least squares sense is given in Reference 9.
Figure 7-1. PREPOD Flow Diagram
The formulation of the least squares curve fit as programmed in PREPOD could be easily modified to make it a weighted least squares curve fit. In Figure 7-5 the provision for printing the standard deviation for each observation is shown, should this modification be implemented.

7.1 PROGRAM INPUT

7.1.1 Deck Set-Up

The input deck consists of NAMELIST cards (program options), sensor cards, and observation cards, in that order. Cases may be stacked although some care must be exercised when program constants are changed from their nominal values. Program constants which are changed by NAMELIST are not restored to the nominal value after each case; hence, nominal values must be restored by the user in the subsequent cases of a stacked run. The deck set-up is shown pictorially in Figure 7-2.

When the program is run without any options, the following nominal program conditions prevail:

1) The polynomial fit is of degree 4.
2) The preliminary estimate is output in polar spherical coordinates.
3) The epoch is selected at the time of the first observation.
4) The observations must be presorted.

7.1.2 Input Cards

There are three basic types of input cards for the PREPOD program:

NAMELIST inputs
Sensor cards
Observation cards

The NAMELIST inputs constitute the preliminary data cards that specify the program options and the constants. The various types of cards are explained below. Some remarks concerning the use of the NAMELIST format are given in Reference 1, Section 1.2.4.
Figure 7-2. PREPOD Input Deck
The KDEG card specifies the degree of the polynomial fit. If this card is omitted from the input deck, the degree is set to 4. And if there are fewer than five observation cards, KDEG is automatically adjusted to be equal to one less than the number of observations. The maximum permissible value for KDEG is 4. This input is a single entry array.

Example

KDEG = 3,

This input permits the analyst to select the epoch to which the derived preliminary estimate is to be referenced. If this card is omitted from the input deck, the epoch is selected as the time of the earliest observation. If the selected time precedes the first observation, the program automatically resets the epoch to the time of the first observation. The TIME card is a six-entry array as follows:

Example

TIME = Year-1900, Month number, day number, hours, minutes, seconds,

This card specifies the coordinate system of the preliminary estimate of the orbit. If TYPE = 1, the output state vector is in polar spherical coordinates (ADBARV). And if TYPE = 2, the output is in Cartesian components of position and velocity. In both cases, the reference direction is the vernal equinox and the reference plane is the equator; and both coordinate systems are earth-centered inertial (ECI).

This input is a flag which indicates if the observations are presorted. The flag is nominally zero, which indicates that the observation cards are in time sort. If PRFLG5 = 1, the observations processor will sort the cards. (Maximum = 300 cards).

The CONS input is used to change the constants of the program. The program constants are printed in a block at the beginning of each run. The identification of each constant with respect to program location is given in Section 7.3. The constants which are changed with a CONS card are not restored to the nominal value after each case; therefore, nominal values must be restored by the analyst if stacked cases are submitted on one computer run.

7-5
Example

\[ \text{CONS}(3) = 0.437528 \times 10^{-2}, \]

Sensor Cards  
PREPOD uses the standard NRTPOD sensor cards as described in Reference 1, Section 1.2.5. However, the program only accepts sensor cards Type 1 and 2, station coordinates and observable biases respectively. Since the PREPOD curve fit is not weighted, standard deviations (type = 3) are not required. If a sensor bias card is input, the biases are removed from the observations before normal observation processing begins. It should be noted that there are two sensor cards at most in any one determination since PREPOD derives a preliminary estimate from the observations of one sensor only.

Observation Cards  
PREPOD uses the standard NRTPOD observation cards as described in Reference 1, Section 1.2.6. The program accepts range, azimuth, and elevation observables only; however, if range-rate or range-acceleration observables are input also, the program ignores them. The maximum number of observations the program will handle is 300. The observations do not have to be time sorted; however, an additional NAMELIST card (PRFLG5) must be input if the cards are presorted.

Figure 7-3 is a sample input sheet for two cases. The first case consists of nominal inputs only; in the second case, some of the program options are called. The parenthetical information of Figure 7-3 does not constitute part of the input; it is explanatory only.

