NASA CONTRACTOR REPORT

NASA CR-599

NASA CR-599

GPO PRICE $ ______

CFSTI PRICE(S) $ 2.50

Hard copy (HC) ______

Microfiche (MF) .75

ATTITUDE DETERMINATION USING A KALMAN FILTER

VOLUME II

N66 36301
(ACCESSION NUMBER)
77
(PAGES)
CR-599
(NASA CR OR TMX OR AD NUMBER)
(THRU)
1
(CODE)
14
(CATEGORY)
FACILITY FORM

by James L. Farrell

Prepared by
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.
for Goddard Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1966

NASA CR-599

ATTITUDE DETERMINATION USING A KALMAN FILTER

VOLUME II

By James L. Farrell

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS 5-9195 by
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.

for Goddard Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $2.50

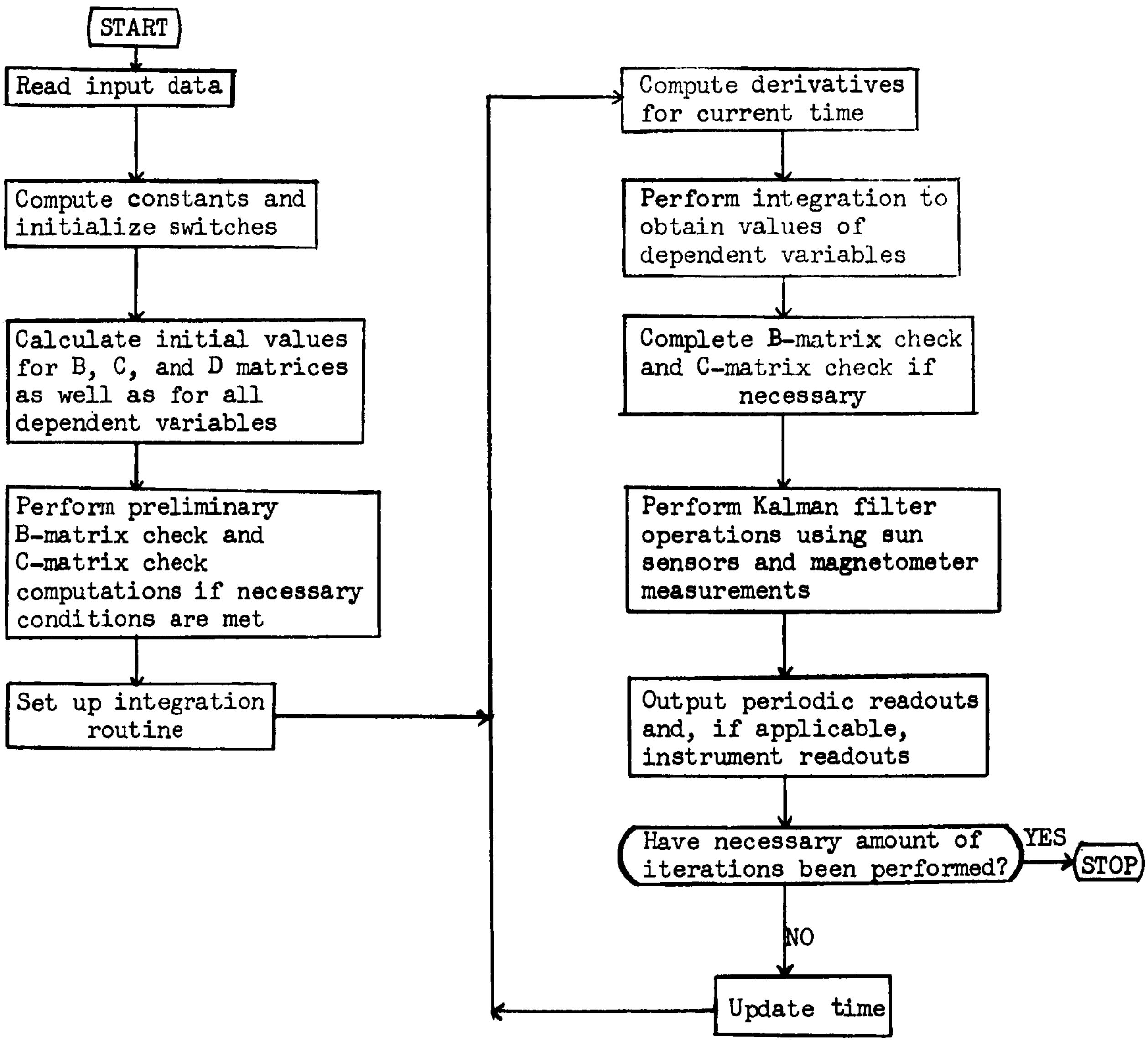
Section I. Introduction

This volume contains documentation of the computer program written under contract NAS5-9195. The program determines the attitude of an orbiting vehicle using a Kalman filter according to the analysis performed in Volume I of this report.

The program is written in FORTRAN IV with subroutines in FORTRAN and MAP. It is written for an IBM 7094 and has been successfully run on the Moonlight System at NASA. The naming of variables follows FORTRAN II conventions as to fixed and floating point names with most floating point variables being typed double precision. One large main program with a few small subroutines was originally intended but, due to compiler limitations, the creation of several other subroutines was necessary.

In addition to a description of the program and its usage, Volume II also contains a sample set of input data and a portion of the results generated from this data.

BLOCK DIAGRAM



Section II. Description of Routines

A. MAIN PROGRAM: JF

The main program performs the majority of the work required by the analysis. After the data is read-in and all necessary initialization is done, the initial values of the dependent variables are calculated.

The integration loop is then entered. The derivatives are calculated in BOX A (see part B of this section); the integration is performed by subroutine DICE and the results of the integration are used in the Kalman filter logic in BOX C, followed by the periodic and instrument readouts.

When the required number of iterations have been performed, the program begins to process the next case.

B. SUBROUTINE DICE (P,T,TP,E1,E2,N,Y,DY,F,L,INDEX,I)

PURPOSE: DICE is a double precision routine designed to be used in conjunction with an integration subroutine (DINTEG) to provide a numerical solution of an Nth order system of linear and/or non-linear differential equations expressed as a system of N first order equations:

$\dot{Y}_n = f_n\ (t,\ Y_1,\ Y_2,\ \ldots\ldots Y_n)\quad Y_n\ (t_o) = Y_{n,o};\ n = 1,\ 2,\ 3,\ \ldots\ldots N$

where Y_n are the N dependent variables and t is the independent variable. The local error generated by the numerical process is controlled by adjusting the integration step size based on the relative error as estimated by extrapolation to zero step size.

The basic method of the DICE routine can be summarized as follows:

1. Use initial conditions for the N dependent variables as solutions at $t = t_o$ and adopt the maximum step size permissible.
2. Perform the computation (integration subroutine) to find the single step solution for the new value of the independent variable, t.
3. Halve the integration step size.
4. Solve the problem again (integration subroutine) from the last convergent point using two applications of the reduced step size.
5. Compute an extrapolated solution making use of both solutions.
6. Compare the two-step solution with the extrapolated solution.
7. If the two solutions are sufficiently close, the procedure is continued using the extrapolated solution as initial values at the updated time t.

8. If the two solutions are not sufficiently close, the reduced step size is again halved and the procedure restarted from step 4 using the conditions at the last convergent point.

9. If the two solutions agree too closely then the basic step is doubled and the procedure restarted at step 1 using the extrapolated solution at the new value of the independent variable.

The above briefly describes the method; however, additional details are given below.

The superscripts (I) and (II) refer to the one-step and two-step approximations respectively. The subscript $i + 1$ refers to the solution at the new value of the independent variable, i referring to the old value. $h = t_{i+1} - t_i$ is the step size. The one-step approximation is found by the method in the integration subroutine. The two-step approximation is found by halving the step size and computing a one-step approximation at time $t_i + h/2$. This approximation is then used as the starting point for another one-step approximation again using the halved value of step size. This two-step approximation then can be thought of as two one-step approximations.

The procedure used to automatically control the error is as follows:

1. An error estimate for each of the N dependent variables using extrapolation to zero step size is expressed as:

$$E_{n,i+1} = \frac{Y^{(II)}_{n,i+1} - Y^{(I)}_{n,i+1}}{2^A - 1}$$

where A is the order of integration.

2. These values of $E_{n,i+1}$ are added to the two-step results to obtain the extrapolated solutions:

$$\overset{(II)}{Y_{n,i+1}} = Y_{n,i+1} + E_{n,i+1}$$

3. A relative error term is next defined and computed for each variable:

$$R_{n,i+1} = \frac{E_{n,i+1}}{Y_{n,i+1}}$$

4. The maximum relative error $(R_{n,i+1})_{max}$ is designated E and is tested against two bounds E1 and E2. E1 is to keep the integration step size from remaining too small and E2 is used to keep the interval from becoming too large. The program then halves the interval and repeats the step, continues at the same interval, or continues at twice the interval.

5. The subscript n of the maximum relative error is recorded in INDEX, an argument of the calling sequence, and if printed out, can be helpful in locating the trouble point in cases of non-convergence.

The arguments in the calling sequence are defined as:

P - maximum allowable step size.

T - independent variable.

TP - value of independent variable at which DICE will provide entry to BOXC.

E1 - lower error bound for controlling step size.

E2 - upper error bound for controlling step size.

N - number of dependent variables expressed as a fixed point variable or constant.

Y - first dependent variable followed by the remaining N-1 dependent variables (must be dimensioned).

DY - derivative of first dependent variable followed by the remaining N-1 derivatives in the same order as the corresponding dependent variables (must be dimensioned).

F - the first cell of at least 7N cells required by DICE for storage (must be dimensioned).

L - a fixed point variable controlled by DICE to exit to the proper section of the program calling DICE.

INDEX - records the subscript of the variable having the greatest relative error.

I - order of integration (2, 3, or 4).

BOXA - statement number for derivative computations.

BOXB - statement number for supplementry computations after valid integration step. Checks for end-of-case based on conditions of dependent variables should be done in BOXB.

BOXC - statement number for print routine.

BOXD - statement number for non-convergence.

All floating point arguments are double precision.

The basic setup necessary to use the routine is as follows:

```
        .
        .
        .
        L = 4
S       CALL DICE(P, . . . . .,I)
        GO TO(BOXA,BOXB,BOXC,BOXD),L

BOXA    Evaluate N derivatives
        GO TO S
BOXB    Perform any necessary calculations after a valid integration step
        GO TO S
```

BOXC Print normal output

TP = TP+P

GO TO S

BOXD Dump or check print to determine reason for non-convergence

GO TO next case

This is a standard Westinghouse routine which has been used locally for several years.

C. SUBROUTINE DINTEG (T, DT, N, Y, DY, F, J, I)

PURPOSE: DINTEG is a double precision routine designed to be used in conjunction with DICE to provide a second, third, or fourth order Runge-Kutta solution of an Nth order system of linear and/or non-linear differential equations expressed as a system of N first order equations.

The basic method of the DINTEG routine can be summarized as follows: Given the initial conditions of the N dependent variables at time t_i, and their derivatives at time t_i, this routine will calculate a one-step approximation for each of the N dependent variables at time $t_i + h$, where h is the step size. Given the function

$$Y_n = f_n\ (t, Y_1, Y_2, Y_3, \ldots\ldots\ldots Y_n)$$

and the initial values

$$t_i;\ Y_{1,i};\ Y_{2,i};\ Y_{3,i};\ \ldots\ldots\ldots Y_{n,i})$$

then a second order Runge-Kutta solution gives a one-step approximation of:

$$Y_{n,i+1} = Y_{n,i} + 1/2(K_{o,n} + K_{1,n})$$

where:

$$K_{o,n} = hf_n\ (t_i;\ Y_{1,i};\ Y_{2,i};\ \ldots\ldots;\ Y_{n,i})$$

$$K_{1,n} = hf_n\ (t_{i+1};\ Y_{1,i} + K_{o,1};\ Y_{2,i} + K_{o,2};\ \ldots\ldots;\ Y_{n,i} + K_{o,n})$$

The third order Runge-Kutta solution gives an approximation of:

$$Y_{n,i+1} = Y_{n,i} + 1/6\ \ (K_{o,n} + 4K_{1,n} + K_{2,n})$$

where:

$$K_{o,n} = hf_n\ (t_i;\ Y_{1,i};\ Y_{2,i};\ \ldots\ldots\ldots;\ Y_{n,i})$$

$$K_{1,n} = hf_n\ (t_i + h/2;\ Y_{1,i} + 1/2\ K_{o,1};\ \ldots\ldots;\ Y_{n,i} + 1/2\ K_{o,n})$$

$$K_{2,n} = hf_n\,(t_i + h;\ Y_{1,i} + 2K_{1,1} - K_{o,1};\ Y_{2,i} + 2K_{1,2} - K_{o,2};\ \ldots;\ Y_{n,i} + 2K_{1,n} - K_{o,n})$$

and the fourth order Runge-Kutta solution gives a one-step approximation of:

$$Y_{n,i+1} = Y_{n,i} + 1/6\,(K_{o,n} + 2K_{1,n} + 2K_{2,n} + K_{3,n})$$

where

$$K_{o,n} = hf_n\,(t_i;\ Y_{1,i};\ Y_{2,i};\ \ldots\ldots;\ Y_{n,i})$$

$$K_{1,n} = hf_n\,(t_{i+h/2};\ Y_{1,i} + 1/2K_{o,1};\ Y_{2,i} + 1/2K_{o,2};\ \ldots;Y_{n,i}+1/2K_{o,n})$$

$$K_{2,n} = hf_n\,(t_{i+h/2};\ Y_{1,i} + 1/2K_{1,1};\ Y_{2,i} + 1/2K_{1,2};\ \ldots;Y_{n,i}+1/2K_{1,n})$$

$$K_{3,n} = hf_n\,(t_{i+h};\ Y_{1,i} + K_{2,1};\ Y_{2,i} + K_{2,2};\ \ldots;\ K_{n,i} + K_{2,n})$$

The arguments of the calling sequence are defined as:

T - independent variable - controlled by DINTEG

DT - step size - controlled by calling program

N - number of dependent variables

Y - dependent variables

DY - derivative of dependent variables

F - storage cells used by DICE and DINTEG (must be dimensioned)

J - fixed point variable controlled by DICE to enter proper section of DINTEG

I - order of integration (2, 3, or 4)

This is a standard Westinghouse routine which has been used locally for several years. In this program it is called exclusively by DICE.

D. FUNCTION RANNUM (SIGMA, XMEAN,I)

PURPOSE: To produce a normally distributed random number. A set of such numbers has the characteristics of a set of normally distributed random numbers with a given mean and standard deviation.

A uniformly distributed number U_j in the interval (0,1) is generated by UDRNRT. The normally distributed number, X_n, having a mean (m) and standard deviation (σ) is computed by the routine as follows:

$$V_i = \sqrt{-2 \log_e .5 (1-|1-2 U_j|)}$$

$$X_n = m + \sigma \left[\operatorname{sgn} (U_j - .5) \left(V_i - \frac{a_0 + a_1 V_i + a_2 V_i^2}{1 + b_1 V_i + b_2 V_i^2 + b_3 V_i^3} \right) \right]$$

where $a_0 = 2.515517$ $\quad b_1 = 1.432788$

$a_1 = .802853$ $\quad b_2 = .189269$

$a_2 = .010328$ $\quad b_3 = .001308$

The calling sequence is

XN = RANNUM (SIGMA, XMEAN,I)

where SIGMA is the standard deviation (floating point),

XMEAN is the mean of the sample (floating point),

I is the initial random number as a fixed point odd integer and

XN is the normally distributed random number.

This is a standard Westinghouse routine which has been used locally for several years.

E. FUNCTION UDRNRT (I)

PURPOSE: To produce a random number uniformly distributed in the interval (0,1) having a period of 2^{33}.

The routine is given a starter, I, from which it computes a uniformly distributed random number, U_j, in the interval (0,1) as follows:

$I_j = M\ I_{j-1} \pmod{2^{35}}$ where $M = 5^{13}$

Then $U_j = I_j/2^{35}$

The calling sequence is

UJ = UDRNRT(I) where

I is the initial random number as a fixed point <u>odd</u> integer and

UJ is the uniformly distributed random number (floating point)

This is a standard Westinghouse routine which has been used locally for several years.

F. FUNCTION ECCAN (EMM,E)

PURPOSE: To compute eccentric anomaly

The function subroutine ECCAN solves Kepler's equation

EA = EMM + E sin EA as follows:

$$E_o = EMM + E \sin EMM\,(1 + E \cos EMM) \qquad (1)$$

$$\Delta E = \frac{EMM - E_o + E \sin E_o}{1 - E \cos E_o} \qquad (2)$$

$$E_1 = E_o + \Delta E \qquad (3)$$

If E_o and E_1 agree to within $.5 \times 10^{-13}$, then E_1 is accepted as the eccentric anomaly, EA. Otherwise, E_o is replaced by E_1, a new E_1 is computed, and the iteration continues.

This is a standard Westinghouse routine which has been used locally for several years.

G. SUBROUTINE CEE (A1, A2, A3, XMAT)

PURPOSE: To perform coordinate transformations

The routine performs a coordinate transformation based on the angles A1, A2, A3 and returns the resulting 3 x 3 matrix, XMAT, to the calling program. It is used to compute the matrixes C, HC, G, and HG.

H. SUBROUTINE EXTRA (B, HB, G, HG, SPRM, HSPRM, HSDP)

PURPOSE: To compute and write the error matrix.

The routine computes the error matrix, $B^T \cdot HB$, and outputs it along with the matrix HB. During checkout, additional output was obtained from EXTRA by means of the last five arguments in the calling sequence. These arguments could now be dropped without introducing any ill effects.

I. SUBROUTINE CHECK1

PURPOSE: To determine whether to use numerical integration or closed-form solution for the transition matrix, whether to compute the B-matrix check and the C-matrix check, and to perform preliminary calculations on the B-matrix and C-matrix check if they are to be computed.

