NASA Technical Paper 1145

The N-BOD2 User's and Programmer's Manual

Harold P. Frisch

FEBRUARY 1978

NASA

NASA Technical Paper 1145

The N-BOD2 User's and Programmer's Manual

Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1978

All measurement values are expressed in the International System of Units (SI) in accordance with NASA Policy Directive 2220.4, paragraph 4.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

THE N-BOD2 USER'S AND PROGRAMMER'S MANUAL

Harold P. Frisch
Goddard Space Flight Center
Greenbelt, Maryland

INTRODUCTION

The general purpose digital program N-BOD, and its present version N-BOD2, have been developed with the needs of the spacecraft attitude dynamicist foremost in mind. This fact, however, does not detract in any way from its applicability to a much broader class of problems. The program may be used to both derive and output in a vector-dyadic form, and/or to solve numerically the equations of motion of any system that may be adequately modeled, for the purposes of dynamic simulation, as a topological tree of point-connected rigid bodies, flexible bodies, point masses, and symmetrical momentum wheels (commonly called "symmetric wheel" in this document).

The underlying theoretical development upon which the program N-BOD2 is based may be found in References 1 and 2. In the FORTRAN IV coding of this program, a definite attempt has been made to draw a balance between available generality, ease of user operation, and computational speed and efficiency.

The program is structured to expand or contract with the required needs of the user. By an internal interrogation of the input data, the program sets up a series of logic flags that determine the computational options that should be used and those that should not.

In most practical problems, the user will be required to interface with the program. The output routine, which is called every integration step, prints virtually all system state variables along with numerous other physically meaningful parameters that are useful for initial checkout purposes. In setting up the output routine for production runs, the user is expected to literally delete the unwanted print statements from the output subroutine and insert his own.

If external forces and torques act on the system or if nongyroscopic forces and torques act internal to the system at hinge points, the user is required to define them. This will present no particular problem since a number of examples are provided which show exactly how most commonly occurring forces and torques are defined and coded.

Frequently programs already exist that define, for example, the workings of a complex attitude control system. State-variable information is required as input to it and motor torques are outputted from it. By a few simple interface statements, which create the proper state-variable parameters for the existing torque routine and the required force and torque parameters for N-BOD2, the debugged routine may be inserted in total into N-BOD2.

The input of data is provided for in three separate subroutines which are called sequentially. The first is of a rigid format. In this routine a basic coupled rigid body, point mass, and symmetric wheel model of the system is described. Connection topology, degrees of freedom, mass, inertia, and geometric parameters, along with relative body orientation and initial kinematic conditions, are inputted here. The second input routine is more flexible. Numerous options are available and may be called for by use of option cards. For example, some of the bodies may be redefined to be flexible; if so, the flexible body data would be inputted at this point. The third input routine is for the user. Any data that will be required for the torque and output routines, written by the user, are entered here. It may be left empty, or again the user may insert an existing input routine and set up the proper interface statements. It is permissible, and at times is convenient, for the user to use this routine to override some data already inputted in the first input routine.

When the analyst is first presented with a particular problem for dynamic simulation, he is usually provided with just enough information to create a topological model, define the character of each body in the model (rigid body, flexible body, point mass, or symmetric wheel), and to define the number of degrees of relative freedom between contiguous bodies. With this information and a fictitious set of mass and inertia properties, the user may obtain the equations of motion for the simulation model. At times, this is the end product.

With the outputted equations of motion, the user will immediately recognize familiar terms and will need only a cursory glance at the rather lengthy symbol list for exact definitions. There is a one-for-one correspondence between the outputted equations and those provided in References 1 and 2. For example, the outputted statements

$$\text{FOMC(5)} \times (\text{XIC(5)} \cdot \text{FOMC(5)})$$

and

$$\text{FOMC(3)} \times (\text{FOMC(3)} \times \text{CAC(3)})$$

translate to the familiar vector tensor expressions

$$\vec{\omega}_5 \times (\Phi_5 \cdot \vec{\omega}_5)$$

and

$$\vec{\omega}_3 \times (\vec{\omega}_3 \times \vec{\alpha}_3)$$

respectively.