7.2 PROGRAM OUTPUT

PREPOD produces printed output and punched output on cards suitable for input to NRTPOD.

7.2.1 Printed Output

The printed output is sectioned into one or more pages of particular information. The sections are outlined below in the output sequence.

<table>
<thead>
<tr>
<th>Data</th>
<th>No. of Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAMELIST input</td>
<td>1</td>
</tr>
<tr>
<td>Constants, sensors, and</td>
<td>1 or more</td>
</tr>
<tr>
<td>observations</td>
<td></td>
</tr>
<tr>
<td>Initial Conditions</td>
<td>1</td>
</tr>
</tbody>
</table>

7-6
The NAMELIST input print is essentially a card image listing of the input NAMELIST cards. Figure 7-4 is a sample input listing of PREPOD.

Following the NAMELIST input listing is a listing of the remaining input cards and the program constants. Figure 7-5 is a sample of this output. The program constants /BLK1/CONS(30) are printed at the top of the page. The definition of these constants is given in Section 7.3. Following the program constants is a listing of the sensor ID, location, and biases, if any. The observations are listed following the sensor information. Note that provision has been made for the standard deviation of each observable. Following the observations print is a message indicating the degree of the polynomial fit and the value of \( \alpha_0 \), the right ascension of Greenwich at 0.0 hrs. on the day of epoch.

The last page of output consists of the derived preliminary estimate of the orbit and its associated epoch. The type of coordinate system in which the estimate is output is also indicated. Figure 7-6 is a sample printout of the Initial Conditions page. The output units are degrees, kilometers, and kilometers/second.

### 7.2.2 Punched Output

The punched output of PREPOD is in the proper format for input to NRTPOD. The three output variables are listed below:

<table>
<thead>
<tr>
<th>Output Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>STVEC</td>
<td>The derived initial conditions in either polar spherical or Cartesian coordinates</td>
</tr>
<tr>
<td>TIME</td>
<td>The epoch date and time associated with the initial conditions</td>
</tr>
<tr>
<td>TYPE</td>
<td>The type of coordinate system; TYPE = 1, Polar Spherical Coordinates TYPE = 2, Cartesian Coordinates</td>
</tr>
</tbody>
</table>
Figure 7-3. PREPOD Sample Input Sheet for Two Stacked Cases
$INPUT

FCC TEST CASE

TYPE=2
KCEG=2
TIME=65,10,1,1,34,0.999985

Figure 7-4. Sample NAMELIST Input Listing
### PRELIMINARY HeliT Determination Program

#### Program Constants

<table>
<thead>
<tr>
<th>ID</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Alt</th>
<th>R Bias</th>
<th>A Bias</th>
<th>E Bias</th>
<th>Time Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.63781660E04</td>
<td>2 0.57255779E02</td>
<td>3 0.43726560E02</td>
<td>4 0.34566194E02</td>
<td>5 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.30599599E02</td>
<td>7 0.27999999E02</td>
<td>8 0.30599599E02</td>
<td>9 0.30000000E02</td>
<td>10 0.30999599E02</td>
<td>15 0.30999599E02</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.30000000E02</td>
<td>12 0.30599599E02</td>
<td>13 0.30599599E02</td>
<td>14 0.30000000E02</td>
<td>15 0.30999599E02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.30000000E02</td>
<td>17 0.30595959E02</td>
<td>18 0.62831853E01</td>
<td>19 0.3141925E01</td>
<td>20 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.</td>
<td>22 0.</td>
<td>23 0.</td>
<td>24 0.</td>
<td>25 0.30000000E01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.12000000E01</td>
<td>27 0.20000000E01</td>
<td>28 0.20000000E01</td>
<td>29 0.20000000E01</td>
<td>30 0.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Sensor Location

<table>
<thead>
<tr>
<th>ID</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Alt</th>
<th>R Bias</th>
<th>A Bias</th>
<th>E Bias</th>
<th>Time Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH</td>
<td>156.</td>
<td>7.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Observation Type