The conditions are examined for the closed-form solution of the transition matrix as well as for the B-matrix and C-matrix checks and the corresponding switches are set accordingly. If the B-matrix check is to be used, the matrix EM is computed for later use. If the criteria for the C-matrix check are met, further calculations produce PSI11, PSI13, PSI21, PSI23.

If the C-matrix check is to be omitted, the reason for omission will appear in the output. The interpretation of this reason may be found in Section IV.

The variables required by this routine are contained in the named COMMON regions GET, GIVE, and OMEGA.

J. SUBROUTINE HELP

PURPOSE: To compute the closed form solution of the transition matrix.

If the closed form solution for PH is to be used, the necessary partial derivatives are computed and used to calculate the transition matrix.

In either case, several other matrices are computed in this routine in order to keep the size of the main program within the bounds of the compiler.

The variables required by this routine are contained in the named COMMON regions GET, GIVE, OMEGA, and XCIN.

SECTION III. Input

All input to the program is made from data cards which are expected to be read from FORTRAN logical unit 5. Each case requires exactly 5 cards.

A. Makeup of Data Cards

<u>Data Card #1</u> Format 7F10.0

FIELD #1	OIZ	Orbital inclination angle	radians
FIELD #2	OOZ	Longitude of ascending node	radians
FIELD #3	OWZ	Argument of perigee in vehicle orbit	radians
FIELD #4	OAPZ	Semi-major axis of orbit	Astronomical units in sun orbiter case. Statute miles in earth orbiter case.
FIELD #5	OEZ	Eccentricity of orbit	
FIELD #6	OTPZ	Time of perigee relative to start time of simulation	hours
FIELD #7	CTP	Total duration of simulation	

<u>Data Card #2</u> Format 7F10.0

FIELD #1	VI(1)	Principal moment of inertia about x-axis	kilogram-meters^2
FIELD #2	VI(2)	Principal moment of inertia about y-axis	kilogram-meters^2
FIELD #3	VI(3)	Principal moment of inertia about z-axis	kilogram-meters^2
FIELD #4	VQ	Displacement from center of pressure to center of gravity	meters
FIELD #5	VLQ	Vehicular elevation of the displacement vector from c.p. to c.g.	radians

FIELD #6	VPQ	Vehicular azimuth of the displacement vector from c.p. to c.g.	radians
FIELD #7	VAV	Effective area of the vehicle	square meters

Data Card #3 Format 7F10.0

FIELD #1	OBZD	Minimum allowable angle between sun line and sensor slit for which a sun sensor measurement will be allowed	radians
FIELD #2	OBYD	Maximum allowable deviation from right angle for which a sun sensor measurement will be allowed	radians
FIELD #3	SIGZ	Standard deviation of initial angular position uncertainty in each axis	radians
FIELD #4	SIGW	Standard deviation of initial angular rate uncertainty in each axis	radians/second
FIELD #5	SIGS	Standard deviation of sun sensor error uncertainty in each axis	radians
FIELD #6	SIGB	Standard deviation of magnetometer error uncertainty in each axis	webers/sq. meter
FIELD #7	TSO	Initial angular displacement of sun beyond vernal equinox (applies only to earth orbiter)	radians

Data Card #4 Format 7F10.0

FIELD #1	ZI	Initial local Euler angle displacement about roll axis	radians
FIELD #2	Z2	Initial local Euler angle displacement about pitch axis	radians
FIELD #3	Z3	Initial local Euler angle displacement about yaw axis	radians
FIELD #4	WO1	Initial angular rate about roll axis	radians
FIELD #5	WO2	Initial angular rate about pitch axis	radians
FIELD #6	WO3	Initial angular rate about yaw axis	radians
FIELD #7	PBO	Initial celestial longitude of the earth's magnetic pole (applies only to earth orbiter)	radians

Data Card #5 — Format 7I10

FIELD #1	NS	Number of sun sensors (12 is the upper limit)
FIELD #2	M	Total number of measurement intervals
FIELD #3	NZ	=0 sun orbiter is simulated =1 earth orbiter is simulated
FIELD #4	NT	=0 transition matrix computed under torque free conditions =1 transition matrix computed with torques included
FIELD #5	NRS	Random number starter; must be a positive, odd integer
FIELD #6	not used	
FIELD #7	not used	

B. Illegal Practices - There are several input conditions which will result in erroneous results or will cause the program to halt. These inputs usually involve unrealistic or contradictory combinations of parameters.

1) Program stops

a) Semi-major axis of orbit less than the radius of earth for an earth satellite.

b) Nonrealizable combination of inertias (I_1, I_2, I_3). (See Reference 28 of Volume I)

2) Practices leading to erroneous results

a) $N_t = 0$ with high spin rates causes an inaccurate transition matrix.

b) Significant oscillation angle rates with $N_c = 1$. (See condition 7 of Appendix F, Volume I.)

c) High eccentricity for a sun orbiter introduces error in the solar pressure calculation.

d) Either ω_{o1} or both ω_{o2} and ω_{o3} chosen equal to zero causes inaccuracies in the closed form solution of the transition matrix. The zero condition may be satisfactorily simulated by giving the quantities a value on the order of 10^{-12}.

C. Inputs for Sample Case - The following are the input quantities used in obtaining the output seen in Section IV.

Col. 10	Col. 20	Col. 30	Col. 40	Col. 50	Col. 60	Col. 70
0.	0.	0.	4000.	0.	0.	.16666666
150.	200.	100.	0.	0.	0.	0.
1.	.7854	.1	.0001	.01	.000001	0.
.05	.03	.04	-.000049	.00122	.000061	0.
12	20	1	1	101		

SECTION IV. OUTPUT

All output appears on the printer (FORTRAN logical unit 6). There is no need for any operator action such as tape handling, on-line messages or key-ins.

A. Normal output for every run

1) Inputs - The 33 input values are output and labeled, both with their specific name and under the more general headings of ORBIT, VEHICLE, OBSERVATION, STATISTICAL, INITIAL CONDITIONS, and CONTROL.

2) Initial conditions - Prior to entering the integration loop, the initial values of several quantities are seen.

a) CROSI - Cross product of vectors POLI and VERI

b) POLI - Orbit pole unit vector in inertial coordinates

c) VERI - Local vertical unit vector in inertial coordinates

d) SUNI - Unit vector along sunline in inertial coordinates (for earth orbiter only)

e) MAGI - Unit vector along magnetic field in inertial coordinates (for earth orbiter only)

f) CROSV - Cross product of vectors POLV and VERV

g) POLV - Orbit pole unit vector in vehicle coordinates

h) VERV - Local vertical unit vector in vehicle coordinates

i) SUNV - Unit vector along sunline in vehicle coordinates (for earth orbiter only)

j) MAGV - Unit vector along magnetic field in vehicle coordinates (for earth orbiter only)

k) XA - Actual position and velocity of the vehicle

l) XO - Difference between actual and observed position and velocity. Position components are initialized at every interval for both actual and observed values. Thus, the the differences in position are zero at the start of the simulation.

m) C-MATRIX CHECK OMITTED DUE TO XX - If the conditions for the computation of the C-matrix check have not been met, the number of the first test which failed appears in this message. There are twelve such tests arranged in the following order:

i. Earth orbiter

ii. Effective area of the vehicle equals zero

iv. $z_1^2 + z_2^2 + z_3^2 \leq .01$

v. $|\omega_{02} - \eta_0| \leq 0.1 \sqrt{3\xi_2}$

vi. $I_1 > I_3$

vii. $\xi_1 \xi_3 < 0$

viii. $F_1 = (1-3\xi_1 - \xi_1 \xi_3) > 0$

ix. $F_2 = F_1^2 + 16 \xi_1 \xi_3 > 0$

x. $Z_{11} \leq .1;\ Z_{13} \leq .1;\ Z_{21} \leq .1;\ Z_{23} \leq .1$

xi. $|\omega_{01} + \eta_0 z_3| < .01 (r_1 Z_{11} + r_2 Z_{21})$

xii. $|\omega_{03} - \eta_0 z_1| < .01 (r_1 Z_{13} + r_2 Z_{23})$

3) Periodic readouts - The frequency of this type of output is determined by the user and is equal to the duration of the simulation in seconds (3600 times CTP) divided by the number of iterations (M)

a) TIME - Number of seconds since start of simulation

M - Iteration count

GX, GY, GZ - Roll, pitch, and yaw angular displacement since last interval

b) VERI - Current local vertical unit vector in inertial coordinates

c) CROSV - Current cross product of vectors POLV and VERV. If the C-matrix check has been computed, the fourth number on this line represents the closed-form solution for the second value in the cross product. Otherwise, it is set to zero.

d) POLV - Current orbital pole unit vector in vehicle coordinates. If the C-matrix check has been computed, the fourth number on this line represents the closed-form solution for the third component of the vector. Otherwise, it is set to zero.

e) VERV - Current local vertical unit vector in vehicle coordinates. If the C-matrix check has been computed, the fourth number on this line represents the closed-form solution for the first component of the vector. Otherwise, it is set to zero.

f) SUNV - Current unit vector along sunline in vehicle coordinates (for earth orbiter only).

g) MAGV - Current unit vector along magnetic field in vehicle coordinates (for earth orbiter only).

h) TRANSITION MATRIX - The 6 x 6 transition matrix appears under the heading .

i) B-MATRIX - The 3 x 3 B matrix appears under the heading .

j) B-MATRIX CHECK - The 3 x 3 matrix appears under the heading. If the B-matrix check has been computed, the matrix represents the closed form solution for the B-matrix. In a torque free case, it should equal the B-matrix; they should be approximately equal when torques are present. The matrix is set to zero if the computation of the B-matrix check has been omitted.

k) XA - Actual position and velocity of the vehicle. The two numbers on the next line represent the magnitudes of the position and velocity vectors.

l) XR - Square root of the diagonal elements of the 6 x 6 uncertainty covariance matrix. The two numbers on the next line are the square root of the sum of the first three and last three values on the above line.

m) XP - Errors in Euler angles and angular rates. Since these are referenced to the coordinate frame corresponding to the preceding interval, the Euler angle uncertainties take on a meaning less significant than the off-diagonal terms of the error matrix (to be discussed later in this section). The two numbers on the next line are the magnitudes of the position and velocity vector errors respectively.

4. Instrument readouts - This output occurs at every iteration for each magnetometer which is in position with respect to the magnetic field and for each sun sensor which is in position with respect to the sunline. If no instrument is in position, this type of output is omitted for the current iteration.

   a) The first six lines are not labeled since they were originally intended only for debug purposes but were later deemed useful. They represent the 6 x 6 uncertainty covariance matrix.

   b) INST. NO. = XX. For a magnetometer readout, instruments 1, 2, and 3 indicate roll, pitch, and yaw respectively. The sun sensors are arranged as follows:

Instrument Number	Field of View Axis	Direction of Slit
1	+ Roll	Pitch
2	+ Pitch	Yaw
3	+ Yaw	Roll
4	- Roll	Pitch
5	- Pitch	Yaw
6	- Yaw	Roll
7	+ Roll	Yaw
8	+ Pitch	Roll
9	+ Yaw	Pitch
10	- Roll	Yaw
11	- Pitch	Roll
12	- Yaw	Pitch

YA - In magnetometer readout, actual earth dipole flux density in webers/sq. meter.

- In sun sensor readout, actual measured value of angle Y (defined in Volume I).

NOTE: The first three sun sensor readouts are easily distinguished from the magnetometer readouts because the value of YA for the magnetometers is on the order of 10^{-5} while its value for the sun sensors normally ranges from 2.0 to 0.1.

H1A, H2A, H3A- Partial derivatives of the observable with respect to the rotational position state variables as computed from the actual vehicle orientation. These three values currently appear in the output as zero because the necessary calculations overloaded the compiler. The equations may be added as a subroutine at any time.

c) NR - Random number used in the computation of the observed position and velocity vectors.

YP - In magnetometer readout, predicted earth dipole flux density in webers/sq. meter.

- In sun sensor readout, predicted measured value of angle Y.

H1,H2,H3 - Partial derivatives of the observable with respect to the rotational position state variables as computed from the predicted vehicle orientation.

d) XR - Same as the XR discussed above with the exception that it is now computed from uncertainty covariance matrix, which is updated after each instrument readout.

e) XO - Difference between the actual and observed position and velocity vectors. The two numbers of the next line are the magnitudes of the position and velocity vector differences respectively.

f) W - Vector of optimum linear estimator coefficients.

5. End of Iteration - After each iteration, even when there was no instrument readout, the following printout occurs;

a) HB-MATRIX - Observed orthogonal transformation from vehicle to fixed inertial coordinates.

b) ERROR MATRIX - Product of B^{-1} times HB.

NOTE: Since the observed transformation and the actual transformation in this case are defined relative to the same inertial reference, the error matrix contains the net effect of all attitude errors present up to the current interval.

c) The number that appears under the error matrix is the square root of the sum of the squares of terms (1,2), (2,3), and (3,1) of the error matrix where the first index is the row indicator and the second is the column indicator.

B. Diagnostic Messages

The program checks for certain undesirable conditions and outputs an explanatory message when one of these situations is detected. It then goes on to the next case.

1) Errors caused by illegal input

a) TEST = XXXX WHEN SR = XXXX. The current value of TEST would result in attempting to compute the square root of a negative number. This erroneous value can only occur when the semi-major axis of the orbit (input variable OAPZ) is less than the radius of the earth in statute miles.

b) INPUT MOMENTS OF INERTIA NOT REALIZABLE. The difference between any two moments of inertia cannot be greater than the third in a rigid body. The moments of inertia (inputs VI(1), VI(2), VI(3)) are printed out following the message.

2) Unrealistic trigonometric values

Several places throughout the program, an arccosine is calculated from the computed value of a cosine. Clearly, these cosines cannot be greater than 1. The following messages were useful in debugging the mathematical model as well as the program itself. They have remained in the program in order to prohibit analysis with meaningless data should some malfunction occur.

a) N = XX. LOOP = XXX, SL1U = XXX. N is the instrument number; LOOP is the iteration count and SL1U is an intermediate variable in the computation of YA (actual angle between sun line and instrument) and must be less than 1.

b) N = XX. LOOP = XXX, SL1N = XXX, ZETAS = XXX. N and LOOP are as above; SL1N divided by ZETAS forms the cosine of YA and must be less than 1.

c) N = XX. LOOP = XXX, SDPU = XXX. N and LOOP are as above; SDPU is an intermediate value in the computation of YP (predicted angle between sun line and instrument) and must be less than 1.

d) N = XX. LOOP = XXX, SDPL = XXX, HZS = XXX. N and LOOP are as above; SDPL divided by HZS forms the cosine of YP and must be less than 1.

e) TROUBLE IN ARCCOS. The cosine of α_{hv} has been calculated to be greater than 1 in subroutine CHECK1. In addition to the message, the ten quantities that go into the computation are also printed out.