To use this program effectively, the user must become familiar with the array names used to define the various physical characteristics and state variables in the equations of motion. This is most easily done by a study of the computer derived equations of motion. Once this is done the user is in a position to proceed to code the required input, torque, and output routines for his particular problem and to make use of the full power of N-BOD2.

THE SIMULATION MODEL

The analysis of the dynamic characteristics of a complex system is usually done through the study of a simulation model. The simulation model must possess the same dynamic characteristics as the complex system and at the same time be amenable to mathematical analysis.

Rigid bodies, point masses, and symmetric wheels are idealized bodies that readily lend themselves to mathematical analysis. Flexible bodies are more complex; however, if the deformation is small and the laws of linear elasticity can be applied, the dynamic characteristics can be simulated by the natural modes and frequencies of vibration, the determination of which is a subject in itself. [N-BOD2 simply assumes that the modes and frequencies of flexible bodies are obtainable; it accepts only resultant mode-dependent parameters in the input data stream (see Appendix B).]

N-BOD2 *assumes* that the complex system under study may be simulated by a model comprised of rigid bodies, point masses, symmetric wheels, and flexible bodies for which natural modes and frequencies of vibration are available. Contiguous rigid and flexible bodies are *assumed* to be point connected in such a manner that the total system forms a topological tree (no closed paths). Furthermore, point masses are *assumed* to exist only at limb ends while symmetric wheels are *assumed* to be imbedded within either rigid or flexible bodies.

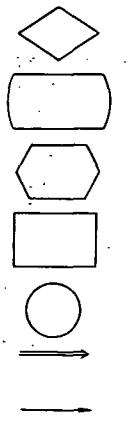
The present version of N-BOD2 has certain *limitations* with respect to the simulation of flexible bodies. The author has not coded the equations required to allow hinge points and wheel attachment points on flexible bodies to be time varying in the flexible body's body-fixed reference frame. This will be done if future simulation demands require it. It should be noted that the equations exist and can be found in Reference 2, in a form amenable for inclusion into N-BOD2.

The simulation model must reflect the fact that relative motion between contiguous bodies may be kinematically constrained. To do this, N-BOD2 *assumes* that contiguous rigid and flexible bodies are connected together by physically realizable 0-, 1-, 2-, or 3-degrees-of-freedom gimbals. The gimbals prohibit relative translational motion and permit, at most, 3 degrees of relative rotational freedom at hinge points. Point masses may have 0, 1, 2, or 3 degrees of relative translational freedom and symmetric wheels may have 1 degree of rotational freedom relative to the body in which they are imbedded. Flexible bodies may possess several degrees of vibrational freedom in addition to the degrees of freedom associated with the connection gimbals. Practical computer storage and speed limitations put a limit on the total number of vibrational modes allowed for the total system of coupled bodies.

PROGRAM OVERVIEW, MAIN PROGRAM, AND SUBROUTINE DYN

The basic flow logic of the program N-BOD2 is shown on the flowchart in figures 1 and 2 using the symbology listed below. Immediately upon entry into the MAIN program a single control card is read. The data on this card sets several logic flags that control the mode of data input and output.

Flow Chart Symbology



Conditional Statement

Call Subroutine

Call to I/O Device

Computation

Program Point

Primary Path for Program Flow

Secondary Path for Program Flow
(Usually option related or a once-only path for initial setup)

The input data may either define the physical and kinematic parameters for a new job that will start at time zero, or define the parameters needed to restart the program at the termination time of a previous run.

The output data may either be a listing of the system equations of motion or the output of selected system state variables. Upon reaching the termination time of a particular job, a restart tape may be requested. This tape contains all data required to restart the job at a later date.