<table>
<thead>
<tr>
<th>ID</th>
<th>T-TO</th>
<th>YR</th>
<th>MN</th>
<th>DY</th>
<th>Fr Min</th>
<th>Secs</th>
<th>Range</th>
<th>Az</th>
<th>El</th>
<th>R DOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH</td>
<td>-1.000</td>
<td>65</td>
<td>10</td>
<td>1</td>
<td>133</td>
<td>1.000</td>
<td>9997.</td>
<td>265.4197</td>
<td>10.617</td>
<td>0.</td>
</tr>
<tr>
<td>MH</td>
<td>0.</td>
<td>65</td>
<td>10</td>
<td>1</td>
<td>34</td>
<td>1.000</td>
<td>9518.</td>
<td>267.1456</td>
<td>11.5001</td>
<td>0.</td>
</tr>
<tr>
<td>MH</td>
<td>1.000</td>
<td>65</td>
<td>10</td>
<td>1</td>
<td>25</td>
<td>1.000</td>
<td>9843.</td>
<td>268.9048</td>
<td>12.3007</td>
<td>0.</td>
</tr>
<tr>
<td>MH</td>
<td>2.000</td>
<td>65</td>
<td>10</td>
<td>1</td>
<td>36</td>
<td>1.000</td>
<td>5770.</td>
<td>270.6976</td>
<td>13.0822</td>
<td>0.</td>
</tr>
</tbody>
</table>

### Polynomial Fitting OBS is OF Degree 2

**Alpha G Zero = 5.5148 Degrees**

---

*Figure 7-5. Sample Output of Program Constants, Sensor Location and Biases, and Observations*
<table>
<thead>
<tr>
<th>YEAR</th>
<th>MONTH</th>
<th>DAY</th>
<th>HOUR</th>
<th>MINUTE</th>
<th>SECCAD</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>XDOT</th>
<th>YDOT</th>
<th>ZDOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1965</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>34</td>
<td>1000</td>
<td>-5.4569442E-02</td>
<td>-1.1634653E-04</td>
<td>5.2769009E-03</td>
<td>3.7164931E-01</td>
<td>2.2651133E-00</td>
<td>5.0287881E-00</td>
</tr>
</tbody>
</table>

**Type = 2**

Figure 7-6. Sample Initial Conditions Page
STVEC=-0.94969442E 03,-0.11634953E 05, 0.52785009E 04
  0.37164931E-00, 0.22511337E 01, 0.50287880E 01
TIME=65,10, 1, 1,34, 1.000
TYPE=2

Figure 7-7. Listing of Punched Card Output
Figure 7-7 is a card listing of the punched card output of PREPOD. These punched cards correspond to the sample printout case described in the previous section.

7.3 PREPOD STORAGE MAP

/BLK/CONS(30)          Program Constants

CONS(1)  CKMER  Kilometers/earth radii
CONS(2)  CDEG  Degrees/radian
CONS(3)  CWE  $\omega_e$, rotation rate
CONS(4)  CELLIP  $f$, ellipticity of earth
*CONS(5)  CAE  Internal units/earth radii
CONS(6)  CDAYMN(12)  Day of the month, non-leap year
                 Jan = CDAYMN(1), etc.
CONS(18)  C2PI  $2\pi$
CONS(19)  CPI  $\pi$
CONS(25)  KOUT  System output tape
CONS(26)  IOUT  System punch tape
CONS(27)  KIN  System input tape
CONS(28)  ITYPE  Type of output
                 1 = ADBARV
                 2 = Cartesian
CONS(29)  KDEG  Degree of polynomial
                 4 = Nominal

COMMON/BLK2/WSTR(100)

WSTR(1)  TIME(6)  Epoch time, year, month, day, hour, minutes, seconds
WSTR(7)  TEPOCH  Minutes from midnight, day of epoch
WSTR(8)  STVEC(6)  State vector
              ITYPE = 1 $\alpha, \beta, \delta, A, R, V$
              ITYPE = 2 $X, Y, Z, \hat{X}, \hat{Y}, \hat{Z}$