3. If the integration routine cannot converge, a PDUMP results.

C. Output from Sample Case

The following pages contain a portion of the results generated from the data cards listed in Section III. All the preliminary output is shown as well as the periodic and instrument readouts from the first and last (twentieth) iteration.

```
*** INPUTS ***

ORBIT  OIZ=       0.            OOZ=       0.            OWZ=       0.            OATP=    4000.00000000
       OEZ=       0.           OTPZ=       0.

VEHICLE I1=    150.00000000     I2=    200.00000000     I3=    100.00000000     VQ=        0.
       VLQ=       0.            VPQ=       0.           VAV=       0.

OBSERVATION NS=  12 OBZD=      1.00000000 OBYD=      0.78540000  M=      20

STAT. SIGZ=       0.09999999 SIGW=       0.00009999 SIGS=       0.01000000 SIGB=       0.00000100

INIT.   Z1=       0.05000000  Z2=       0.02999999   Z3=       0.04000000  PBO=       0.
       WO1=      -0.00004900 WO2=       0.00122000  WO3=       0.00006100  TSO=       0.

CONTROL NZ=    1 NT=     1  CTP=      0.16666666  NRS=  101

CROSI     -0.            1.00000000      0.

 POLI      0.           -0.              1.00000000

 VERI     1.00000000     0.              0.

 SUNI      1.00000000     0.             0.

 MAGI      0.19252197     0.             0.98129266

CROSV     0.99675049     0.04143731     -0.02793542

 POLV     -0.03997134     0.99789141     0.05113719

 VERV      0.02999549    -0.04995668     0.99830085

 SUNV     0.02999549    -0.04995668     0.99830085

 MAGV     -0.03344879     0.96960577     0.24237540

  XA       0.             0.             0.             -0.00004900     0.00122000     0.00006100

  XO      0.             0.             0.              0.00011010     0.00005222    -0.00000307
```

```
TIME=     29.99999833  M=    1  GX=     -0.00130654  GW=      0.03660594  GZ=      0.00181516

 VERI        0.99932850        0.03664070        0.

CROSV        0.99875040        0.04146658       -0.02789525        C11            0.03998602       C13

 POLV       -0.04000519        0.99789449        0.05105063        C21            C22      0.04991054

 VERV        0.02995341       -0.04987088        0.99830641        0.03003373     C32              C33

 SUNW       -0.00665623       -0.05135416        0.99865832

 MAGV       -0.04011849        0.96935594        0.24236245

                          TRANSITION MATRIX

        0.99942919     0.00176237     0.00018770   -29.99184489    -0.04387658    -0.95734597
       -0.00174256     0.99951407    -0.00003576     0.03224137   -29.99512768     0.03712121
        0.03836773     0.00013714     0.99993780    -0.28855269    -0.01648527   -30.01691461
        0.00008677    -0.00000077    -0.00001305     0.99845108     0.00113906     0.02566102
       -0.00000030     0.00003228     0.00000238    -0.00041799     0.99951552    -0.00052785
       -0.00000937    -0.00000469     0.00000106    -0.01897722    -0.00091541     0.99974332

                 B-MATRIX                                          B-MATRIX CHECK

        -0.00666162     -0.05135676      0.99865815     0.          0.          0.
         0.99917726      0.03961143      0.00870213     0.          0.          0.
        -0.04000519      0.99789449      0.05105063     0.          0.          0.

 XA       0.00134744     -0.03661566     -0.00181638     -0.00004393      0.00122132      0.00006174

              0.03663644                                 0.00122334

 XR         0.09998811       0.09999655      0.10011737       0.00010026       0.00010000       0.00009999

              0.17326402                                 0.00017335

  XP        -0.00333004      -0.00161228      0.00006863       0.00011183       0.00005545      -0.00003566

              0.00370046                                 0.00012495

         0.00999762      0.00000020      0.00038571      0.00000057     -0.00000000     -0.00000010

         0.00000020      0.00999931      0.00000034     -0.00000001      0.00000002     -0.00000004

         0.00038571      0.00000034      0.01002348     -0.00000010      0.00000002     -0.00000029

         0.00000057     -0.00000001     -0.00000010      0.00000001      0.00000000      0.00000000

        -0.00000000      0.00000002      0.00000002      0.00000000      0.00000001     -0.00000000

        -0.00000010     -0.00000004     -0.00000029      0.00000000     -0.00000000      0.00000001

INST. NO.=  1       YA=      0.00000171  H1A=     -0.00000000  H2A=     -0.00000000  H3A=     -0.00000000

NR=  -0.00000071  WP=      0.00000188  H1 =      0.          H2 =      0.00000919  H3 =      0.00002994

XR        0.09992648        0.09602190       0.04130232       0.00010026       0.00010000       0.00009996

              0.14460771                                 0.00017333

XO      -0.00427401      -0.00911893      -0.02445851       0.00011210       0.00005537      -0.00000491
```

```
          0.02645072                                     0.00012513

      1066.31451416    8479.58911133   27706.13891602   -0.30600432    0.08498897   -0.84913993
       0.00998530     -0.00009777      0.00006559      0.00000057   -0.00000000   -0.00000008
      -0.00009777      0.00922021     -0.00254530      0.00000001    0.00000001    0.00000003
       0.00006559     -0.00254530      0.00170588     -0.00000001   -0.00000000   -0.00000004
       0.00000057      0.00000001     -0.00000001      0.00000001    0.00000000    0.00000000
      -0.00000000      0.00000001     -0.00000000      0.00000000    0.00000001   -0.00000000
      -0.00000008      0.00000003     -0.00000004      0.00000000   -0.00000000    0.00000001
INST. NO.=  3        YA=     1.51941842  H1A=    -0.00000000  H2A=    -0.00000000  H3A=    -0.00000000
NR=   -0.00518899  YP=     1.49842587  H1 =    -0.99999999  H2 =    -0.00621978  H3 =     0.12808057
XR        0.01104067     0.09597985      0.04126818      0.00010009    0.00010000    0.00009995
          0.10510035                                     0.00017323
XO       -0.01988449    -0.00956579     -0.02419449      0.00011121    0.00005538   -0.00000478
          0.03274555                                     0.00012433
W        -0.98778310    -0.02827623      0.01670609     -0.00005675    0.00000034    0.00000818
       0.00013088     -0.00037986      0.00023226      0.00000000   -0.00000000   -0.00000000
      -0.00037986      0.00921213     -0.00254053      0.00000000    0.00000001    0.00000003
       0.00023226     -0.00250053      0.00170306     -0.00000000   -0.00000000   -0.00000000
       0.00000000      0.00000000     -0.00000000      0.00000001    0.00000000    0.00000000
      -0.00000000      0.00000001     -0.00000000      0.00000000    0.00000001   -0.00000000
      -0.00000000      0.00000003     -0.00000004      0.00000000   -0.00000000    0.00000001
INST. NO.=  9        YA=     1.57746141  H1A=    -0.00000000  H2A=    -0.00000000  H3A=    -0.00000000
NR=    0.01330772  YP=     1.48859671  H1 =    -0.00464664  H2 =    -0.99878273  H3 =    -0.07463337
XR        0.01078133     0.01044849      0.03243642      0.00010009    0.00010000    0.00009995
          0.03574254                                     0.00017323
XO       -0.01574127    -0.11283800      0.00342166      0.00011119    0.00005520   -0.00000514
          0.11398206                                     0.00012424
W         0.04053539    -1.01036856      0.27018404     -0.00000025   -0.00000168   -0.00000349
                 HB-MATRIX                                            ERROR MATRIX
      -0.01891927     -0.05702077      0.99819371      0.99970139   -0.02180577    0.01102854
       0.99964295      0.01776277      0.01996142      0.02173974    0.99974522    0.00607175
      -0.01886890      0.99821496      0.05666436     -0.01115812   -0.00583018    0.99992075

                                                                   0.02523612
```

```
TIME=   599.99996948  M=   20  GX=     0.00152341  GY=     0.03713800  GZ=     0.00175871
VERI      0.74318521      0.65908597      0.
CROSV     0.99919731      0.03270868     -0.02312732      C11          0.03078592     C13
POLV     -0.03225874      0.99928799      0.01956759      C21          C22        0.01889952
VERV      0.02375088     -0.01880582      0.99954101      0.02506999   C32            C33
SUNV     -0.65081599     -0.03581221      0.75839042
MAGV     -0.15046063      0.97379526      0.17054145
                         TRANSITION MATRIX
   0.99971369    0.00018587   -0.00062907  -29.99529386   -0.00510046   -0.34125425
  -0.00022113    0.99986196   -0.00001293    0.00372640  -29.99867368    0.03838499
   0.03334866    0.00003419    0.99965001   -0.24717700   -0.01569536  -30.00974488
   0.00005635    0.00000174    0.00004130    0.99898001    0.00008047    0.02191122
   0.00000068    0.00000384    0.00000038   -0.00005003    0.99987298   -0.00052549
   0.00003141   -0.00000182    0.00002290   -0.01737799   -0.00099773    0.99945277

              B-MATRIX                                       B-MATRIX CHECK
   -0.65039761    -0.03586113     0.75831806     0.            0.            0.
    0.75247984     0.01172589     0.65159100     0.            0.            0.
   -0.03225874     0.99928799     0.01956759     0.            0.            0.

XA      -0.00152447   -0.03714396   -0.00175992    0.00005131    0.00123856    0.00005724
             0.03721686                            0.00124095
XR       0.00857292    0.00478179    0.01344602    0.00001847    0.00001387    0.00003858
             0.01664800                            0.00004496
XP      -0.00052679    0.00332821    0.00154828    0.00001816   -0.00011926   -0.00005124
             0.00370833                            0.00013107
     0.00007349     0.00000521    -0.00005759    -0.00000013    -0.00000000     0.00000011
     0.00000521     0.00002286    -0.00001269    -0.00000000    -0.00000005     0.00000003
    -0.00005759    -0.00001269     0.00018079     0.00000004     0.00000002    -0.00000045
    -0.00000013    -0.00000000     0.00000004     0.00000000    -0.00000000    -0.00000000
    -0.00000000    -0.00000005     0.00000002    -0.00000000     0.00000000    -0.00000000
     0.00000011     0.00000003    -0.00000045    -0.00000000    -0.00000000     0.00000000
INST. NO.=  1      YA=     0.00000480  H1A=    -0.00000000  H2A=    -0.00000000  H3A=    -0.00000000
NR=   0.00000151  YP=     0.00000527  H1 =      0.          H2 =     0.00000850  H3 =     0.00002907
XR       0.00843687    0.00477902    0.01255819    0.00001842    0.00001386    0.00003665
             0.01586594                            0.00004330
XO      -0.00201473    0.00316899    0.00624797    0.00001945   -0.00011893   -0.00006302
```

```
           0.00728953                                      0.00013600

   -1419.68252563  -151.91398811  4484.07849121     1.23030491     0.32222372   -11.24645090

      0.00007118     0.00000496    -0.00005028    -0.00000013    -0.00000000     0.00000009

      0.00000496     0.00002284    -0.00001191    -0.00000000    -0.00000005     0.00000003

     -0.00005028    -0.00001191     0.00015771     0.00000004     0.00000002    -0.00000039

     -0.00000013    -0.00000000     0.00000004     0.00000000    -0.00000000    -0.00000000

     -0.00000000    -0.00000005     0.00000002    -0.00000000     0.00000000    -0.00000000

      0.00000009     0.00000003    -0.00000039    -0.00000000    -0.00000000     0.00000000

INST. NO.=  9        YA=     2.28000534  H1A=    -0.00000000  H2A=    -0.00000000  H3A=    -0.00000000

NR=  -0.00159872  YP=     2.27273029  H1 =     0.02339360  H2 =    -0.99989582  H3 =    -0.03198076

XR       0.00843549     0.00433038     0.01254765     0.00001842     0.00001290     0.00003661

           0.01572747                                      0.00004297

XO      -0.00209326     0.00213057     0.00651231     0.00001949    -0.00011633    -0.00006379

           0.00716459                                      0.00013410

W       -0.01383517    -0.18293860     0.04656979     0.00000601     0.00045828    -0.00013521

                 HB-MATRIX                                            ERROR MATRIX

     -0.64613497    -0.03636406     0.76235638     0.99980085     0.01918313    -0.00550101
      0.76149011     0.03655954     0.64714465    -0.01926733     0.99969132    -0.01568504
     -0.05140421     0.99866964     0.00406846     0.00519842     0.01578791     0.99986184

                                                                   0.02531870
```

APPENDIX A: GLOSSARY

The purpose of this glossary is to relate program names to the engineering symbols defined in Volume I and to define those variables which are introduced in the program itself. It is meant to contain all major parameters of interest as well as most of the intermediate variables.

The Glossary in Volume I is referenced for all variables defined in that volume. Section III of Volume II (Input) is referenced for the description of all input quantities.

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
A(I,J)	$a_{i,j}$	Volume I
ABC	A_c	Cosine of $\omega_A \tau$
ABSI	A_s	Sine of $\omega_A \tau$
ABETA	$\lvert\beta\rvert$	Magnitude of magnetic flux density
ACP	A'_c	Cosine of $(\omega'_A t + \alpha_{Kh})$
AHV	α_{hv}	Angle between the angular momentum vector and vector V
ALSC		Cosine of angle between spin axis and angular momentum vector as computed from observed attitude (for transition matrix)
ALSS		Sine of angle between spin axis and angular momentum vector as computed from observed attitude (for transition matrix)
ASCP	α_c	Cosine of angle between spin axis and angular momentum vector
ASKH	α_{kh}	Angle between initial yaw axis and angular momentum vector
ASKV	α_{kv}	Angle between initial yaw axis and vector V
ASSP	α_s	Sine of angle between spin axis and angular momentum vector
ASUBK	α_k	Difference between α_{kv} and α_{kh}
ASUBM	A_m	Mean anomaly of satellite
ASP	A'_s	Sine of $(\omega'_A t + \alpha_{Kh})$
AZ	a_o	Semi-major axis in meters
B(I,J)	$B_{i,j}$	Volume I
BBC	B_c	Cosine of $\omega_B \tau$
BBS	B_s	Sine of $\omega_B \tau$
BCHK (I,J)		B-check matrix
BCP	B'_c	Cosine of $(\omega'_B t - \alpha_K)$

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
BDUM(I,J)		Intermediate array used in the computation of BDUM
BETA(I)	β_i	Volume I
BETAP(I)	β''_i	Volume I
BSP	B'_s	Sine of $(\omega'_g t - \alpha_k)$
BTSP(I)	β_i	Volume I
C(I,J)	$C_{i,j}$	Volume I
CAHV		Cosine of α_{hv}
CAK		Cosine of α_k
CAKH		Cosine of α_{kh}
CAPEZ		Eccentric anomaly
CCHK1		First term in the C-matrix check
CCHK2		Second term in the C-matrix check
CCHK3		Third term in the C-matrix check
CHX(I)		Cosines of the observed position state variables
CIC	i_c	Volume I
CIS	i_s	Volume I
COMC	Ω_c	Cosine of Ω_o
COSAKV		Cosine of α_{kv}
CSUBB	c_β	Earth magnetic field constant
CSUBH	c_h	Solar measurement sensitivity constant
CTHDS	$\dot{\theta}_s$	Time derivative of solar true anomaly
CTHE		Cosine of θ
CTP	T'	Input

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
CUBM	$[\mu_B \mid m \mid]$	Earth magnetic field constant
CUC		Cosine of $(\omega_o + \theta_o)$
CWS		Volume I
D(I,J)	$D_{i,j}$	Volume I
DELA(I)		Variables of differentiation in transition matrix derivation
DELBA(I)		Variables of differentiation in transition matrix derivation
DELF1(I)		Variables of differentiation in transition matrix derivation
DELF2(I)		Variables of differentiation in transition matrix derivation
DELG1(I)		Variables of differentiation in transition matrix derivation
DELG2(I)		Variables of differentiation in transition matrix derivation
DELG3(I)		Variables of differentiation in transition matrix derivation
DELK(I)		Variables of differentiation in transition matrix derivation
DELL(I)		Variables of differentiation in transition matrix derivation
DELX1(I)		Variables of differentiation in transition matrix derivation
DELX2(I)		Variables of differentiation in transition matrix derivation
DELX3(I)		Variables of differentiation in transition matrix derivation
EL(I,J)	$L_{i,j}$	Volume I
EL1(I,J)	$L1_{i,j}$	Volume I
EM(I,J)	M(I,J)	Intermediate array in the computation of B-matrix check

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
ERR1		Lower error limit for the integration routine
ERR2		Upper error limit for the integration routine
ETAZ	η_o	Volume I
FG(I)	f_{γ_i}	Scalar products in non-homogeneous transition matrix derivation
FS(I)	f_{s_i}	Scalar products in non-homogeneous transition matrix derivation
F1	F_1	Intermediate variable in closed form transition matrix solution
F2	F_2	Intermediate variable in closed form transition matrix solution
F7N(I)		Storage array required by DICE
G(I,J)	$G_{i,j}$	Volume I
GAM(I,J)	$\Gamma_{i,j}$	Volume I
GNU(I,J)	$\nu_{i,j}$	Volume I
HB(I,J)	$\hat{B}_{i,j}$	Volume I
HBTSP(I)	$\hat{\beta}_i$	Volume I
HC(I,J)	$\hat{C}_{i,j}$	Volume I
HCAPF(I,J)	$\hat{F}_{i,j}$	Parameters in the observed equations of motion
HCAPN(I)	N_i	Volume I
HCSUBH	$\hat{c}_h$	Solar measurement sensitivity constant using observed quantities
HDUM(I)		Intermediate array used in the calculation of PP
HFG(I)	$\hat{f}_{\gamma_i}$	Scalar products in non-homogeneous transition matrix (computed from observed values)
HFS(I)	$\hat{f}_{s_i}$	Scalar products in non-homogeneous transition matrix (computed from observed values)

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
HG(I,J)	$\hat{G}_{i,j}$	Volume I
HGAM(I,J)	$\hat{\Gamma}_{i,j}$	Volume I
HGNU(I,J)	$\hat{\nu}_{i,j}$	Volume I
HIA(I)	H_iA	Measurement sensitivities computed from actual attitude (Actual computations not in present program)
HH(I)	$\hat{H}_i$	Volume I
HLOWF(I,J)	$\hat{f}_{i,j}$	Parameters in the observed equations of motion
HSDP(I)	$\hat{s}''_i$	Volume I
HSPRM(I)	$\hat{s}'_i$	Volume I
HU	$\hat{\mu}$	Volume I
HWO1	$\hat{\omega}_{o1}$	Observed angular rate about roll axis
HWO2	$\hat{\omega}_{o2}$	Observed angular rate about pitch axis
HWO3	$\hat{\omega}_{o3}$	Observed angular rate about yaw axis
HX(I)	$\hat{X}_i$	Observed position and velocity state variables
HXD(I)	$\hat{\dot{X}}_i$	Derivatives of observed position and velocity state variables
HZETAS	$\hat{\zeta}_s$	Sine of the observed angle, ζ
HZ1	$\hat{Z}_1$	Observed local Euler angle displacement about roll axis
HZ2	$\hat{Z}_2$	Observed local Euler angle displacement about pitch axis
HZ3	$\hat{Z}_3$	Observed local Euler angle displacement about yaw axis
INDEX		Index of the most widely varying dependent variable
IORD		Order of integration

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
IWHY		Indicator of the reason for omission of the C-matrix check
KTR		Magnetometer counter
LOOP	m	Iteration counter
LQQQ		Control variable in the integration routine
M	M	Input
MAG		Switch set in program =1; use magnetometers =2; skip magnetometers
N		Sun sensor index
NRS	NRS	Input
NS	N_s	Input
NSUBB	N_b	=1; compute the B-matrix check =2; omit the B-matrix check
NSUBC	N_c	=1; compute the C-matrix check =2; omit the C-matrix check
NSUBM	N_m	=1; compute closed form transition matrix =2; integrate to solve for the transition matrix
NT	N_t	Input
NVAR		Number of dependent variables
NZ	N_o	Input
OAPZ	a_o	Input
OBYD	Y_Δ	Input
OBZD	$\int_\Delta$	Input
OEZ	e_o	Input
OIZ	i_o	Input
OOZ	Ω_o	Input
OSA	ω_A	Spin rate as computed from observed state

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
OSB	ω_B	Precession as computed from observed state
OTPZ	t_o'	Input
OWZ	ω_o	Input
OYZ	ω_{yz}	$\sqrt{\omega_2^2 + \omega_3^2}$
PBO	ψ_{β_o}	Input
PH(I,J)	$\phi_{i,j}$	Transition matrix; Volume I
PHD(I,J)	$\dot{\phi}_{i,j}$	Derivative of transition matrix
PI	π	Pi, to 16 significant digits
PM(I,J)	$P^{(-)}{}_{i,j}$	Volume I
PMDUM(I,J)		Intermediate array used in the computation of PM
PP(I,J)	$P^{(+)}{}_{i,j}$	Volume I
PSIB	ψ_β	Volume I
PSI2	ψ_2	Volume I
PSI11	ψ_{11}	Volume I
PSI13	ψ_{13}	Volume I
PSI21	ψ_{21}	Volume I
PSI23	ψ_{23}	Volume I
PSUBC	P_c	Cosine of $\omega_p t$
PSUBP	P_p	Volume I
PSUBS	P_s	Sine of $\omega_p t$
PZ	P_o	Parameter of orbit
Q(I)	q_i	Volume I
QKB(I)	k_{β_i}	Volume I
QKBDR	$K_\beta \cdot R$	Dot product of K_β and R