If a restart run is to be made, the restart tape created at the end of a previous run becomes the input tape for the present run. It contains the magnitude of every computer-generated parameter required for a reinitiation of computation. The tape is read in and the program then immediately branches to the start of the integration loop and proceeds as if the previous run had never been terminated.

If a new run starting at time zero is to be made, three input subroutines are sequentially called. The first called is INBS. In this subroutine, the basic system is defined. The data are checked in INEROR for physical realizability, and then subroutine SETS is entered. In SETS, various integer sets are computed that enable the program to avoid redundant and trivial computation. Subroutine INOPT, the second input routine, is programmed to recognize various option codes. The available options fall into two categories: those that expand modeling capability, e.g., flexible body option, caged degree of freedom option; and those that attempt to

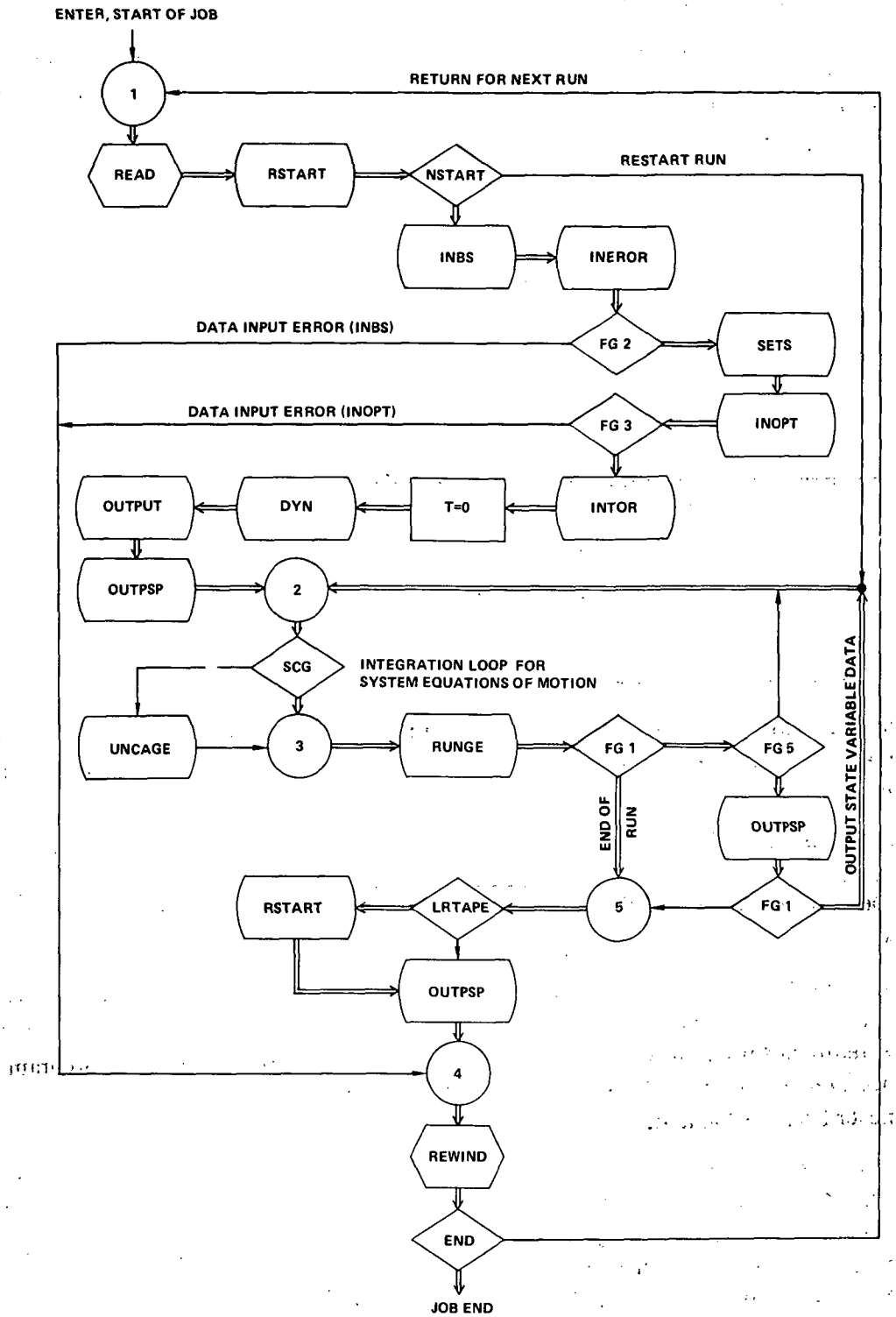


Figure 1. Main program flowchart.