*Not used when internal units are earth radii

7-13
7.4 PREPOD SUBROUTINE DESCRIPTIONS

This section contains a glossary and description of the PREPOD program subroutines.
7.4.1 Subroutine Glossary

This glossary is an alphabetical list of the PREPOD subroutines. The subroutines having an asterisk denote the following documentation:

* NRTPOD documentation, Reference 1
** ESPOD documentation, Reference 2

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDOBS</td>
<td>Reads one observation card.</td>
</tr>
<tr>
<td>CLTIME*</td>
<td>Computes Gregorian time.</td>
</tr>
<tr>
<td>CTOP**</td>
<td>Converts a Cartesian state vector to polar spherical coordinates (ADBARV).</td>
</tr>
<tr>
<td>DOT*</td>
<td>Computes scalar product.</td>
</tr>
<tr>
<td>INPTC</td>
<td>Reads NAMELIST input and NRTPOD sensor cards.</td>
</tr>
<tr>
<td>LODOBS</td>
<td>Main control for observation card processor.</td>
</tr>
<tr>
<td>LSQFIT</td>
<td>Computes the coefficients of an orthogonal polynomial for fitting observations in the least square sense.</td>
</tr>
<tr>
<td>MAGN**</td>
<td>Computes the magnitude and magnitude squared of a 3-D vector.</td>
</tr>
<tr>
<td>OBSIN</td>
<td>Scales observations to internal units and applies biases, if any.</td>
</tr>
<tr>
<td>OBSSRT</td>
<td>Sorts observations by time.</td>
</tr>
<tr>
<td>PIMOD*</td>
<td>Modulates an angle between 0 and 2π.</td>
</tr>
<tr>
<td>PNCHVC</td>
<td>Punches the PREPOD output in NAMELIST format.</td>
</tr>
<tr>
<td>PRNTC</td>
<td>Prints the block of program constants.</td>
</tr>
<tr>
<td>RADSQ*</td>
<td>Computes the magnitude and magnitude squared of a 3-D vector.</td>
</tr>
<tr>
<td>SCLOUT</td>
<td>Scales the ECI state vector from internal units to external units.</td>
</tr>
<tr>
<td>SETCON</td>
<td>Sets constants for program.</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Functional Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>TIME**</td>
<td>Converts Y, M, D, H, M, S to Julian date: days + fraction.</td>
</tr>
<tr>
<td>TINIT**</td>
<td>Sets up initial time, computes $a_g$.</td>
</tr>
<tr>
<td>TPTOIN</td>
<td>Converts topocentric state vector to Cartesian (ECI) coordinates.</td>
</tr>
</tbody>
</table>
7.4.2 Subroutine Descriptions

SUBROUTINE IDENTIFICATION

A. Title
BCDOBS

B. Segment
PREPOD

C. Called by subroutine
LODOBS

FUNCTION

To read one BCD observation card and store for processing by LODOBS. It also calls TINIT to compute $\alpha_{g_0}$ and epoch time from the first observation card if no epoch time is requested or if epoch time precedes the first observation.

USAGE

A. Calling sequence
Call BCDOBS (A, SEOF)

B. Input

1. COMMON
   /BLK1/CONS(30)
   KOUT Output tape number
   KIN Input tape number
   /BLK2/WSTR(100)
   ITFLG Flag to indicate if epoch time and $\alpha_{g_0}$ desired
   IFRST Flag to test if first observation card

2. Calling sequence

C. Output

1. COMMON
   /BLK2/WSTR(100) (If no epoch time requested)
Calling Sequence

A

SEOF

D. Error/action message

"PROGRAM IGNORES TYPES 1 AND 2 OBSERVATION CARDS."

If there is a punch in column 20, card is ignored.

SUBROUTINES USED

A. Library
   Input/output

B. Program
   TINIT
   TIME
SUBROUTINE IDENTIFICATION

A. Title
   INPTC

B. Segment
   PREPOD

C. Called by subroutine
   PREPOD

FUNCTION

To read the NAMELIST input and the NRTPOD sensor card. Also the working storage (WSTR) is initialized for stacked cases.