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
QLBC	λ_{bc}	Cosine of λ_b
QQ(I,J)		Intermediate array in the calculation of the B-matrix check
QSC	k_c	Cosine of $\hat{\omega}_{o3}/\omega_{yz}$
QSS	k_s	Sine of $\hat{\omega}_{o2}/\omega_{yz}$
R(I)	r_i	Volume I
RANDOM	N_R	Random number
RHO	ρ	Volume I
RHOX(I)	pXi	Predicted state
RR(I,J)		Intermediate array in the computation of the B-matrix check
RT1	r_1	Volume I
RT2	r_2	Volume I
R11	R_{11}	Parameter in libration analysis
R13	R_{13}	Parameter in libration analysis
R21	R_{21}	Parameter in libration analysis
R23	R_{23}	Parameter in libration analysis
R31	R_{31}	Parameter in libration analysis
R33	R_{33}	Parameter in libration analysis
R41	R_{41}	Parameter in libration analysis
R43	R_{43}	Parameter in libration analysis
SAHV		Sine of α_{hv}
SAK		Sine of α_k
SAKH		Sine of α_{kh}
SDPL(I)	$\hat{s}'' \cdot L$	Dot product of $\hat{s}''$ and L
SDPU(I)	$\hat{s}'' \cdot U$	Dot product of $\hat{s}''$ and U

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
SHX(I)		Sines of the observed position state variables
SIC	i_f	Sine of i_o
SIGB	σ_B	Input
SIGS	σ_S	Input
SIGW	σ_W	Input
SIGZ	σ_Z	Input
SOMC	Ω_c	Sine of Ω_o
SPRM(I)	s'_i	Volume I
SR	r	Volume I
SSRL(I)	s_i	Volume I
SSP(I)	s''_i	Volume I
STHE		Sine of θ
SX(I)		Sines of the position state variables
SUF	μ_f	Sine of $(\omega_o + \theta_o)$
T	T	Duration of simulation in seconds
TAU	τ	Interval between measurements
TERM1		Intermediate variable used in several calculations
TERM2		Intermediate variable used in several calculations
TESL		Intermediate value used in the test for shadow
TESR		Intermediate value used in the test for shadow
TEST		Intermediate value used in the test for shadow
THES	θ_s	Volume I

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
THETA	θ	Volume I
THETAL	θ_{m-1}	Volume I
TM	t	Time;independent variable in the integration
TP		Box C entry time; see Volume II, Section II, B
TSO	θ_{so}	Input
TZ	t_o	Time at perigee in seconds (relative to reference time at start of simulation)
U	μ	Volume I
UA(I,J)	$U_{i,j}$	Volume I
VAV	A_v	Input
V(I)		Vector about which the angular momentum precesses
VDP(I)		Vector V in vehicle coordinates
VI(I)	I_i	Input
VLQ	λ_Q	Input
VPQ	ψ_Q	Input
VQ	q	Input
VXH		Magnitude of the cross product of V and H
W(I)	W_i	Volume I
WDUM(I)		Intermediate array used in the computation of W
WSAP	ω'_A	Volume I
WSBP	ω'_B	Volume I
WSUBP	ω_b	Volume I
WO1	ω_{o1}	Input

PROGRAM SYMBOL	ENGINEERING SYMBOL	DESCRIPTION
WO2	ω_{o2}	Input
WO3	ω_{o3}	Input
X(I)	X_i	Position and velocity state variables
XD(I)	$\dot{X}_i$	Derivatives of position and velocity state variables
XDIFF(I)		Difference between actual and observed position and velocity
XIC	ζ_c	Cosine of limiting plane angle, ζ
YA	Y_A	Actual value of observable
YP	Y_P	Predicted value of observable
YOUC		Cosine of $(\omega_o+\theta)$
YOUF		Sine of $(\omega_o+\theta)$
ZB11	Z_{11}	Parameter in libration analysis
ZB13	Z_{13}	Parameter in libration analysis
ZB21	Z_{21}	Parameter in libration analysis
ZB23	Z_{23}	Parameter in libration analysis
ZE1	ξ_1	Volume I
ZE2	ξ_2	Volume I
ZE3	ξ_3	Volume I
ZETAS		Sine of ζ
ZMEAN		Mean for the computed random numbers
ZSQ		ZE2 for symetric vehicle
Z1	Z_1	Input
Z2	Z_2	Input
Z3	Z_3	Input