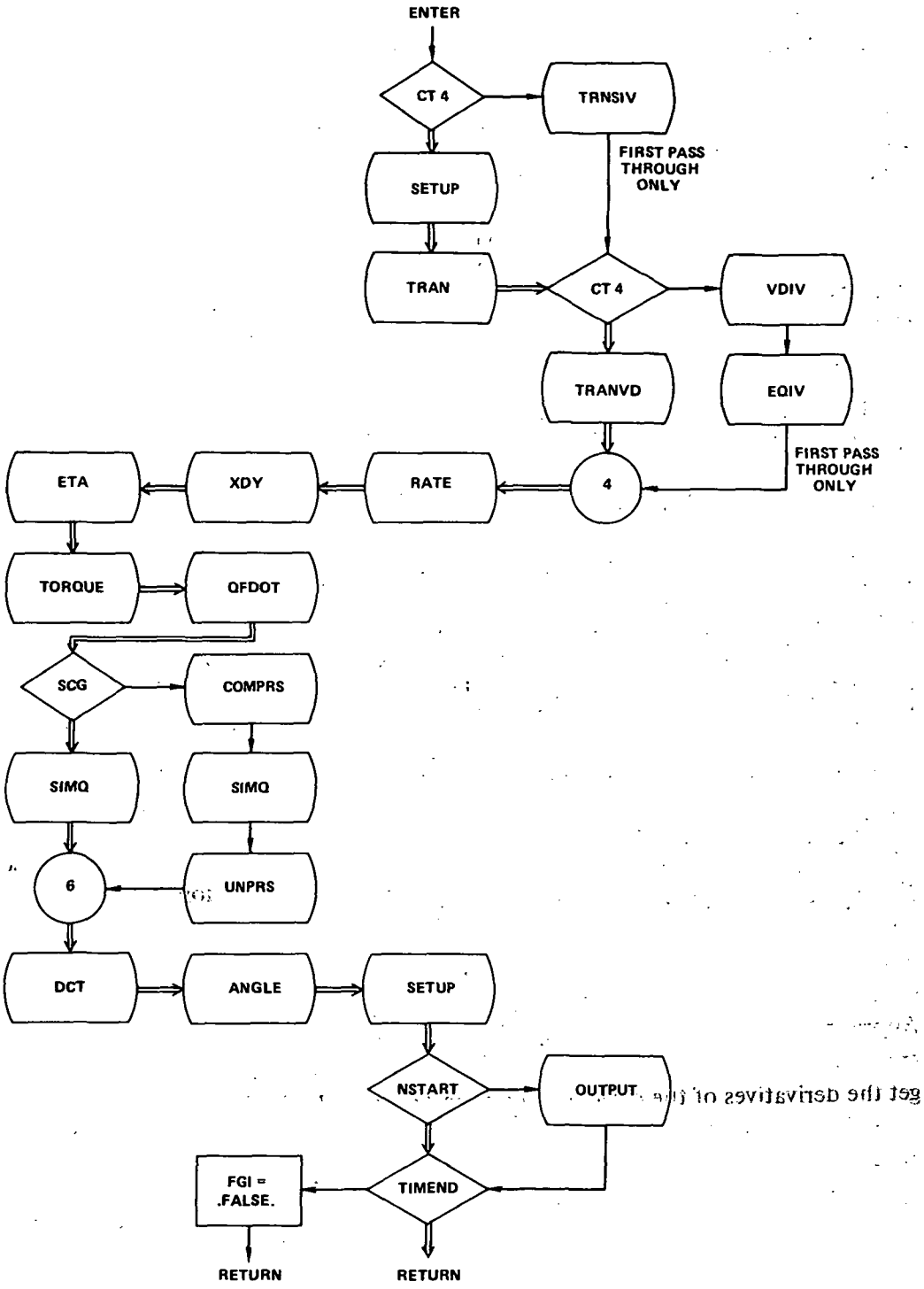


Figure 2. Subroutine DYN flowchart.

improve computational efficiency and run time by allowing the user to introduce engineering judgment in deleting computations that he feels will have a negligible impact upon computed results. The last input routine is INTOR. Normally this routine is empty. It is inserted for the user. Any input data that the user requires for a description of the nongyroscopic forces and torques acting on this system are entered here according to a user written code. The user's input data should be stored in common block /SATELL/ which is reserved for the user. Normally, it contains an empty dummy array of 1000 real double precision words.

With all necessary input data stored in computer memory, subroutine DYN is entered for the first time. The complete state of the system is computed at time zero and outputted. The primary integration loop is then entered. RUNGE is the integration package; it is a fixed-step, four-order Runge Kutta numerical integration routine that calls DYN to compute the equations of motion to be integrated.

At the end of each integration loop, a check is made to determine if the output routine OUTPSP should be entered. At the termination time (a flag is reset in DYN), the program branches out of the integration loop, creates a restart tape if called for, and terminates if there is no succeeding run to follow.

Subroutine DYN is used to call the subroutines in which the equations of motion are coded. Upon entry, a check is made to see if it is the first pass through; if so, certain initialization procedures must be carried out. Initial values for transformation matrices are obtained by TRNSIV; initial values for all vectors and dyads are obtained by VDIV; and initial values for all differential equations are defined in EQIV. If it is not the first pass through DYN, the call for DYN has been made by RUNGE. All of the system state variables