USAGE

A. Calling sequence
   Call INPTC

B. Input
   1. COMMON
      /BLK1/CONS(30)
      KOUT System output tape number
      KIN System input tape number
   2. Calling sequence

C. Output
   1. COMMON
      /BLK1/CONS(30)
      KDEG Degree of polynomial
      ITYPE Type of output 1 = ADBARV
              2 = XYZXYŻ
/BLK2/WSTR(100)

TIME(6)  Calendar date of epoch if entered
ITFLG    Flag to indicate if an epoch time was entered
STID     Station identification
STLAT    Station latitude
STLONG   Station longitude
STALT    Station altitude
ABIAS     Station azimuth bias
EBIAS     Station elevation bias
RBIAS     Station range bias
TBIAS     Station time bias
PRFLG5   Flag to indicate that observations are sorted
IFRST    Flag to indicate that the epoch time must be checked against the first observation time

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library
   Input/output

B. Program
   -
SUBROUTINE IDENTIFICATION

A. Title
   LODOBS

B. Segment
   PREPOD

C. Called by subroutine
   PREPOD

FUNCTION

To read and sort the observation cards. This subroutine is essentially unchanged from the NRTPOD version as described in Section 5.3 of Reference 1. The following characteristics distinguish the PREPOD version:

1. U(3500), the temporary storage used as a buffer for observations, is smaller and is used as working storage by the LSQFIT subroutine.

2. 300 is the maximum number of observations which may be processed.

3. No tape is written.
SUBROUTINE IDENTIFICATION

A. Title
LSQFIT

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To calculate orthogonal polynomial coefficients for fitting a polynomial to a number of observations in the least-square sense.

USAGE

A. Calling sequence
Call LSQFIT (MAX, KDEG, PDJ, TDJ, XDJ, BUFS, BMD, A)

B. Input

1. COMMON
   -

2. Calling sequence
   MAX          Number of observations
   KDEG         Degree of polynomial
   PDJ          (KDEG+1)*MAX locations of temporary storage
   TDJ          Independent variable (normalized) MAX number
   XDJ          Dependent variable MAX number
   BUFS         (KDEG+1)*4 locations of temporary storage
   BMD          ((KDEG+2)*(KDEG+3))/2 locations of temporary storage

C. Output

1. COMMON
   -
2. Calling sequence

A. \(A(1) = a_0\)  
   \(A(2) = a_1\)  
   \(\ldots\)  
   \(A(d+1) = a_d\)

where

\[ X = a_0 + \left(\frac{a_1}{(t_n - t_0)}\right)t + \left(\frac{a_2}{(t_n - t_0)^2}\right)t^2 + \ldots \]

\[ + \left(\frac{a_d}{(t_n - t_0)^d}\right)t^d \]

D. Error/action messages

SUBROUTINES USED

A. Library
   -

B. Program
   -

EQUATIONS

Given \(M\) observations \(t_1, x_1; t_2, x_2; \ldots; t_M, x_M\) a polynomial of degree \(d\) is to be fitted to these observations. First, a \((d+1) \times M\) matrix of orthogonal polynomials is formed recursively as follows:
The times are normalized

\[ t_{j'} = \frac{t_j - t_1}{t_M - t_1} \]

\[ P_1(t_{j'}) = 1 \quad (j = 1, M) \text{ for all } P_j \text{'s} \]

\[ P_2(t_{j'}) = (t_{j'} - u_1) P_1(t_{j'}) \]

\[ P_3(t_{j'}) = (t_{j'} - u_2) P_2(t_{j'}) - V_2 P_1(t_{j'}) \]

\[ \vdots \]

\[ P_{d+1}(t_{j'}) = (t_{j'} - u_d) P_d(t_{j'}) - V_d P_{d-1}(t_{j'}) \]

where

\[ u_d = \sum_{j=1}^{M} t_{j'} \left[ P_d(t_{j'}) \right]^2 / \sum_{j=1}^{M} \left[ P_d(t_{j'}) \right]^2 \]