APPENDIX B: Program Listings

Due to the volume of the listings, this table of contents is provided as an aid in finding the listing of any particular routine.

```
C       MAIN PROGRAM
$EXECUTE        IBJOB
$IBJOB
$IBFTC JF
      COMMON/OMEGA/ HWO1,HWO2,HWO3,RT1,RT2,ZB11,ZB13,ZB21,ZB23
      COMMON/XCIN/X,TAU,CHX,SHX,HG,GAM,STHE,CTHE,THETA,COMC,SOMC,
     1CIC,SIC,G,SPRM,PM,PP,PMDUM,TM,CWS,PBO,QKB,QLBC,QKBP,R,SR,QKBDR,
     2BETA,CSUBB,SRCUB,BETAP,OWZ
      COMMON/GIVE/NT,NZ,OAPZ,VAV,VQ,OEZ,ZE1,ZE2,ZE3,D,VI,
     1ETAZ,T,SSP,B,C,PSUBP,Z1,Z2,Z3,WO1,WO2,WO3
      COMMON/GET/NSUBB,NSUBC,NSUBM,NVAR,EM,PSI11,PSI13,PSI21,PSI23
     1,WSAP,WSBP,ASKH,ASUBK,WSUBP,ASCP,ASSP,CAHV,SAHV
      DOUBLE PRECISION V,VDP,QQ,RR,EM,BCHK,ECCAN,CAPEZ,TERM1
      DOUBLE PRECISION PSI11,PSI13,PSI21,PSI23,DUM,RT1,RT2,
     1ZB11,ZB13,ZB21,ZB23,BTSP,HBTSP,ABETA,HBETAP,
     3ASP,ACP,BCP,BSP,PSUBC,PSUBS,PSI2,CCHK1,CCHK2,CCHK3
      DOUBLE PRECISION WSAP,WSBP,ASKH,ASUBK,WSUBP,ASCP,ASSP,CAHV,SAHV
      DOUBLE PRECISION PI,CIS,CTHDS,CWS,CUBM,QLBC,ACON,PSUBP,AZ,TZ,T,U,
     1ETAZ,CSUBB,PZ,COMC,SOMC,CIC,SIC,ZE1,ZE2,ZE3,RHO,UC,QM,TAU,
    2XIC,ASUBM,THETAL,CUC,SUF,                           HZ1,HZ2,HZ3,
     3HWO1,HWO2,HWO3,                                    TP,TM,ERR1,ERR2,
     4TDIFF,THETA,THDIF,CTHE,STHE,SR,THES,THX2,SRCUB,YOUC,YOUF,PSIB,
     5QKBDR,TEST,TESL,TESR,OUTA,OUTB,OUTC,OUTD,OUTE,OUTF,OUTG,OUTH,
     6OUTI,OUTJ,OUTK,OUTL,C64,SL1N,SL1U,ZDUM,ZETAS,YA,CSUBH,SDPU,
     7SDPL,HZETAS,YP,HCSUBH,SCALAR,PSUM1,PSUM2,APXM1,APXM2
      DOUBLE PRECISION OOZ,OIZ,Z1,Z2,Z3,VLQ,VPQ,OBZD,PBO,TSO
      DOUBLE PRECISION GNU,Q,HGNU,HH,D,C,B,X,XD,HC,HB,HX,HXD,PH,PHD,
     1PP,SSP,QKB,QKBP,SS,XDIFF,F7N,GAM,SSRL,SPRM,G,SX,CX,SHX,CHX,
     2FS,FG,HGAM,HSPRM,HG,HFS,HFG,HCAPF,HLOWF,A,ZEE,RHOX,BDUM,PM,
     3PMDUM,R,CAPN,HIA,HSDP,HCAPN,W,ADUM,WDUM,HDUM,PDIAG,APX
     4,BETA,BETAP,VI
      DIMENSION V(3),VDP(3),QQ(3,3),RR(3,3),EM(3,3),BCHK(3,3)
      DIMENSION GNU(3,3),Q(3),VI(3),HGNU(3,3),HH(6),EL(12,3),
     1D(3,3),C(3,3),B(3,3),X(48),XD(48),HC(3,3),HB(3,3),HX(6),HXD(6),
     2PH(6,6),PHD(6,6),PP(6,6),SSP(3),QKB(3),QKBP(3),SS(3),XDIFF(6),
     3F7N(350),GAM(3,3),SSRL(3),SPRM(3),G(3,3),SX(3),SHX(3),CX(3),
     4CHX(3),FG(3),FS(3),HGAM(3,3),HSPRM(3),HG(3,3),HFG(3),HFS(3),
     5HCAPF(3,3),HLOWF(3,3),A(3,3),ZEE(3),RHOX(6),BDUM(3,3),PM(6,6),
     6PMDUM(6,6),R(3),UA(12,3),CAPN(3),HIA(6),HSDP(3),HCAPN(3),W(6),
     7ADUM(6),WDUM(6,6),HDUM(6,6),PDIAG(6),APX(6),BTSP(3),HBTSP(3)
```

```
     8,BETA(3),BETAP(3),EL1(12,3),HBETAP(3)
      DATA((EL1(I,J),J=1,3),I=1,12)/1.,0.,0.,0.,1.,0.,0.,0.,1.,
     1-1.,0.,0.,0.,-1.,0.,0.,0.,-1.,1.,0.,0.,0.,1.,0.,0.,0.,1.,
     2-1.,0.,0.,0.,-1.,0.,0.,0.,-1./
      DATA((EL(I,J),J=1,3),I=1,12)/0.,0.,-1.,-1.,0.,0.,0.,-1.,0.,0.,
     10.,1.,1.,0.,0.,0.,1.,0.,0.,1.,0.,0.,0.,1.,1.,0.,0.,0.,-1.,
     20.,0.,0.,-1.,-1.,0.,0./
      EQUIVALENCE (X(7),HX(1)),(XD(7),HXD(1)),(X(13),PH(1)),(XD(13),
     1PHD(1)),(ZE1,ZEE(1))
      DATA((UA(I,J),J=1,3),I=1,12)/0.,1.,0.,0.,0.,1.,1.,0.,0.,0.,1.,0.,
     10.,0.,1.,1.,0.,0.,0.,0.,1.,1.,0.,0.,0.,1.,0.,0.,0.,1.,1.,
     20.,0.,0.,1.,0./
C          SETTING CONSTANTS
      PI = 3.141592653589793
      CLS = 23.5*PI/180.
      CTHDS =2.*PI/(365.*24.*3600.)
      CWS = 7.292E-05
      CUBM = 1.E 17
      DUM = 78.9*PI/180.
      QLBC = DCOS(DUM)
      ZMEAN = 0.0
      C64 = DCOS(64.*PI/180.)
C          READING DATA
 999  CONTINUE
      READ(5,1000) DAV,FRA,OWZ,OAPZ,OEZ,OTPZ,CTP,(VI(I),I=1,3),VQ,VLQ,
     1VPQ,VAV,OBZD,OBYD,SIGZ,SIGW,SIGS,SIGB,TSO,Z1,Z2,Z3,WO1,WO2,
     2WO3,PBO
      READ(5,1001) NS,M,NZ,NT,NRS
      OIZ = DAV
      OOZ = FRA
      WRITE(6,1002) OIZ,OOZ,OWZ,OAPZ,OEZ,OTPZ,(VI(I),I=1,3),VQ,VLQ,
     1VPQ,VAV,NS,OBZD,OBYD,M,SIGZ,SIGW,SIGS,SIGB,Z1,Z2,Z3,PBO,
     2WO1,WO2,WO3,TSO,NZ,NT,CTP,NRS
      LOOP = 0
      IF(NZ) 1,1,2
 1    NZ = 2
      PSUBP = .9E-05/(OAPZ**2)
      AZ = OAPZ*9.E 07*1610.
      TZ = OTPZ*3600.
      T = CTP*3600.
```

```
      U = 0.0
      ETAZ = DSQRT(1.325E 20)/DSQRT(AZ**3)
      DO 19 I = 1,3
      SS(I) = 0.0
      QKB(I) = 0.0
19    QKBP(I) = 0.0
      GO TO 3
2     PSUBP = .9E-05
      QKB(3) = DSIN(DUM)
      AZ = OAPZ*1610.
      TZ = OTPZ*3600.
      T = CTP*3600.
      U = 3.986E 14
      ETAZ = DSQRT(U)/DSQRT(AZ**3)
3     CSUBB = -CUBM/(4.*PI)
      PZ = AZ*(1.-OEZ**2)
      COMC = DCOS(OOZ)
      SOMC = DSIN(OOZ)
      CIC = DCOS(OIZ)
      SIC = DSIN(OIZ)
      ZE1 = (VI(3)-VI(2))/VI(1)
      ZE2 = (VI(1)-VI(3))/VI(2)
      ZE3 = (VI(2)-VI(1))/VI(3)
      DO 15 K=1,3
      IF(DABS(ZEE(K))-1.) 15,15,16
16    WRITE(6,1019)(ZEE(I),I=1,3)
      GO TO 400
15    CONTINUE
      RHO = PSUBP*VAV
      Q(1) = VQ*DCOS(VLQ)*DCOS(VPQ)
      Q(2) = VQ*DCOS(VLQ)*DSIN(VPQ)
      Q(3) = VQ*DSIN(VLQ)
      DO 4 I = 1,3
      DO 4 J = 1,3
      IF(I-J) 5,6,5
5     GNU(I,J) = RHO*Q(I)/VI(J)
      GO TO 4
6     GNU(I,J) = 0.0
4     CONTINUE
      IF(NT) 7,7,8
```

```
7     HU = 0.0
      DO 10 I = 1,3
      DO 10 J = 1,3
10    HGNU(I,J) = 0.0
      GO TO 11
8     HU = U
      DO 9 I = 1,3
      DO 9 J = 1,3
9     HGNU(I,J) = GNU(I,J)
11    QM = M
      TAU = T/QM
      XIC = DCOS(OBZD)
      HH(4) = 0.0
      HH(5) = 0.0
      HH(6) = 0.0
      ASUBM = -ETAZ*TZ
      CAPEZ = ECCAN(ASUBM,OEZ)
      TERM1 = (DCOS(CAPEZ)-OEZ)/(1.-OEZ*DCOS(CAPEZ))
      IF(TERM1)5401,5402,5401
5402  THETAL = (CAPEZ*PI)/(2.*DABS(CAPEZ))
      GO TO 5405
5401  IF(DABS(TERM1)-1.)5403,5403,5404
5404  WRITE(6,1041)
1041  FORMAT(27HTROUBLE IN ARCCOS FOR THETA)
      GO TO 999
5403  THETAL = (CAPEZ/DABS(CAPEZ))*DATAN(DSQRT(1.-TERM1**2)/TERM1)
5405  CONTINUE
      CUC = DCOS(OWZ+THETAL)
      SUF = DSIN(OWZ+THETAL)
      D(1,1) = -COMC*SUF-SOMC*CIC*CUC
      D(1,2) = SOMC*SIC
      D(1,3) = COMC*CUC-SOMC*CIC*SUF
      D(2,1) = -SOMC*SUF+COMC*CIC*CUC
      D(2,2) = -COMC*SIC
      D(2,3) = SOMC*CUC+COMC*CIC*SUF
      D(3,1) = SIC*CUC
      D(3,2) = CIC
      D(3,3) = SIC*SUF
      CALL CEE(Z1,Z2,Z3,C)
      DO 25 I = 1,3
```

```
      DO 25 J = 1,3
25    B(I,J) = D(I,1)*C(1,J)+D(I,2)*C(2,J)+D(I,3)*C(3,J)
      DO 26 L = 1,3
26    X(L) = 0.0
      X(4) = WO1
      X(5) = WO2
      X(6) = WO3
      HZ1 = RANNUM(SIGZ,ZMEAN,NRS)+Z1
      HZ2 = RANNUM(SIGZ,ZMEAN,NRS)+Z2
      HZ3 = RANNUM(SIGZ,ZMEAN,NRS)+Z3
      HWO1 = RANNUM(SIGW,ZMEAN,NRS)+WO1
      HWO2 = RANNUM(SIGW,ZMEAN,NRS)+WO2
      HWO3 = RANNUM(SIGW,ZMEAN,NRS)+WO3
      CALL CEE(HZ1,HZ2,HZ3,HC)
      DO 30 I = 1,3
      DO 30 J = 1,3
30    HB(I,J) = D(I,1)*HC(1,J)+D(I,2)*HC(2,J)+D(I,3)*HC(3,J)
      DO 31 L = 1,3
31    HX(L) = 0.0
      HX(4) = HWO1
      HX(5) = HWO2
      HX(6) = HWO3
      DO 50 IJ = 1,6
50    XDIFF(IJ) = HX(IJ)-X(IJ)
      DO 35 I = 1,6
      DO 35 J = 1,6
35    PP(I,J) = 0.0
      SIGZ2 = SIGZ**2
      DO 36 I = 1,3
36    PP(I,I) = SIGZ2
      SIGW2 = SIGW**2
      DO 37 I = 4,6
37    PP(I,I) = SIGW2
      GO TO (41,42),NZ
42    SSP(1) = -C(3,1)
      SSP(2) = -C(3,2)
      SSP(3) = -C(3,3)
      GO TO 45
41    QKB(1) = QLBC*DCOS(PBO)
      QKB(2) = QLBC*DSIN(PBO)
```

```
      DO 46 I = 1,3
   46 QKBP(I) = B(1,I)*QKB(1)+B(2,I)*QKB(2)+B(3,I)*QKB(3)
      SS(1) = DCOS(TSO)
      SS(2) = DCOS(CIS)*DSIN(TSO)
      SS(3) = DSIN(CIS)*DSIN(TSO)
      DO 47 I = 1,3
   47 SSP(I) = B(1,I)*SS(1)+B(2,I)*SS(2)+B(3,I)*SS(3)
   45 WRITE(6,1003) ((D(I,J),I=1,3),J=1,3),SS,QKB,((C(K,L),L=1,3),
     1K=1,3),SSP,QKBP,(X(II),II=1,6),(XDIFF(IJ),IJ=1,6)
      CALL CHECK1
C          BEGINNING OF RECURSION LOOP
C
C          SETTING UP ICE CALLING SEQUENCE
      TM = 0.0
      TP = 0.0
      ERR1 = 1.E-10
      ERR2 = 1.E-08
      IORD = 4
C          INITIALIZE PHI ARRAY
      DO 54 I = 1,6
      DO 54 J = 1,6
      IF(I-J) 52,53,52
   52 PH(I,J) = 0.0
      GO TO 54
   53 PH(I,J) = 1.0
   54 CONTINUE
   55 LQQQ=4
   60 CALL DICE(TAU,TM,TP,ERR1,ERR2,NVAR,X,XD,F7N,LQQQ,INDEX,IORD)
      GO TO (100,200,300,400),LQQQ
C          BOX A
  100 TDIFF = ETAZ*(TM-TZ)
      CAPEZ = ECCAN(TDIFF,OEZ)
      TERM1 = (DCOS(CAPEZ)-OEZ)/(1.-OEZ*DCOS(CAPEZ))
      IF(TERM1) 5406,5407,5406
 5407 THETA = (CAPEZ*PI)/(2.*DABS(CAPEZ))
      GO TO 5410
 5406 IF(DABS(TERM1)-1.)5408,5408,5409
 5409 WRITE(6,1042)
 1042 FORMAT(29HBAD ARCCOS FOR THETA IN BOX A)
      GO TO 999
```

```
5408 THETA = (CAPEZ/DABS(CAPEZ))*DATAN(DSQRT(1.-TERM1**2)/TERM1)
5410 CONTINUE
     THDIF = THETA-THETAL
     CTHE = DCOS(THDIF)
     STHE = DSIN(THDIF)
     DO 101 I = 1,3
101  GAM(I,3) = C(3,I)*CTHE+C(1,I)*STHE
     GO TO (102,103),NZ
102  SR = PZ/(1.+OEZ*DCOS(THETA))
     THES = TSO+CTHDS*TM
     SSRL(1) = DCOS(THES)
     SSRL(2) = DCOS(CIS)*DSIN(THES)
     SSRL(3) = DSIN(CIS)*DSIN(THES)
     DO 104 I = 1,3
104  SPRM(I) = B(1,I)*SSRL(1)+B(2,I)*SSRL(2)+B(3,I)*SSRL(3)
     GO TO 107
103  SR = 1.0
     DO 106 I = 1,3
106  SPRM(I) = -GAM(I,3)
107  DO 110 I = 1,3
     SX(I) = DSIN(X(I))
110  CX(I) = DCOS(X(I))
     CALL CEE (X(1),X(2),X(3),G)
     DO 111 I = 1,3
     FG(I) = G(1,I)*GAM(1,3)+G(2,I)*GAM(2,3)+G(3,I)*GAM(3,3)
111  FS(I) = G(1,I)*SPRM(1)+G(2,I)*SPRM(2)+G(3,I)*SPRM(3)
     SRCUB = SR**3
     XD(1) = -X(4)+(SX(2)/CX(2))*(X(6)*CX(1)-X(5)*SX(1))
     XD(2) = -X(5)*CX(1)-X(6)*SX(1)
     XD(3) =-(1./CX(2))*(X(6)*CX(1)-X(5)*SX(1))
     XD(4) = ZE1*(-X(5)*X(6)+3.*U*FG(3)*FG(2)/SRCUB)+GNU(2,1)*FS(3)-
    1GNU(3,1)*FS(2)
     XD(5) = ZE2*(-X(6)*X(4)+3.*U*FG(1)*FG(3)/SRCUB)+GNU(3,2)*FS(1)-
    1GNU(1,2)*FS(3)
     XD(6) = ZE3*(-X(4)*X(5)+3.*U*FG(2)*FG(1)/SRCUB)+GNU(1,3)*FS(2)-
    1GNU(2,3)*FS(1)
C         OBSERVED QUANTITIES
     DO 115 I = 1,3
115  HGAM(I,3) = HC(3,I)*CTHE+HC(1,I)*STHE
     GO TO (116,118),NZ
```

```
116   DO 117 J = 1,3
117   HSPRM(J) = HB(1,J)*SSRL(1)+HB(2,J)*SSRL(2)+HB(3,J)*SSRL(3)
      GO TO 120
118   DO 122 I = 1,3
122   HSPRM(I) = -HGAM(I,3)
120   DO 125 I = 1,3
      SHX(I) = DSIN(HX(I))
125   CHX(I) = DCOS(HX(I))
      CALL CEE (HX(1),HX(2),HX(3),HG)
      DO 128 I = 1,3
      HFG(I) = HG(1,I)*HGAM(1,3)+HG(2,I)*HGAM(2,3)+HG(3,I)*HGAM(3,3)
128   HFS(I) = HG(1,I)*HSPRM(1)+HG(2,I)*HSPRM(2)+HG(3,I)*HSPRM(3)
      HXD(1) = -HX(4)+(SHX(2)/CHX(2))*(HX(6)*CHX(1)-HX(5)*SHX(1))
      HXD(2) = -HX(5)*CHX(1)-HX(6)*SHX(1)
      HXD(3) =-(1./CHX(2))*( HX(6)*CHX(1)-HX(5)*SHX(1))
      HXD(4) = ZE1*(-HX(5)*HX(6)+3.*HU*HFG(3)*HFG(2)/SRCUB)+HGNU(2,1)*
     1HFS(3)-HGNU(3,1)*HFS(2)
      HXD(5) = ZE2*(-HX(6)*HX(4)+3.*HU*HFG(1)*HFG(3)/SRCUB)+HGNU(3,2)*
     1HFS(1)-HGNU(1,2)*HFS(3)
      HXD(6) = ZE3*(-HX(4)*HX(5)+3.*HU*HFG(2)*HFG(1)/SRCUB)+HGNU(1,3)*
     1HFS(2)-HGNU(2,3)*HFS(1)
C          PHI EQUATIONS - SET UP
      IF(NT) 130,130,131
131   HCAPF(1,1) = 0.0
      HCAPF(1,2) = HFS(3)*CHX(1)-HFS(2)*SHX(1)
      HCAPF(1,3) = HSPRM(1)*HG(2,1)-HSPRM(2)*HG(1,1)
      HCAPF(2,1) = -HFS(3)
      HCAPF(2,2) = HFS(1)*SHX(1)
      HCAPF(2,3) = HSPRM(1)*HG(2,2)-HSPRM(2)*HG(1,2)
      HCAPF(3,1) = HFS(2)
      HCAPF(3,2) =-HFS(1)*CHX(1)
      HCAPF(3,3) = HSPRM(1)*HG(2,3)-HSPRM(2)*HG(1,3)
      HLOWF(1,1) = 0.0
      HLOWF(1,2) = HFG(3)*CHX(1)-HFG(2)*SHX(1)
      HLOWF(1,3) = HGAM(1,3)*HG(2,1)-HGAM(2,3)*HG(1,1)
      HLOWF(2,1) = -HFG(3)
      HLOWF(2,2) = HFG(1)*SHX(1)
      HLOWF(2,3) = HGAM(1,3)*HG(2,2)-HGAM(2,3)*HG(1,2)
      HLOWF(3,1) = HFG(2)
      HLOWF(3,2) =-HFG(1)*CHX(1)
```

```
      HLOWF(3,3) = HGAM(1,3)*HG(2,3)-HGAM(2,3)*HG(1,3)
      ACON = 3.*HU/SRCUB
      DO 135 K = 1,3
      K1 = MOD(K,3)+1
      K2 = MOD(K+1,3)+1
      DO 135 L = 1,3
  135 A(K,L) = ZEE(K)*ACON*(HFG(K2)*HLOWF(K1,L)+HFG(K1)*HLOWF(K2,L))+
     1HGNU(K1,K)*HCAPF(K2,L)-HGNU(K2,K)*HCAPF(K1,L)
      GO TO 140
  130 DO 138 I = 1,3
      DO 138 J = 1,3
  138 A(I,J) = 0.0
C          PHI EQUATIONS - DERIVATIVES
  140 THX2 = SHX(2)/CHX(2)
      GO TO (60,145),NSUBM
  145 CONTINUE
      DO 150 I = 1,6
      PHD(1,I) = HXD(2)*THX2*PH(1,I)-(HXD(3)/CHX(2))*PH(2,I)-PH(4,I)-
     1THX2*SHX(1)*PH(5,I)+THX2*CHX(1)*PH(6,I)
      PHD(2,I) = HXD(3)*CHX(2)*PH(1,I)-CHX(1)*PH(5,I)-SHX(1)*PH(6,I)
      PHD(3,I) = (-HXD(2)/CHX(2))*PH(1,I)+HXD(3)*THX2*PH(2,I)+
     1(SHX(1)/CHX(2))*PH(5,I)-(CHX(1)/CHX(2))*PH(6,I)
      PHD(4,I) = A(1,1)*PH(1,I)+A(1,2)*PH(2,I)+A(1,3)*PH(3,I)-ZE1*
     1HX(6)*PH(5,I)-ZE1*HX(5)*PH(6,I)
      PHD(5,I) = A(2,1)*PH(1,I)+A(2,2)*PH(2,I)+A(2,3)*PH(3,I)-ZE2*
     1HX(6)*PH(4,I)-ZE2*HX(4)*PH(6,I)
  150 PHD(6,I) = A(3,1)*PH(1,I)+A(3,2)*PH(2,I)+A(3,3)*PH(3,I)-ZE3*
     1HX(5)*PH(4,I)-ZE3*HX(4)*PH(5,I)
      GO TO 60
C          BOX B
  200 CONTINUE
      GO TO 60
C          BOX C
  300 IF(DABS(TM)-.0001) 4093,4093,3332
 3332 DO 301 I = 1,6
  301 RHOX(I) = HX(I)
      CALL HELP
      DO 672 I = 1,3
      BTSP(I) = G(I,1)*BETAP(1)+G(I,2)*BETAP(2)+G(I,3)*BETAP(3)
  672 HBTSP(I) = HB(1,I)*BETA(1)+HB(2,I)*BETA(2)+HB(3,I)*BETA(3)
```

```
      ABETA = DSQRT(BETA(1)**2+BETA(2)**2+BETA(3)**2)
      GO TO (321,322),NZ
  321 TEST = 1.-(6.4E 06/SR)**2
      IF(TEST) 329,330,330
  329 WRITE(6,1005) TEST,SR
      GO TO 999
  330 TESR = -DSQRT(TEST)
      TESL = D(1,3)*SSRL(1)+D(2,3)*SSRL(2)+D(3,3)*SSRL(3)
      IF(TESL-TESR) 331,331,332
  331 DO 333 I = 1,3
      DO 333 J = 1,3
      GNU(I,J) = 0.0
  333 HGNU(I,J) = 0.0
      GO TO 322
  332 DO 334 I = 1,3
      DO 334 J = 1,3
      IF(I-J) 335,336,335
  335 GNU(I,J) = RHO*Q(I)/VI(J)
      GO TO 334
  336 GNU(I,J) = 0.0
  334 CONTINUE
      IF(NT) 337,337,338
  337 DO 339 I = 1,3
      DO 339 J = 1,3
  339 HGNU(I,J) = 0.0
  338 DO 340 K = 1,3
      DO 340 L = 1,3
  340 HGNU(K,L) = GNU(K,L)
C           PERIODIC READOUTS
  322 LOOP = LOOP+1
      OUTA = DSQRT(X(1)**2+X(2)**2+X(3)**2)
      OUTB = DSQRT(X(4)**2+X(5)**2+X(6)**2)
      OUTC = DSQRT(PM(1,1))
      OUTD = DSQRT(PM(2,2))
      OUTE = DSQRT(PM(3,3))
      OUTF = DSQRT(PM(4,4))
      OUTG = DSQRT(PM(5,5))
      OUTH = DSQRT(PM(6,6))
      OUTI = DSQRT(PM(1,1)+PM(2,2)+PM(3,3))
      OUTJ = DSQRT(PM(4,4)+PM(5,5)+PM(6,6))
```

```
      DO 341 L = 1,6
  341 XDIFF(L) = RHOX(L)-X(L)
      OUTK = DSQRT(XDIFF(1)**2+XDIFF(2)**2+XDIFF(3)**2)
      OUTL = DSQRT(XDIFF(4)**2+XDIFF(5)**2+XDIFF(6)**2)
      GO TO (801,802),NSUBB
  801 ACP=DCOS(WSAP*TM+ASKH)
      ASP=DSIN(WSAP*TM+ASKH)
      BCP=DCOS(WSBP*TM-ASUBK)
      BSP=DSIN(WSBP*TM-ASUBK)
      PSUBC=DCOS(WSUBP*TM)
      PSUBS =  DSIN(WSUBP*TM)
      QQ(1,1)=ASCP
      QQ(1,2)=ASSP*ASP