obtained from RUNGE are sorted out in SETUP. TRAN computes all transformation matrices; TRANVD transforms all vectors and dyads to a common frame of computation; RATE computes all rate related vectors; XDY computes all inertia and pseudo inertia tensors on the left-hand side of the system equations of motion; and ETA computes all gyroscopic cross coupling torques found on the right-hand side of the equations.

Subroutine TORQUE is user defined; all nongyroscopic forces and torques are defined here in vector form. To reduce the equations to a set of simultaneous scalar equations, QFDOT is called. Upon exit, a check is made in DYN to see if any degrees of freedom are caged. Appropriate action is taken by COMPRS if they are caged, and the simultaneous equations are solved by SIMQ. All acceleration parameters are now at hand. DCT is then entered to get the derivatives of the direction cosine matrices, and ANGLE is entered to get the equations that define relative displacement. All differential equations are then stacked in a one-dimensional array by SETUP in a form acceptable to RUNGE and return to RUNGE is made after a time check.

PROGRAMMING TECHNIQUES USED

A definite attempt has been made to reflect as much of the subscript notation as possible that is used in References 1 and 2 in the FORTRAN IV coding of the equations. Deviations from this practice occur at points where a more streamlined subscript notation leads to a considerable saving in computer storage and an increase in computational speed. It should be realized that computation speed is severely compromised by the unnecessary use of multi-dimensioned arrays.

N-BOD2 is built on the basis that the only labeling which should be done by the user is the labeling of bodies and wheels. All other required labeling is done by the computer. The user, however, must be aware of the labeling rules that are used internally if he is to effectively interface with the program.