\[ V_d = \sum_{j=1}^{M} \left[ P_d(t_{j'}) \right]^2 / \sum_{j=1}^{M} \left[ P_{d-1}(t_{j'}) \right]^2 \quad \text{except } V_1 = 0. \]
the desired polynomial is

\[ y(t'_{j}) = \sum_{i=1}^{d+1} S_i P_i(t'_{i}) \]

the \( S_i \) are calculated as follows:

\[
S_i = \sum_{j=1}^{M} X_j P_i(t'_{j}) / \sum_{j=1}^{M} [P_i(t'_{j})]^2 \quad i = 1, 2, \ldots, d+1
\]

or in matrix notation:

\[
[S'_1, S'_2, \ldots, S'_{d+1}] = [X_1, X_2, \ldots, X_m] \times \begin{bmatrix}
1.0 & 1.0 & 1.0 \\
1.0 & 2.0 & 1.0 \\
\vdots & \vdots & \vdots \\
1.0 & P_{M,1} & P_{M,M}
\end{bmatrix}
\]

and

\[
S_i = S'_i / \sum_{j=1}^{M} [P_{ij}]^2
\]

To obtain the coefficients in the usual form:

\[
P_i(t'_{j}) = \sum_{i=1}^{d+1} b_{i,d+1} (t'_{j})^i
\]
\[ b_{i,k} = b_{i-1,k-1} - c_k b_{i-1,k-1} - V_k b_{i,k-2} \]

where

\[ b_{i,k} = \begin{cases} 1.0 & i = k \\ 0.0 & i < 0 \text{ or } i > d+1 \end{cases} \]

Yielding a lower triangular array with 1's down the diagonal:

\[
\begin{bmatrix}
1.0 & b_{1,2} & b_{1,3} & \cdots & b_{1,d+1} \\
0.0 & 1.0 & b_{2,3} & \cdots & b_{2,d+1} \\
0.0 & 0.0 & 1.0 & \cdots & b_{3,d+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0.0 & 0.0 & 0.0 & \cdots & 1.0 \\
\end{bmatrix}
\]

\[ [S_1 \ldots S_{d+1}] \left[ \begin{array}{c} b_{1,1} \\ b_{1,2} \\ b_{1,3} \\ \vdots \\ b_{1,d+1} \end{array} \right] = a_0, a_1, a_2, \ldots, a_d \]
SUBROUTINE IDENTIFICATION

A. Title
   OBSIN

B. Segment
   PREPOD

C. Called by subroutine
   LODOBS

FUNCTION

To subtract station biases, convert to internal units, and compute the
time of the observations in minutes from midnight, day of epoch.

USAGE

The usage is essentially as described in the NRTPOD document, Reference 1, Section 5.3, the only difference being that the logic to handle range-rate observations has been eliminated.

SUBROUTINES USED

A. Library
   -

B. Program

   TIME  Computes Julian date and minutes
          from midnight of epoch day
OBSSRT

SUBROUTINE IDENTIFICATION

A. Title
   OBSSRT

B. Segment
   PREPOD

C. Called by subroutine
   LODOBS

FUNCTION

To sort the observations by time with respect to the earliest time.

USAGE

The usage of this version of OBSSRT is essentially as described in
the NRTPOD version, Reference 1, Section 5.3, with the exception that
the logic which sorts the observations with respect to epoch has been
eliminated.

SUBROUTINES USED

A. Library
   -

B. Program
   -
PNCHVC

SUBROUTINE IDENTIFICATION

A. Title
PNCHVC

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To punch and print the state vector, STVEC, (ADBARV or X, Y, Z, X, Y, Z) on cards, the epoch TIME card and the TYPE card for use as input to NRTPOD in the NAMELIST format.