      QQ(1,3)=ASSP*ACP
      QQ(2,1)=0.0
      QQ(2,2)=ACP
      QQ(2,3)=-ASP
      QQ(3,1)=-ASSP
      QQ(3,2)=ASCP*ASP
      QQ(3,3)=ASCP*ACP
      RR(1,1)=QQ(1,1)
      RR(1,2)=QQ(1,2)
      RR(1,3)=QQ(1,3)
      RR(2,1)=BCP*QQ(2,1)+BSP*QQ(3,1)
      RR(2,2)=BCP*QQ(2,2)+BSP*QQ(3,2)
      RR(2,3)=BCP*QQ(2,3)+BSP*QQ(3,3)
      RR(3,1)=-BSP*QQ(2,1)+BCP*QQ(3,1)
      RR(3,2)=-BSP*QQ(2,2)+BCP*QQ(3,2)
      RR(3,3)=-BSP*QQ(2,3)+BCP*QQ(3,3)
      QQ(1,1)=CAHV*RR(1,1)+SAHV*RR(3,1)
      QQ(1,2)=CAHV*RR(1,2)+SAHV*RR(3,2)
      QQ(1,3)=CAHV*RR(1,3)+SAHV*RR(3,3)
      QQ(2,1)=RR(2,1)
      QQ(2,2)=RR(2,2)
      QQ(2,3)=RR(2,3)
      QQ(3,1)=-SAHV*RR(1,1)+CAHV*RR(3,1)
      QQ(3,2)=-SAHV*RR(1,2)+CAHV*RR(3,2)
      QQ(3,3)=-SAHV*RR(1,3)+CAHV*RR(3,3)
      RR(1,1)=QQ(1,1)
      RR(1,2)=QQ(1,2)
```

```
      RR(1,3)=QQ(1,3)
      RR(2,1)=PSUBC*QQ(2,1)-PSUBS*QQ(3,1)
      RR(2,2)=PSUBC*QQ(2,2)-PSUBS*QQ(3,2)
      RR(2,3) = PSUBC*QQ(2,3)-PSUBS*QQ(3,3)
      RR(3,1)=PSUBS*QQ(2,1)+PSUBC*QQ(3,1)
      RR(3,2)=PSUBS*QQ(2,2)+PSUBC*QQ(3,2)
      RR(3,3)=PSUBS*QQ(2,3)+PSUBC*QQ(3,3)
      DO 803 I=1,3
      DO 803 J=1,3
803   BCHK(I,J)=EM(I,1)*RR(1,J)+EM(I,2)*RR(2,J)+EM(I,3)*RR(3,J)
      GO TO 805
802   DO 804 I=1,3
      DO 804 J=1,3
804   BCHK(I,J)=0.0
805   GO TO (807,808),NSUBC
807   PSI2=DATAN2(WO2-ETAZ,ETAZ*Z2*DSQRT(3.*ZE2))
      CCHK1=ZB13*DCOS(RT1*TM+PSI13)-ZB23*DCOS(RT2*TM+PSI23)
      CCHK2=ZB11*DCOS(RT1*TM+PSI11)-ZB21*DCOS(RT2*TM+PSI21)
      CCHK3=DSQRT(Z2**2+(ETAZ-WO2)**2/(3.*ZE2*ETAZ**2))*DCOS(PSI2+DSQRT
     1(3.*ZE2)*ETAZ*TM)
      GO TO 810
808   CCHK1=0.0
      CCHK2=0.0
      CCHK3=0.0
810   WRITE(6,1006)TM,LOOP,G(3,2),G(1,3),G(2,1),(D(I,3),I=1,3),(C(1,I),
     1I=1,3),CCHK1,(C(2,J),J=1,3),CCHK2,(C(3,K),K=1,3),CCHK3,(SSP(L),
     2L=1,3),(QKBP(I),I=1,3),((PH(J,K),K=1,6),J=1,6)
      WRITE (6,1023)  ((B(I,J),J=1,3),(BCHK(I,K),K=1,3),I=1,3),
     1(X(I),I=1,6),OUTA,OUTB,OUTC,OUTD,OUTE,OUTF,OUTG,OUTH,OUTI,OUTJ,
     2(XDIFF(J),J=1,6),OUTK,OUTL
      GO TO (345,346),NZ
345   MAG = 1
      KTR = 1
5412  IF(DABS(BETAP(KTR)/ABETA)-.200)5413,5425,5425
5413  CONTINUE
5417  RANDOM = RANNUM(SIGB,ZMEAN,NRS)
      DO 5418 I=1,3
5418  HBETAP(I) = HG(1,I)*HBTSP(1)+HG(2,I)*HBTSP(2)+HG(3,I)*HBTSP(3)
      YA = BETAP(KTR)
      YP = HBETAP(KTR)
```

```
      GO TO (5421,5422,5423),KTR
5421  HH(1) = 0.
      HH(2) = HBETAP(3)*CHX(1)-HBETAP(2)*SHX(1)
      HH(3) = HBTSP(1)*HG(2,1)-HBTSP(2)*HG(1,1)
      GO TO 5424
5422  HH(1) = -HBETAP(3)
      HH(2) = HBETAP(1)*SHX(1)
      HH(3) = HBTSP(1)*HG(2,2)-HBTSP(2)*HG(1,2)
      GO TO 5424
5423  HH(1) = HBETAP(2)
      HH(2) = -HBETAP(1)*CHX(1)
      HH(3) = HBTSP(1)*HG(2,3)-HBTSP(2)*HG(1,3)
5424  DO 5426 I=1,3
5426  ADUM(I) = HH(1)*PM(1,I)+HH(2)*PM(2,I)+HH(3)*PM(3,I)
      SCALAR = 1./(ADUM(1)*HH(1)+ADUM(2)*HH(2)+ADUM(3)*HH(3)+SIGB**2)
      DO 5427 J=1,6
5427  W(J) = (PM(J,1)*HH(1)+PM(J,2)*HH(2)+PM(J,3)*HH(3))*SCALAR
      GO TO 3371
5429  IF(TESL-TESR) 375,375,346
346   N = 1
      MAG = 2
347   SL1N = SSP(1)*EL1(N,1)+SSP(2)*EL1(N,2)+SSP(3)*EL1(N,3)
      IF(SL1N-C64) 348,349,349
348   N = N+1
      IF(N-NS) 347,347,375
349   SL1U = SSP(1)*UA(N,1)+SSP(2)*UA(N,2)+SSP(3)*UA(N,3)
      IF(DABS(SL1U)-XIC) 350,350,348
350   ZDUM = 1.-SL1U**2
      IF(ZDUM) 351,352,352
351   WRITE(6,1008) N,LOOP,SL1U
      GO TO 999
352   ZETAS = DSQRT(ZDUM)
      SLN  = SSP(1)*EL (N,1)+SSP(2)*EL (N,2)+SSP(3)*EL (N,3)
      ZDUM = SLN/ZETAS
      IF(ABS(ZDUM)-1.) 353,353,354
354   WRITE(6,1009) N,LOOP,SLN ,ZETAS
      GO TO 999
353   YA =DATAN2(DSQRT(1.-ZDUM**2),ZDUM)
      IF(ABS(YA-PI/2.)-OBYD) 356,356,348
356   DO 357 I = 1,3
```

```
357  CAPN(I) = ZETAS*EL(N,I)+DCOS(YA)*SL1U*UA(N,I)
     CSUBH = -1./(ZETAS**2*DSIN(YA))
     RANDOM = RANNUM(SIGS,ZMEAN,NRS)
     DO 360 J = 1,3
360  HSDP(J) = HG(1,J)*HSPRM(1)+HG(2,J)*HSPRM(2)+HG(3,J)*HSPRM(3)
     SDPU = HSDP(1)*UA(N,1)+HSDP(2)*UA(N,2)+HSDP(3)*UA(N,3)
     ZDUM = 1.-SDPU**2
     IF(ZDUM) 361,362,362
361  WRITE(6,1010) N,LOOP,SDPU
     GO TO 999
362  HZETAS = DSQRT(ZDUM)
     SDPL = HSDP(1)*EL(N,1)+HSDP(2)*EL(N,2)+HSDP(3)*EL(N,3)
     ZDUM = SDPL/HZETAS
     IF(ABS(ZDUM)-1.) 363,363,364
364  WRITE(6,1011) N,LOOP,SDPL,HZETAS
     GO TO 999
363  YP =DATAN2(DSQRT(1.-ZDUM**2),ZDUM)
     DO 365 I = 1,3
365  HCAPN(I) = HZETAS*EL(N,I)+DCOS(YP)*SDPU*UA(N,I)
     HCSUBH = -1./(HZETAS**2*DSIN(YP))
     HH(1) = HCSUBH*(HCAPN(3)*HSDP(2)-HCAPN(2)*HSDP(3))
     HH(2) =HCSUBH*((CHX(1)*HSDP(3)-SHX(1)*HSDP(2))*HCAPN(1)+
    1HSDP(1)*(SHX(1)*HCAPN(2)-CHX(1)*HCAPN(3)))
     HH(3) = HCSUBH*((HSPRM(1)*HG(2,1)-HSPRM(2)*HG(1,1))*HCAPN(1)+
    1(HSPRM(1)*HG(2,2)-HSPRM(2)*HG(1,2))*HCAPN(2)+(HSPRM(1)*
    2HG(2,3)-HSPRM(2)*HG(1,3))*HCAPN(3))
     DO 367 I = 1,3
367  ADUM(I) = HH(1)*PM(1,I)+HH(2)*PM(2,I)+HH(3)*PM(3,I)
     SCALAR = 1./(ADUM(1)*HH(1)+ADUM(2)*HH(2)+ADUM(3)*HH(3)+SIGS**2)
     DO 369 J = 1,6
369  W(J) = (PM(J,1)*HH(1)+PM(J,2)*HH(2)+PM(J,3)*HH(3))*SCALAR
3371 DO 371 I = 1,6
     DO 371 J = 1,6
371  WDUM(I,J) = W(I)*HH(J)
     DO 373 K = 1,6
     DO 373 L = 1,6
     IF(K-L) 374,377,374
377  HDUM(K,L) = 1.-WDUM(K,L)
     GO TO 373
374  HDUM(K,L) = -WDUM(K,L)
```

```
373   CONTINUE
      DO 379 I = 1,6
      DO 379 J = 1,6
379   PP(I,J) = HDUM(I,1)*PM(1,J)+HDUM(I,2)*PM(2,J)+HDUM(I,3)*
     1PM(3,J)+HDUM(I,4)*PM(4,J)+HDUM(I,5)*PM(5,J)+HDUM(I,6)*PM(6,J)
      WRITE (6,1073) ((PM(I,J),J=1,6),I=1,6)
1073  FORMAT(1H0,6F18.8)
      DO 381 K = 1,6
381   PDIAG(K) = DSQRT(PP(K,K))
      PSUM1 = DSQRT(PP(1,1)+PP(2,2)+PP(3,3))
      PSUM2 = DSQRT(PP(4,4)+PP(5,5)+PP(6,6))
      DO 383 J = 1,6
      HX(J) = RHOX(J)+W(J)*(YA-YP+RANDOM)
383   APX(J) = HX(J)-X(J)
      APXM1 = DSQRT(APX(1)**2+APX(2)**2+APX(3)**2)
      APXM2 = DSQRT(APX(4)**2+APX(5)**2+APX(6)**2)
      GO TO(3383,3384),MAG
3383  N = KTR
3384  CONTINUE
      WRITE (6,1015) N,YA,(HIA(K),K=1,3),RANDOM,YP,(HH(I),I=1,3),
     1(PDIAG(K),K=1,6),PSUM1,PSUM2,(APX(J),J=1,6),APXM1,APXM2,
     2(W(L),L=1,6)
      DO 385 K = 1,6
385   RHOX(K) = HX(K)
      DO 386 I = 1,3
      SHX(I) = DSIN(HX(I))
386   CHX(I) = DCOS(HX(I))
      CALL CEE (HX(1),HX(2),HX(3),HG)
      DO 389 I = 1,6
      DO 389 J = 1,6
389   PM(I,J) = PP(I,J)
      GO TO (5425,348),MAG
5425  IF(KTR-3) 5428,5429,5429
5428  KTR = KTR+1
      GO TO 5412
375   THETAL = THETA
      DO 387 I = 1,3
      DO 387 J = 1,3
387   C(I,J) = GAM(1,I)*G(1,J)+GAM(2,I)*G(2,J)+GAM(3,I)*G(3,J)
      X(1) = 0.0
```

```
      X(2) = 0.0
      X(3) = 0.0
      DO 392 I = 1,3
      DO 392 J = 1,3
392   HC(I,J) = HGAM(1,I)*HG(1,J)+HGAM(2,I)*HG(2,J)+HGAM(3,I)*HG(3,J)
      DO 393 K = 1,3
      DO 393 L = 1,3
393   BDUM(K,L) = HB(K,1)*HG(1,L)+HB(K,2)*HG(2,L)+HB(K,3)*HG(3,L)
      DO 394 I = 1,3
      DO 394 J = 1,3
394   HB(I,J) = BDUM(I,J)
      HX(1) = 0.0
      HX(2) = 0.0
      HX(3) = 0.0
      DO 409 I = 1,6
      DO 409 J = 1,6
      IF(I-J) 410,411,410
410   PH(I,J) = 0.0
      GO TO 409
411   PH(I,J) = 1.0
409   CONTINUE
      CALL EXTRA (B,HB,G,HG,SPRM,HSPRM,HSDP)
4093  CONTINUE
      TP = TP+TAU
      IF(TP-T) 55,55,999
C             BOX D
400   CALL PDUMP
      GO TO 999
1000  FORMAT(7F10.0)
1001  FORMAT(7I10)
1002  FORMAT(1H1,14H*** INPUTS ***/
     16H0ORBIT,6H  OIZ=F16.8,6H  OOZ=F16.8,6H  OWZ=F16.8,6H OATP=F16.8/
     21H ,5X  ,6H  OEZ=F16.8,6H OTPZ=F16.8/
     38H0VEHICLE,4H I1=F16.8,6H   I2=F16.8,6H   I3=F16.8,6H   VQ=F16.8/
     41H ,5X  ,6H  VLQ=F16.8,6H  VPQ=F16.8,6H  VAV=F16.8/
     512H0OBSERVATION,4H NS=I4,6H OBZD=F16.8,6H OBYD=F16.8,4H  M=I7/
     66H0STAT.,6H SIGZ=F16.8,6H SIGW=F16.8,6H SIGS=F16.8,6H SIGB=F16.8/
     76H0INIT.,6H   Z1=F16.8,6H   Z2=F16.8,6H   Z3=F16.8,6H  PBO=F16.8/
     81H ,5X  ,6H  WO1=F16.8,6H  WO2=F16.8,6H  WO3=F16.8,6H  TSO=F16.8/
     98H0CONTROL,4H NZ=I5,4H NT=I5,6H  CTP=F16.8,6H  NRS=I5)
```

```
1003 FORMAT(1H0,6HCROSI ,3F16.8/1H0,6H POLI ,3F16.8/1H0,5H VERI,3F16.8/
    1        1H0,6H SUNI ,3F16.8/1H0,6H MAGI ,3F16.8/1H0,5HCROSV,3F16.8/
    2        1H0,6H POLV ,3F16.8/1H0,6H VERV ,3F16.8/1H0,5H SUNV,3F16.8/
    3        1H0,6H MAGV ,3F16.8/1H0,6H  XA  ,6F16.8/1H0,5H XO  ,6F16.8)
1005 FORMAT(1H0,5HTEST=F16.8,8HWHEN SR=F16.8)
1008 FORMAT(1H0,2HN=I3,7H  LOOP=,I5,7H  SL1U=F16.8)
1009 FORMAT(1H0,2HN=I3,7H  LOOP=,I5,7H  SL1N=F16.8,7H ZETAS=F16.8)
1011 FORMAT(1H0,2HN=I3,7H  LOOP=,I5,7H  SDPL=F16.8,7H   HZS=F16.8)
1010 FORMAT(1H0,2HN=I3,7H  LOOP=,I5,7H  SDPU=F16.8)
1006 FORMAT(1H1,5HTIME=F16.8,2X,2HM=I5,2X,3HGX=F16.8,2X,3HGY=F16.8,
    12X,3HGZ=F16.8/6H0 VERI,2X,3(F16.8,2X)/6H0CROSV,2X,3(F16.8,2X),
    27X,3HC11,6X,F16.8,7X,3HC13/6H0 POLV,2X,3(F16.8,2X),7X,3HC21,13X,
    33HC22,F16.8/6H0 VERV,2X,3(F16.8,2X),F16.8,7X,3HC32,13X,3HC33/
    46H0 SUNV,2X,3(F16.8,2X)/6H0 MAGV,2X,3(F16.8,2X)/1H0,30X,17HTRANSIT
    5ION MATRIX/1H0,6(6F16.8,/1H ))
1019 FORMAT(40H0INPUT MOMENTS OF INERTIA NOT REALIZABLE/1H0,3F18.8)
1023 FORMAT(1H0,23X,8HB-MATRIX,43X,14HB-MATRIX CHECK/1H0,3(6F18.8/1H )/
    15H0 XA ,2X,6F16.8/1H0,10X,F16.8,30X,F16.8/5H0 XR ,2X,6F18.8/1H0,
    210X,F16.8,30X,F16.8/6H0  XP ,2X,6F18.8/1H0,10X,F16.8,30X,F16.8)
1015 FORMAT (1H0,10HINST. NO.=I3,9X,3HYA=F16.8,2X,4HH1A=F16.8,2X,4HH2A=
    1,F16.8,2X,4HH3A=F16.8/1H0,3HNR=F14.8,2X,3HYP=F16.8,2X,4HH1 =F16.8,
    22X,4HH2 =F16.8,2X,4HH3 =F16.8/1H0,2HXR,2X,6(F16.8,2X)/1H0,10X,
    3F16.8,30X,F16.8/1H0,2HXO,2X,6(F16.8,2X)/1H0,10X,F16.8,30X,F16.8/
    41H0,1HW,3X,6(F16.8,2X))
     STOP
     END
```

```
$IBFTC DICE     DECK
CDICE           DOUBLE PRECISION ICE                                    DICE0000
      SUBROUTINE DICE(P,TT,TP,E1,E2,NN,Y,DY,F,LL,INDEX,I)               DICE0010
      DIMENSION Y(50),DY(50),F(350)                                     DICE0020
C                                                                       DICE0040
      DOUBLE PRECISION  Y      , DY     , F      , T      , TT          DICE0050
      DOUBLE PRECISION  TS     , H      , DT     , E      , E1          DICE0060
      DOUBLE PRECISION  E2     , H1     , P      , T2     , TP          DICE0070
      DOUBLE PRECISION  H2                                              DICE0080
      L = LL                                                            DICE0100
      T = TT                                                            DICE0110
      GO TO (100,200,300,400),L                                         DICE0120
  100 IG=IG                                                             DICE0130
      GO TO (101,102),IG                                                DICE0140
  101 J = 1                                                             DICE0150
      L = 2                                                             DICE0160
      M = 0                                                             DICE0170
      TS = T                                                            DICE0180
      DO 106 K = 1,N                                                    DICE0190
      K1 = K+3*N                                                        DICE0200
      K2 = K1+N                                                         DICE0210
      K3 = N + K                                                        DICE0220
      F(K1) = Y(K)                                                      DICE0230
      F(K3) = F(K1)                                                     DICE0240
  106 F(K2) = DY(K)                                                     DICE0250
      GO TO 402                                                         DICE0260
  102 CALL DINTEG(T,DT,N,Y,DY,F,J,I)
      J = J+1                                                           DICE0290
      IF(J-I   ) 103,103,104                                            DICE0300
  103 L = 1                                                             DICE0310
      GO TO 402                                                         DICE0320
  104 M = M+1                                                           DICE0330
  105 GO TO (110,120,130),M                                             DICE0340
  110 DO 111 K = 1,N                                                    DICE0350
      K1 = K+N+N                                                        DICE0360
  111 F(K1) = Y(K)                                                      DICE0370
  112 DO 113 K = 1,N                                                    DICE0380
      K1 = K+3*N                                                        DICE0390
      K2 = K1+N                                                         DICE0400
      K3 = N + K                                                        DICE0410
```

```
      Y(K) = F(K1)                                          DICE0420
      F(K3) = F(K1)                                         DICE0430
  113 DY(K) = F(K2)                                         DICE0440
      T = TS                                                DICE0450
      IF(PART(T,T,1)) 114,116,114                           DICE0460
  114 IF(DABS(H/T)-.000001D0) 115,115,116                   DICE0470
  115 M = 0                                                 DICE0480
      L = 4                                                 DICE0490
      GO TO 402                                             DICE0500
  116 DT = .5D0*H                                           DICE0510
      M = 1                                                 DICE0520
      J = 1                                                 DICE0530
      GO TO 300                                             DICE0540
  120 DO 121 K = 1,N                                        DICE0550
      K1 = K+N                                              DICE0560
  121 F(K1) = Y(K)                                          DICE0570
      M = 2                                                 DICE0580
      J = 1                                                 DICE0590
      IG = 2                                                DICE0600
      L = 1                                                 DICE0610
      GO TO 402                                             DICE0620
  130 DO 131 K = 1,N                                        DICE0630
      K1 = K+2*N                                            DICE0640
      F(K) = (Y(K)-F(K1))/(2.D0**I-1.D0)                    DICE0650