USAGE

A. Calling sequence
Call PNCHVC

B. Input

1. COMMON
   /BLK1/CONS(30)
   KOUT System output tape number
   IOUT System punch tape number
   ITYPE Type of output 1 = ADBARV
            2 = X, Y, Z, X, Y, Z
   /BLK2/WSTR(100)
   TIME(5) Year, Month, Day, Hour, Minutes of epoch
   SECS Seconds of epoch
   STVEC(6) Polar or cartesian coordinates

2. Calling sequence
   ---

C. Output

1. COMMON
   ---
2. Calling sequence

3. Printed output
   Print and punch STVEC, TIME and TYPE cards

D. Error/action messages
   -

SUBROUTINES USED

A. Library
   Input/Output

B. Program
   -
PRNTC

SUBROUTINE IDENTIFICATION

A. Title
   PRNTC

B. Segment
   PREPOD

C. Called by subroutine
   PREPOD

FUNCTION

To print the block of program constants, station coordinates, and biases and to convert the station coordinates and biases to internal units.

USAGE

A. Calling sequence
   Call PRNTC

B. Input

1. COMMON
   /BLK1/CONS(30)
   KOUT System output tape number
   CDEG Degrees to radians
   CKMER Kilometers to earth radii
   /BLK2/WSTR(100)
   STID
   STLAT External units degrees
   STLONG External units degrees
   STALT External units meters
   RBIAS External units kilometers
   ABIAS External units degrees
   EBIAS External units degrees
   TBIAS External units minutes

2. Calling sequence
   ---

7-35
C. Output

1. COMMON
   /BLK2/WSTR(100)
   STLAT Internal units radians
   STLONG Internal units radians
   STALT Internal units earth radii
   RBIAS Internal units earth radii
   ABIAS Internal units radians
   EBIAS Internal units radians
   TBIAS Internal units minutes

2. Calling sequence

3. Printed output
   Program Constants/BLK1/CONS
   Sensor coordinates and biases

D. Error/action messages
   ---

SUBROUTINES USED

A. Library
   Input/output

B. Program
   -
SCLOUT

SUBROUTINE IDENTIFICATION

A. Title
SCLOUT

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION

To scale the state vector \((X, Y, Z, \dot{X}, \dot{Y}, \dot{Z} \text{ or ADBARV})\) from internal units \((\text{E.R., E.R./min, and radians})\) to external units \((\text{Km, Km/sec, and degrees})\).

USAGE

A. Calling sequence
Call SCLOUT

B. Input

1. COMMON
   /BLK1/CONS(30)
   CKMER
   CDEG
   ITYPE
   Kilometers/earth radii
   Degrees/radian
   Type of output desired
   1 = ADBARV
   2 = Cartesian

   /BLK2/WSTR(100)
   STVEC(6)
   State vector in internal units

2. Calling sequence

C. Output

1. COMMON
   /BLK2/WSTR(100)
   STVEC(6)
   State vector in external units
SCLOUT

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
SETCON

B. Segment
PREPOD

C. Called by subroutine
PREPOD

FUNCTION
To assign nominal values of program constants and clear memory. Nominal outputs are ADBARV state vector, epoch time, and Type = 1. Program is only entered once and is not entered for stacked cases.

USAGE

A. Calling sequence
Call SETCON

B. Input
1. COMMON
   -

2. Calling sequence
   -

C. Output
1. COMMON
   /BLKI/CONS(30)
   CKMER = 6378.165 Kilometers/earth radii
   CDEG = 57.29577951 Degrees/radian
   CWE = 4.37526906E - 3 Rotation rate
   CELLIP = 1.0/298.3 Earth flattening
   CDAYMN
   (1) = 31. January
   (2) = 28. February
   (3) = 31. March
   (4) = 30. April
(5) = 31. May
(6) = 30. June
(7) = 31. July
(8) = 31. August
(9) = 30. September
(10) = 31. October
(11) = 30. November
(12) = 31. December
CPI = 3.1415926536 \pi
C2PI = 6.2831853072 2\pi
KOUT = 3 System output tape
IOUT = 12 System punch tape
KIN = 2 System input tape
ITYPE = 1 ADBARV output
KDEG = 4 4th degree polynomial

2. Calling sequence

D. Error/action messages

SUBROUTINES USED

A. Library

B. Program
SUBROUTINE IDENTIFICATION

A. Title
   TPTOIN

B. Segment
   PREPOD

C. Called by subroutine
   PREPOD

FUNCTION

To convert the state vector from topocentric rectangular coordinates to geocentric inertial coordinates.