      Y(K) = Y(K) +F(K)                                     DICE0660
      IF(DABS(Y(K))-.00001D0)139,139,140                    DICE0670
  139 F(K) = 0.D0                                           DICE0680
      GO TO 131                                             DICE0690
  140 F(K) = DABS(F(K)/Y(K))                                DICE0700
  131 CONTINUE                                              DICE0710
      E = F                                                 DICE0720
      INDEX = 1                                             DICE0730
      IF (N-1)1335,1335,1315                                DICE0740
 1315 DO 133 K = 2,N                                        DICE0750
      IF(E-F(K)) 132,133,133                                DICE0760
  132 INDEX = K                                             DICE0770
      E = F(K)                                              DICE0780
  133 CONTINUE                                              DICE0790
 1335 IF(E -E1) 134,135,135                                 DICE0800
  134 H = H+H                                               DICE0810
```

```
1345 DT = H                                                              DICE0820
     GO TO 401                                                           DICE0830
135  IF(E-E2)1345,1345,136                                               DICE0840
136  DO 137 K = 1,N                                                      DICE0850
     K1 = K+N                                                            DICE0860
     K2 = K+N+N                                                          DICE0870
137  F(K2) = F(K1)                                                       DICE0880
138  H = .5D0*H                                                          DICE0890
     GO TO 112                                                           DICE0900
200  MU=MU                                                               DICE0910
     GO TO (203,204),MU                                                  DICE0920
203  H=0.D0                                                              DICE0930
     CALL STORE(H,H,1,AMAX1(AMAX1(PART(H,H,1),PART(H1,H1,1)),PART(H2,H2DICE0940
    1,1)))                                                               DICE0950
     MU = 2                                                              DICE0960
204  H1 = DABS(H)                                                        DICE0970
     IF(PART(P,P,1)) 205,206,206                                         DICE0980
205  H = -H1                                                             DICE0990
     GO TO 207                                                           DICE1000
206  H = H1                                                              DICE1010
207  IF(DABS(P)-H1) 208,209,209                                          DICE1020
208  H = P                                                               DICE1030
209  T2 = TP-T                                                           DICE1040
     IF(PART(T2,T2,1)) 210,212,210                                       DICE1050
210  H2 = DABS(T2)                                                       DICE1060
     IF(PART(TP,TP,1)) 211,213,211                                       DICE1070
211  IF(DABS(T2/TP)-.00001D0) 212,213,213                                DICE1080
212  T = TP                                                              DICE1090
     DT = H                                                              DICE1100
     L = 3                                                               DICE1110
     GO TO 402                                                           DICE1120
213  M = 0                                                               DICE1130
     J = 1                                                               DICE1140
     IF(H1-H2) 215,215,214                                               DICE1150
214  MU = 1                                                              DICE1160
     H = T2                                                              DICE1170
215  DT = H                                                              DICE1180
300  IG = 2                                                              DICE1190
     GO TO 102                                                           DICE1200
400  MU = 2                                                              DICE1210
```

```
      H = P                                                        DICE1220
      DT = H                                                       DICE1230
      N = NN                                                       DICE1240
  401 IG = 1                                                       DICE1250
      L = 1                                                        DICE1260
  402 TT = T                                                       DICE1270
      LL = L                                                       DICE1280
      RETURN                                                       DICE1290
      END                                                          DICE1300
```

```
$IBFTC DING     DECK
CDING       DICE INTEGRATION SUBROUTINE                                 DING0000
C                I = 2     SECOND ORDER RUNGE-KUTTA                      DING0010
C                I = 3     THIRD ORDER RUNGE-KUTTA                       DING0020
C                I = 4     FOURTH ORDER RUNGE-KUTTA                      DING0030
C           STORAGE     F1 = E     = Z1                                  DING0040
C                       F2 = YHAF1          TEMPORARY STORAGE REQUIRED=  DING0050
C                       F3 = YFULL          DIMENSION OF F ARRAY =       DING0060
C                       F4 = YSAVE             N*(3+I)                   DING0070
C                       F5 = DYSAVE         WHERE N = NO OF DERIVATIVES  DING0080
C                       F6 = Z2             AND I = ORDER OF INTEGRATION DING0090
C                       F7 = Z3                          PROCESS         DING0100
      SUBROUTINE DINTEG (T,TD,N,Y,DY,F,J,II)                             DING0110
      DIMENSION Y(50),DY(50),F(350)                                      DING0120
C THE FOLLOWING STATEMENT(S) HAVE BEEN MANUFACTURED BY THE TRANSLATOR---DING0130
C                                                                        DING0140
      DOUBLE PRECISION  Y      , DY      , F       , DT      , TD        DING0150
      DOUBLE PRECISION  T                                                DING0160
C     DIMENSION Y(50),DY(50),F(350)                                      DING0170
      I = II                                                             DING0180
      DT = TD                                                            DING0190
      DO 100 K = 1,N                                                     DING0200
      K1 = K                                                             DING0210
      K2 = K+5*N                                                         DING0220
      K3 = K2+N                                                          DING0230
      K4 = K + N                                                         DING0240
      GO TO (999,85,95,95),I                                             DING0250
 85   GO TO (86,2,999,999),J                                             DING0260
 86   F(K1) = DY(K)*DT                                                   DING0270
      Y(K) = F(K4) + F(K1)                                               DING0280
      GO TO 100                                                          DING0290
 95   GO TO (1,2,3,4),J                                                  DING0300
 1    F(K1) = DY(K)*DT                                                   DING0310
      Y(K) = F(K4)+.5D0*F(K1)                                            DING0320
      GO TO 100                                                          DING0330
 2    F(K2) = DY(K)*DT                                                   DING0340
      GO TO (999,22,23,24),I                                             DING0350
 3    F(K3) = DY(K)*DT                                                   DING0360
      GO TO (999,33,33,34),I                                             DING0370
 4    Y(K) = F(K4)+(F(K1)+2.D0*(F(K2)+F(K3))+DY(K)*DT)/6.D0              DING0380
```

```
      GO TO 100                                          DING0390
   22 Y(K) = .5D0*(F(K1)+F(K2))                          DING0400
      GO TO 25                                           DING0410
   23 Y(K) = 2.D0*F(K2)-F(K1)                            DING0420
      GO TO 25                                           DING0430
   24 Y(K) = .5D0*F(K2)                                  DING0440
   25 Y(K) = Y(K)+F(K4)                                  DING0450
      GO TO 100                                          DING0460
   33 Y(K) = F(K4)+(F(K1)+4.D0*F(K2)+F(K3))/6.D0         DING0470
      GO TO 100                                          DING0480
   34 Y(K) = F(K4)+F(K3)                                 DING0490
  100 CONTINUE                                           DING0500
      GO TO (110,120,130,140),J                          DING0510
  110 GO TO (999,131,132,132),I                          DING0520
  120 GO TO (999,140,132,140),I                          DING0530
  130 GO TO (999,140,140,132),I                          DING0540
  131 T = DT + T                                         DING0550
      GO TO 140                                          DING0560
  132 T = T + .5D0*DT                                    DING0570
  140 RETURN                                             DING0580
  999 PAUSE                                              DING0590
      GO TO 140                                          DING0600
      END                                                DING0610
```

```
$IBMAP RANN     DECK
       ENTRY RANNUM
RANNUM SXD CHRIS,4
       CLA 3,4                SIGMA
       STA A                                                     SRNRT0018
       CLA 4,4                X MEAN
       STA B                                                     SRNRT0020
       CLA 5,4
       STO C                                                     SRNRT0022
       TSX UDRNRT,4           COMPUTE UI                         SRNRT0023
       TXI *+3,,0
       PZE 26,0,0
 C     HTR 0
       STO D                                                     SRNRT0025
       LRS 35                                                    SRNRT0026
       FMP E                                                     SRNRT0027
       FAD F                                                     SRNRT0028
 S     SSM                                                       SRNRT0029
       FAD F                                                     SRNRT0030
       LRS 35                                                    SRNRT0031
       FMP G                                                     SRNRT0032
       STO ARG
       TSX ALOG,4
       TXI *+3,,0
       PZE 27,0,0
       HTR ARG
       LRS 35                                                    SRNRT0034
       FMP E                                                     SRNRT0035
       STO H                                                     SRNRT0036
       TSX SQRT,4                                                SRNRT0037
       TXI *+3,,0
       PZE 28,0,0
       HTR H
       STO P                                                     SRNRT0038
       LRS 35                                                    SRNRT0039
       FMP H                                                     SRNRT0040
       STO Q                                                     SRNRT0041
       CLA F                                                     SRNRT0042
       STO R                                                     SRNRT0043
       LXA S,4                COMPUTE NI                         SRNRT0044
```

```
V       LDQ T,4                     SRNRT0045
        FMP R,4                     SRNRT0046
        FAD R                       SRNRT0047
        STO R                       SRNRT0048
        TIX V,4,1                   SRNRT0049
        CLA T                       SRNRT0050
        STO W                       SRNRT0051
        LXA X,4                     SRNRT0052
Y       LDQ D,4                     SRNRT0053
        FMP Q,4                     SRNRT0054
        FAD W                       SRNRT0055
        STO W                       SRNRT0056
        TIX Y,4,1                   SRNRT0057
        FDP R                       SRNRT0058
        STQ R                       SRNRT0059
        CLA R                       SRNRT0060
X       CHS                         SRNRT0061
        FAD P                       SRNRT0062
        STO W                       SRNRT0063
        CLA D                       SRNRT0064
        FSB G                       SRNRT0065
        TPL Z                       SRNRT0066
        CLS W                       SRNRT0067
        TRA AA                      SRNRT0068
Z       CLA W                       SRNRT0069
AA      LRS 35                      SRNRT0070
A       FMP 0
B       FAD 0
        LXD CHRIS,4
        TRA 1,4
CHRIS   PZE
ARG     PZE
E       DEC -2.                     SRNRT0075
F       DEC 1.                      SRNRT0076
G       DEC .5                      SRNRT0077
        DEC 1.432788                SRNRT0078
        DEC .189269                 SRNRT0079
        DEC .001308                 SRNRT0080
T       DEC 2.515517                SRNRT0081
        DEC .802853                 SRNRT0082
```

```
        DEC .010328                                                     SRNRT0083
D       HTR 0
P       HTR 0
H       HTR 0
Q       HTR 0
R       HTR 0
W       HTR 0                                                           SRNRT0090
        END
```

```
$IBMAP UDRN    DECK
       ENTRY UDRNRT
UDRNRT SXD CHRIS,4
       CLA 3,4
       STA A                                            UDRN0015
       STA B                                            UDRN0016
       STA C                                            UDRN0017