USAGE

A. Calling sequence
   Call TPTOIN(STLAT, STLONG, STALT, COOR, STVEC)

B. Input

1. COMMON

2. Calling sequence
   STLAT Geodetic latitude of the observer
   STLONG Right Ascension of the observer
   STALT Altitude of the observer
   COOR(6) (R, A, E, R, A, E) of the object from the observer

C. Output

1. COMMON

2. Calling sequence
   STVEC(6) Geocentric Inertial coordinates of the object (X, Y, Z, X, Y, Z)

D. Error/action messages
   -
SUBROUTINES USED

A. Library
   SIN
   COS
   SQRT

B. Program

EQUATIONS

Given \( R, A, E, \dot{R}, \dot{A}, \dot{E} \) of an object observed from a station with coordinates:

\[
\begin{align*}
\varphi &= \text{geodetic latitude} \\
\lambda &= \text{right ascension} \\
h &= \text{altitude} \\
X &= -R \cos E \cos A \\
Y &= R \cos E \sin A \\
Z &= R \sin E \\
\dot{X} &= -\dot{R} \cos E \cos A + R \dot{E} \sin E \cos A + R \dot{A} \cos E \sin A \\
\dot{Y} &= \dot{R} \cos E \sin A - R \dot{E} \sin E \sin A + R \dot{A} \cos E \cos A \\
\dot{Z} &= \dot{R} \sin E + R \dot{E} \cos E
\end{align*}
\]

\[
\begin{align*}
g_1 &= \frac{a_e}{\sqrt{1 - (2f - f^2) \sin^2 \varphi}} + h \\
g_2 &= \frac{a_e (1-f)^2}{\sqrt{1 - (2f - f^2) \sin^2 \varphi}} + h
\end{align*}
\]

\[
\begin{align*}
X_1 &= -g_1 \cos \varphi \cos \lambda \\
Y_1 &= -g_1 \cos \varphi \sin \lambda \\
Z_1 &= -g_2 \sin \varphi \\
\dot{X}_1 &= -\omega_e Y_1
\end{align*}
\]
\[
\begin{align*}
\dot{Y}_1 &= + \omega_e X_1 \\
\dot{Z}_1 &= 0.0 \\
\rho_X &= X \sin \varphi \cos \lambda - Y \sin \lambda + Z \cos \varphi \cos \lambda \\
\rho_Y &= X \sin \varphi \sin \lambda + Y \cos \lambda + Z \sin \lambda \cos \varphi \\
\rho_Z &= -X \cos \varphi + Z \sin \varphi \\
\rho_X' &= \dot{X} \sin \varphi \cos \lambda - \dot{Y} \sin \lambda + \dot{Z} \cos \varphi \cos \lambda \\
\rho_Y' &= \dot{X} \sin \varphi \sin \lambda + \dot{Y} \cos \lambda + \dot{Z} \sin \lambda \cos \varphi \\
\rho_Z' &= -\dot{X} \cos \varphi + \dot{Z} \sin \varphi \\
\dot{\rho}_X &= \rho_X' - \omega_e Y \\
\dot{\rho}_Y &= \rho_Y' + \omega_e X \\
\dot{\rho}_Z &= \rho_Z \\
X_I &= \rho_X - X_1 \\
Y_I &= \rho_Y - Y_1 \\
Z_I &= \rho_Z - Z_1 \\
\dot{X}_I &= \dot{\rho}_X - \dot{X}_1 \\
\dot{Y}_I &= \dot{\rho}_Y - \dot{Y}_1 \\
\dot{Z}_I &= \dot{\rho}_Z - \dot{Z}_1 
\end{align*}
\]
REFERENCES


This document describes the six modifications which have been added to the NRTPOD program in the form of extended capabilities. The report is intended as an analyst's guide to the NRTPOD modifications as well as an operational handbook with input-output instructions. In addition, a separate stand-alone program is described which derives a preliminary estimate of an orbit and is designed to be used in conjunction with NRTPOD.