       STA D                                            UDRN0018
       STA E                                            UDRN0019
       STA K                                            UDRN0020
 A     CLA 0
       ANA MASK                                         UDRN0022
       TNZ C                                            UDRN0023
 K     CLA 0
       ARS 18                                           UDRN0025
 B     STO 0
 C     LDQ 0
       MPY F                                            UDRN0028
 D     STQ 0
 E     CLA 0
       ARS 8                                            UDRN0031
       ADD G                                            UDRN0032
       FAD H                                            UDRN0033
       LXD CHRIS,4
       TRA 1,4
CHRIS  PZE
 MASK  HTR -1                                           UDRN0036
 F     OCT 011060471625                                 UDRN0037
 G     OCT 200000000000                                 UDRN0038
 H     OCT 0                                            UDRN0039
       END                                              UDRN0040
```

```
$IBFTC ECCAN
C          SUBROUTINE ECCAN
      DOUBLE PRECISION FUNCTION ECCAN(XM,E)
      DOUBLE PRECISION EPSIL,XM,E,DELE
      EPSIL = .5*10.**(-13)
      ECCAN = XM+E*DSIN(XM)*(1.+E*DCOS(XM))
 2    DELE = (XM-ECCAN+E*DSIN(ECCAN))/(1.-E*DCOS(ECCAN))
      ECCAN = ECCAN+DELE
      IF(DABS(DELE/ECCAN)-EPSIL) 1,1,2
 1    RETURN
      END
```

```
$IBFTC CEE
      SUBROUTINE CEE (Z1,Z2,Z3,C)
      DOUBLE PRECISION Z1,Z2,Z3,C,CZ1,CZ2,CZ3,SZ1,SZ2,SZ3
      DIMENSION C(3,3)
      SZ1 = DSIN(Z1)
      CZ1 = DCOS(Z1)
      SZ2 = DSIN(Z2)
      CZ2 = DCOS(Z2)
      SZ3 = DSIN(Z3)
      CZ3 = DCOS(Z3)
      C(1,1) = CZ3*CZ2
      C(1,2) = SZ3*CZ1+CZ3*SZ2*SZ1
      C(1,3) = SZ3*SZ1-CZ3*SZ2*CZ1
      C(2,1) = -SZ3*CZ2
      C(2,2) = CZ3*CZ1-SZ3*SZ2*SZ1
      C(2,3) = CZ3*SZ1+SZ3*SZ2*CZ1
      C(3,1) = SZ2
      C(3,2) = -CZ2*SZ1
      C(3,3) = CZ2*CZ1
      RETURN
      END
```

```
$IBFTC EXTRA
      SUBROUTINE EXTRA (B,HB,G,HG,SPRM,HSPRM,HSDP)
      DOUBLE PRECISION B,HB,G,HG,SPRM,HSPRM,HSDP,RR,ESUM
      DIMENSION B(3,3),HB(3,3),G(3,3),HG(3,3),SPRM(3),HSPRM(3),HSDP(3),
     1RR(3,3)
      DO 827 I  = 1,3
      DO 827 J  = 1,3
  827 RR(I,J)    = B(1,I)*HB(1,J)+B(2,I)*HB(2,J)+B(3,I)*HB(3,J)
      ESUM = DSQRT(RR(1,2)**2+RR(2,3)**2+RR(3,1)**2)
      WRITE (6,1062) ((HB(I,J),J=1,3),(  RR (I,K),K=1,3),I=1,3),ESUM
 1062 FORMAT (1H0,22X,9HHB-MATRIX,45X,12HERROR MATRIX/1H0,3(6F18.8/1H )/
     11H0,72X,F18.8)
 1063 FORMAT (1H0,3F18.8)
      RETURN
      END
```

```
$IBFTC CHECK1
      SUBROUTINE CHECK1
      COMMON/OMEGA/ HWO1,HWO2,HWO3,RT1,RT2,ZB11,ZB13,ZB21,ZB23
      COMMON/GIVE/NT,NZ,OAPZ,VAV,VQ,OEZ,ZE1,ZE2,ZE3,D,VI,
     1ETAZ,T,SSP,B,C,PSUBP,Z1,Z2,Z3,WO1,WO2,WO3
      COMMON/GET/NSUBB,NSUBC,NSUBM,NVAR,EM,PSI11,PSI13,PSI21,PSI23
     1,WSAP,WSBP,ASKH,ASUBK,WSUBP,ASCP,ASSP,CAHV,SAHV
      DIMENSION D(3,3),VI(3),SSP(3),B(3,3),EM(3,3),C(3,3),V(3),
     1VDP(3),QQ(3,3),RR(3,3)
      DOUBLE PRECISION TERM2,HWO1,HWO2,HWO3,VXH,COSAKV,PI
      DOUBLE PRECISION V,VDP,QQ,RR,EM,ZE1,ZE2,ZE3,
     1D,VI,ETAZ,T,SSP,B,C,PSUBP,Z1,Z2,Z3
      DOUBLE PRECISION WSAP,WSBP,ASCP,ASSP,TERM1,CAHV,WSUBP,ASKH,
     1ASKV,ASUBK,ASHV,SAHV,SAK,CAK,SAKH,CAKH,AHV,RT1,RT2,R11,R21,R31,
     2R41,R13,R23,R33,R43,ZB11,ZB13,ZB21,ZB23,PSI11,PSI13,PSI21,PSI23
      PI = 3.141592653589793
      IF(NT)601,601,602
  602 NSUBM=2
      NSUBB = 2
      NVAR=48
      GO TO 650
  601 IF(DABS(ZE1)-.0000001) 603,603,602
  603 NSUBM=1
      NVAR=12
      WSAP=-ZE2*WO1
      WSBP=DSQRT((ZE2+1.)**2*WO1**2+WO2**2+WO3**2)
      WSBP = -WSBP
      ASCP=-(ZE2+1.)*WO1/WSBP
      ASSP=DSQRT(1.-ASCP**2)
      IF(VAV)604,604,605
  605 GO TO(606,607),NZ
  606 IF(OAPZ-40000.)608,608,607
  608 NSUBB=2
      GO TO 650
  604 NSUBB=1
      DO 610 I=1,3
      V(I)=D(I,2)
  610 VDP(I)=C(2,I)
      TERM1=VI(2)*(WO2*VDP(2)+WO3*VDP(3))
      CAHV=ASCP*(VI(1)*WO1*VDP(1)+TERM1)/(VI(1)*WO1)
```

```
      TERM1=ASCP*(1.-1.5*ASSP**2)*ZE2
      WSUBP = 0.0
      GO TO 615
607   IF(T-1.E 06)616,616,617
617   NSUBB=2
      GO TO 650
616   NSUBB=1
      DO 618 I=1,3
      VDP(I)=SSP(I)
618   V(I)=B(I,1)*SSP(1)+B(I,2)*SSP(2)+B(I,3)*SSP(3)
      WSUBP = 0.0
615   ASKH=DATAN2(WO2,WO3)
      TERM1 = VDP(3)*VI(1)*WO1-VDP(1)*VI(3)*WO3
      VXH = DSQRT((VDP(2)*VI(3)*WO3-VDP(3)*VI(2)*WO2)**2+TERM1**2
     1+(VDP(1)*VI(2)*WO2-VDP(2)*VI(1)*WO1)**2)
      COSAKV = TERM1/VXH
      TERM2 = (VDP(2)*VI(1)**2*WO1**2)/ASCP**2-VI(2)*WO2*
     1(VDP(1)*VI(1)*WO1+VDP(2)*VI(2)*WO2+VDP(3)*VI(3)*WO3)
      IF (COSAKV) 620,6204,620
620   ASKV = (TERM2/DABS(TERM2))*DATAN2(DSQRT(1.-COSAKV**2),COSAKV)
      GO TO 6203
6204  ASKV = (PI*TERM2)/(2.*DABS(TERM2))
6203  ASUBK = ASKV-ASKH
      TERM1=VI(2)*(WO2*VDP(2)+WO3*VDP(3))
      TERM2=ASCP*(VI(1)*WO1*VDP(1)+TERM1)/(VI(1)*WO1)
      IF(DABS(TERM2)-1.) 6231,6231,624
624   WRITE(6,1022)TERM2,(VDP(I),I=1,3),WO1,WO2,WO3,VI(1),VI(2),ASCP
      GO TO 999
6231  IF(TERM2) 623,6232,623
623   AHV = DATAN2(DSQRT(1.-TERM2**2),TERM2)
      GO TO 6233
6232  AHV = PI/2.
6233  CAHV = DCOS(AHV)
      SAHV=DSIN(AHV)
      SAK=DSIN(ASUBK)
      CAK=DCOS(ASUBK)
      SAKH=DSIN(ASKH)
      CAKH=DCOS(ASKH)
      QQ(1,1)=CAHV
      QQ(1,2)=0.0
```

```
      QQ(1,3)=-SAHV
      QQ(2,1)=SAK*SAHV
      QQ(2,2)=CAK
      QQ(2,3)=SAK*CAHV
      QQ(3,1)=CAK*SAHV
      QQ(3,2)=-SAK
      QQ(3,3)=CAK*CAHV
      RR(1,1)=ASCP*QQ(1,1)-ASSP*QQ(3,1)
      RR(1,2)=ASCP*QQ(1,2)-ASSP*QQ(3,2)
      RR(1,3)=ASCP*QQ(1,3)-ASSP*QQ(3,3)
      RR(2,1)=QQ(2,1)
      RR(2,2)=QQ(2,2)
      RR(2,3)=QQ(2,3)
      RR(3,1)=ASSP*QQ(1,1)+ASCP*QQ(3,1)
      RR(3,2)=ASSP*QQ(1,2)+ASCP*QQ(3,2)
      RR(3,3)=ASSP*QQ(1,3)+ASCP*QQ(3,3)
      QQ(1,1)=RR(1,1)
      QQ(1,2)=RR(1,2)
      QQ(1,3)=RR(1,3)
      QQ(2,1)=CAKH*RR(2,1)+SAKH*RR(3,1)
      QQ(2,2)=CAKH*RR(2,2)+SAKH*RR(3,2)
      QQ(2,3)=CAKH*RR(2,3)+SAKH*RR(3,3)
      QQ(3,1)=-SAKH*RR(2,1)+CAKH*RR(3,1)
      QQ(3,2)=-SAKH*RR(2,2)+CAKH*RR(3,2)
      QQ(3,3)=-SAKH*RR(2,3)+CAKH*RR(3,3)
      DO 627 I=1,3
      DO 627 J=1,3
  627 EM(I,J)=B(I,1)*QQ(1,J)+B(I,2)*QQ(2,J)+B(I,3)*QQ(3,J)
C     C-MATRIX CHECK
  650 GO TO(651,652),NZ
  652 IWHY=1
  670 WRITE(6,1032)IWHY
      NSUBC=2
      GO TO 699
  999 CALL EXIT
  651 IF(VAV)653,653,672
  653 IF(DEZ)654,654,673
  654 IF(Z1*Z1+Z2*Z2+Z3*Z3-.01)655,655,674
  655 TERM1=DABS(WO2-ETAZ)
      TERM2=DSQRT(3.*ZE2)*.1*ETAZ
```

```
      IF(TERM1-TERM2)656,656,675
656   IF(VI(1)-VI(3))676,676,657
657   IF(ZE1*ZE3)658,677,677
658   TERM1 = 1.-3.*ZE1-ZE1*ZE3
      IF (TERM1) 678,678,659
659   TERM2=TERM1**2+16.*ZE1*ZE3
      IF(TERM2)679,679,680
672   IWHY=2
      GO TO 670
673   IWHY=3
      GO TO 670
674   IWHY=4
      GO TO 670
675   IWHY=5
      GO TO 670
676   IWHY=6
      GO TO 670
677   IWHY=7
      GO TO 670
678   IWHY=8
      GO TO 670
679   IWHY=9
      GO TO 670
680   RT1 = ETAZ*DSQRT((TERM1+DSQRT(TERM2))/2.)
      RT2 = ETAZ*DSQRT((TERM1-DSQRT(TERM2))/2.)
      TERM1=RT1**2-RT2**2
      R11=(RT1**2-ETAZ**2*(1.+ZE1-ZE1*ZE3))/TERM1
      R21=(RT2**2-ETAZ**2*(1.+ZE1-ZE1*ZE3))/TERM1
      R31 = ((1.+ZE1)*ZE3*ETAZ**3)/(RT1*TERM1)
      R41 = ((1.+ZE1)*ZE3*ETAZ**3)/(RT2*TERM1)
      R13=(RT1**2-ETAZ**2*(1.-ZE3-3.*ZE1-ZE1*ZE3))/TERM1
      R23=(RT2**2-ETAZ**2*(1.-ZE3-3.*ZE1-ZE1*ZE3))/TERM1
      R33=(4.*(1.-ZE3)*ZE1*ETAZ**3)/(RT1*TERM1)
      R43=(4.*(1.-ZE3)*ZE1*ETAZ**3)/(RT2*TERM1)
      ZB11=DSQRT((R11*Z1)**2+(R31*Z3)**2)
      ZB13=DSQRT((R33*Z1)**2+(R13*Z3)**2)
      ZB21=DSQRT((R21*Z1)**2+(R41*Z3)**2)
      ZB23=DSQRT((R43*Z1)**2+(R23*Z3)**2)
      IF(ZB11-.1)681,681,682
681   IF(ZB13-.1)683,683,682
```

```
683   IF(ZB21-.1)684,684,682
684   IF(ZB23-.1)685,685,682
682   IWHY=10
      GO TO 670
685   TERM1=DABS(WO1+ETAZ*Z3)
      TERM2=.01*(RT1*ZB11+RT2*ZB21)
      IF(TERM1-TERM2)686,686,687
687   IWHY=11
      GO TO 670
686   TERM1=DABS(WO3-ETAZ*Z1)
      TERM2=.01*(RT1*ZB13+RT2*ZB23)
      IF(TERM1-TERM2)688,688,689
689   IWHY=12
      GO TO 670
688   NSUBC=1
      TERM1= R31*Z3
      TERM2=R11*Z1
      PSI11=DATAN2(TERM1,TERM2)
      TERM1= R33*Z1
      TERM2=R13*Z3
      PSI13=DATAN2(TERM1,TERM2)
      TERM1= R41*Z3
      TERM2=R21*Z1
      PSI21=DATAN2(TERM1,TERM2)
      TERM1= R43*Z1
      TERM2=R23*Z3
      PSI23=DATAN2(TERM1,TERM2)
699   CONTINUE
1021 FORMAT(1H0,18HTROUBLE IN B-CHECK/1H0,7F16.8)
1022 FORMAT(1H0,17HTROUBLE IN ARCCOS/1H0,9F13.8)
1032 FORMAT(1H0,30HC-MATRIX CHECK OMITTED DUE TO ,I3)
      RETURN
      END
```

```
$IBFTC HELP
      SUBROUTINE HELP
      COMMON/OMEGA/ HWO1,HWO2,HWO3,RT1,RT2,ZB11,ZB13,ZB21,ZB23
      COMMON/GIVE/NT,NZ,OAPZ,VAV,VQ,OEZ,ZE1,ZE2,ZE3,D,VI,
     1ETAZ,T,SSP,B,C,PSUBP,Z1,Z2,Z3,WO1,WO2,WO3
      COMMON/GET/NSUBB,NSUBC,NSUBM,NVAR,EM,PSI11,PSI13,PSI21,PSI23
     1,WSAP,WSBP,ASKH,ASUBK,WSUBP,ASCP,ASSP,CAHV,SAHV
      COMMON/XCIN/X,TAU,CHX,SHX,HG,GAM,STHE,CTHE,THETA,COMC,SOMC,
     1CIC,SIC,G,SPRM,PM,PP,PMDUM,TM,CWS,PBO,QKB,QLBC,QKBP,R,SR,QKBDR,
     2BETA,CSUBB,SRCUB,BETAP,OWZ
      DIMENSION X(48)
      DIMENSION C(3,3),D(3,3),B(3,3),SSP(3),VI(3),EM(3,3),HX(6)
      DIMENSION DELL(3),DELK(3),DELA(3),DELB(3),DELG1(3),DELG2(3),
     1DELG3(3),DELF1(3),DELF2(3),DELX1(3),DELX2(3),DELX3(3),CHX(3),
     2SHX(3),HG(3,3),PH(6,6),GAM(3,3),G(3,3),BDUM(3,3),SPRM(3),PM(6,6),
     3PP(6,6),PMDUM(6,6),QKB(3),QKBP(3),R(3),BETA(3),BETAP(3)
      DOUBLE PRECISION WSAP,WSBP,ASKH,ASUBK,WSUBP,ASCP,ASSP,CAHV,SAHV,
     1HWO1,HWO2,HWO3,RT1,RT2,ZB11,ZB13,ZB21,ZB23
      DOUBLE PRECISION X
      DOUBLE PRECISION ZE1,ZE2,ZE3,TAU,HX,      D,VI,ETAZ,T,SSP,B,C,
     1PSUBP,Z1,Z2,Z3,EM,PSI11,PSI13,PSI21,PSI23
      DOUBLE PRECISION OSA,OSB,ZSQ,ALSC,ALSS,OYZ,QSC,QSS,TTT,ABC,ABSI,
     1BBC,BBS,F1,F2,CTHE,STHE,THETA,YOUC,YOUF,COMC,SOMC,CIC,SIC,
     2TM,CWS,PBO,PSIB,QLBC,SR,QKBDR,CSUBB,SRCUB
      DOUBLE PRECISION DELL,DELK,DELA,DELB,DELG1,DELG2,DELG3,DELF1,
     1DELF2,DELX1,DELX2,DELX3,CHX,SHX,HG,PH,GAM,G,BDUM,SPRM,PM,PP,
     2PMDUM,QKB,QKBP,R,BETA,BETAP
      EQUIVALENCE (X(7),HX(1)),(PH(1),X(13))
      GO TO (821,822),NSUBM
  821 ZSQ=ZE2
      OSA=-ZSQ*HWO1
      OSB=DSQRT((ZSQ+1.)**2*HWO1**2+HWO2**2+HWO3**2)
      OSB = -OSB
      ALSC=-(ZSQ+1.)*HWO1/OSB
      ALSS=DSQRT(1.-ALSC**2)
      OYZ=DSQRT(HWO2**2+HWO3**2)
      QSC=HWO3/OYZ
      QSS=HWO2/OYZ
      TTT = TAU
      ABC=DCOS(OSA*TTT)
```

```
      ABSI=DSIN(OSA*TTT)
      BBC=DCOS(OSB*TTT)
      BBS=DSIN(OSB*TTT)
      DELL(1) =-(OYZ*ALSC**2)/((ZSQ+1.)*HWO1**2)
      DELL(2) = ( HWO2*ALSC**2)/((ZSQ+1.)*HWO1*OYZ)
      DELL(3) = ( HWO3*ALSC**2)/((ZSQ+1.)*HWO1*OYZ)
      DELK(1)=0.0
      DELK(2)=HWO3/OYZ**2
      DELK(3)=-HWO2/OYZ**2
      DELA(1)=-ZSQ*TTT
      DELA(2)=DELK(2)
      DELA(3)=DELK(3)
      DELB(1)=((ZSQ+1.)**2*HWO1*TTT)/OSB
      DELB(2)=HWO2*TTT/OSB
      DELB(3)=HWO3*TTT/OSB
      DO 824 J=1,3
      DELG1(J)=-2.*ALSC*ALSS*DELL(J)*(1.-BBC)-ALSS**2*BBS*DELB(J)
      DELG2(J)=ALSS*ALSC*QSS*BBS*DELB(J)+(1.-BBC)*(ALSC**2*DELL(J)*QSS-
     1ALSS**2*DELL(J)*QSS+ALSS*ALSC*QSC*DELK(J))-(ALSC*DELL(J)*QSC*
     2BBS-ALSS*QSS*DELK(J)*BBS+ALSS*QSC*BBC*DELB(J))
      DELG3(J)=ALSS*ALSC*QSC*BBS*DELB(J)+(1.-BBC)*(ALSC**2*DELL(J)*QSC-
     1ALSS**2*DELL(J)*QSC-ALSS*ALSC*QSS*DELK(J))+(ALSC*DELL(J)*QSS*
     2BBS+ALSS*QSC*DELK(J)*BBS+ALSS*QSS*BBC*DELB(J))
  824 CONTINUE
      F1=ALSS**2*QSC*(1.-BBC)+QSC*BBC-ALSC*QSS*BBS
      F2=QSS*BBC+ALSC*QSC*BBS
      DO 825 J=1,3
      DELF1(J)=ALSS**2*QSC*BBS*DELB(J)+(1.-BBC)*(-ALSS**2*QSS*DELK(J)+
     12.*ALSS*ALSC*DELL(J)*QSC)-QSC*BBS*DELB(J)-QSS*DELK(J)*BBC+ALSS*
     2DELL(J)*QSS*BBS-ALSC*QSC*DELK(J)*BBS-ALSC*QSS*BBC*DELB(J)
      DELF2(J)=-QSS*BBS*DELB(J)+QSC*DELK(J)*BBC-ALSS*DELL(J)*QSC*BBS-
     1ALSC*QSS*DELK(J)*BBS+ALSC*QSC*BBC*DELB(J)
  825 CONTINUE
      DO 826 J=1,3
      DELX1(J)=(F1*DELF2(J)-F2*DELF1(J)-(F1*F1+F2*F2)*DELA(J))/CHX(2)**2
      DELX2(J)=DELG3(J)/CHX(2)
  826 DELX3(J)=(HG(2,1)*DELG1(J)-HG(1,1)*DELG2(J))/CHX(2)**2
      PH(1,1)=CHX(3)/CHX(2)
      PH(1,2)=-SHX(3)/CHX(2)
      PH(1,3)=0.0
```

```
      PH(2,1)=SHX(3)
      PH(2,2)=CHX(3)
      PH(2,3)=0.0
      PH(3,1)=-CHX(3)*SHX(2)/CHX(2)
      PH(3,2)=SHX(3)*SHX(2)/CHX(2)
      PH(3,3)=1.0
      DO 827 K=4,6
      PH(1,K)=DELX1(K-3)
      PH(2,K)=DELX2(K-3)
  827 PH(3,K)=DELX3(K-3)
      PH(4,4)=1.0
      PH(4,5)=0.0
      PH(4,6)=0.0
      PH(5,4)=ZSQ*HWO2*TTT*ABSI-ZSQ*HWO3*TTT*ABC
      PH(5,5)=ABC
      PH(5,6)=ABSI
      PH(6,4)=ZSQ*HWO3*TTT*ABSI+ZSQ*HWO2*TTT*ABC
      PH(6,5)=-ABSI
      PH(6,6)=ABC
      HWO1=HX(4)
      HWO2=HX(5)
      HWO3=HX(6)
C     END OF CLOSED FORM TRANSITION MATRIX
  822 CONTINUE
      DO 302 I = 1,3
      GAM(I,1) =-C(3,I)*STHE+C(1,I)*CTHE
  302 GAM(I,2) = C(2,I)
      YOUC = DCOS(OWZ+THETA)
      YOUF = DSIN(OWZ+THETA)
      D(1,1) = -COMC*YOUF-SOMC*CIC*YOUC
      D(1,2) = SOMC*SIC
      D(1,3) = COMC*YOUC-SOMC*CIC*YOUF
      D(2,1) = -SOMC*YOUF+COMC*CIC*YOUC
      D(2,2) = -COMC*SIC
      D(2,3) = SOMC*YOUC+COMC*CIC*YOUF
      D(3,1) = SIC*YOUC
      D(3,2) = CIC
      D(3,3) = SIC*YOUF
      DO 305 I = 1,3
      DO 305 J = 1,3
```

```
305   C(I,J) = GAM(1,I)*G(1,J)+GAM(2,I)*G(2,J)+GAM(3,I)*G(3,J)
      DO 306 I = 1,3
      DO 306 J = 1,3
306   BDUM(I,J) = B(I,1)*G(1,J)+B(I,2)*G(2,J)+B(I,3)*G(3,J)
      DO 308 K = 1,3
      DO 308 L = 1,3
308   B(K,L) = BDUM(K,L)
      DO 310 I = 1,3
310   SSP(I) = G(1,I)*SPRM(1)+G(2,I)*SPRM(2)+G(3,I)*SPRM(3)
      DO 314 I = 1,6
      DO 314 J = 1,6
314   PMDUM(I,J) = PH(I,1)*PP(1,J)+PH(I,2)*PP(2,J)+PH(I,3)*PP(3,J)+
     1PH(I,4)*PP(4,J)+PH(I,5)*PP(5,J)+PH(I,6)*PP(6,J)
      DO 316 I = 1,6
      DO 316 J = 1,6
316   PM(I,J) = PMDUM(I,1)*PH(J,1)+PMDUM(I,2)*PH(J,2)+PMDUM(I,3)*
     1PH(J,3)+PMDUM(I,4)*PH(J,4)+PMDUM(I,5)*PH(J,5)+PMDUM(I,6)*PH(J,6)
      GO TO (321,322),NZ
321   PSIB = PBO+CWS*TM
      QKB(1) = QLBC*DCOS(PSIB)
      QKB(2) = QLBC*DSIN(PSIB)
      DO 325 I = 1,3
325   QKBP(I) = B(1,I)*QKB(1)+B(2,I)*QKB(2)+B(3,I)*QKB(3)
      DO 326 I = 1,3
326   R(I) = SR*D(I,3)
      QKBDR = QKB(1)*R(1)+QKB(2)*R(2)+QKB(3)*R(3)
      DO 327 J = 1,3
327   BETA(J) = CSUBB*(QKB(J)/SRCUB-3.*QKBDR*R(J)/(SRCUB*SR**2))
      DO 328 L = 1,3
328   BETAP(L) = B(1,L)*BETA(1)+B(2,L)*BETA(2)+B(3,L)*BETA(3)
322   RETURN
      END
```


NASA-Langley, 1966 CR-599