Table 1 provides a listing of the symbol names used in the coding of N-BOD2, their dimension, type, storage location, and a very brief description of each along with the subroutine in which they are first defined and/or used extensively. A detailed definition of each symbolic name may be found in the comment cards of the subroutine and/or in the subroutine discussion included in this report.

PROGRAMMER'S GUIDE TO MAIN PROGRAM AND SUBROUTINE DYN

The purpose of the MAIN program and subroutine DYN is to sequentially call the primary special purpose subroutines that read input data, define and integrate the equations of motion, and write output data. A detailed discussion of the flow logic is provided in the section entitled "Program Overview, MAIN Program, and Subroutine DYN."

The following variables defined in table 1 are either first introduced or used extensively in the MAIN program and subroutine DYN.

CT4 – a counter. Used to count the number of passes through subroutine DYN. It is updated upon entry into DYN. The fourth order fixed step Runge Kutta integration routine calls DYN four times per integration step, consequently

$$CT4 = 1 + 4 * (N - 1) + J$$

during and after the J*th* pass through DYN of the N*th* integration step.

CT4 = 1. During and after the initialization pass through DYN; i.e., when called by MAIN

FG1 – end of job flag. At the end of each integration step its status is checked in MAIN. If .TRUE., computation continues; if .FALSE., an end-of-run condition has been satisfied. If T.GE.TIMEND in DYN, FG1 is reset to .FALSE.. The user may define any other end-of-run condition he desires in subroutines TORQUE or OUTPSP.

Table 1
Symbol Names Used in N-BOD2

```
C
C
C
C     N-BOD2 IS DIMENSIONED TO ACCEPT A MAXIMUM OF
C       N - BODIES (FLEXIBLE BODIES + RIGID BODIES + POINT MASSES)
C       N - SYMMETRIC WHEELS
C      2N - FLEXIBLE MODES OF VIBRATION (TOTAL FOR ALL FLEXIBLE BODIES)
C      4N - MODAL CROSS-COUPLING COEFFICIENTS (TOTAL)
C      33 - INDEPENDENT DEGREES OF FREEDOM
C     160 - FIRST ORDER NON-LINEAR DIFFERENTIAL EQUATIONS
C
C         THIS VERSION OF N-BOD2 USES
C              N = 10
C
C     MAKING USE OF N.LT.10 SAVES CONSIDERABLE COMPUTER STORAGE
C                   N.GT.10 RUN TIME FOR PRATICAL APPLICATION EXCESSIVE
C
C
C
C           SYMBOL LIST ABBREVIATIONS
C     IDEM2 = N**2 + N + 1 - (N*(N-1))/2
C     IDEM3 = (N-1)**2 + N - ((N-1)*(N-2))/2
C     IDEM4 = SIZE OF /LOGIC/     16    LOGICAL WORDS
C     IDEM5 = SIZE OF /INTG/      724   INTEGER WORDS
C     IDEM6 = SIZE OF /INTGZ/     70    INTEGER WORDS
C     IDEM7 = SIZE OF /REAL/      4354  REAL WORDS
C     IDEM8 = SIZE OF /REALZ/     168   REAL WORDS
C     IDEM9 = SIZE OF /SATELL/    1000  REAL WORDS
C     ALL COMPUTED VECTORS AND TENSORS IN COMPUTING FRAME COORDINATES
C     NA = OPTION NOT AVAILABLE IN N-BOD2
C     # = NUMBER OF
C     BFC = BODY FIXED COORDINATES
C     CFC = COMPUTING FRAME FIXED COORDINATES (BODY 1 OR INERTIAL)
C     IFC = INERTIALLY FIXED COORDINATES
C     EQIV(XMN) = EQUIVALENCED TO XMN ARRAY
C     0:N = BY USE OF A DUMMY VARIABLE SUBSCRIPT 0 ALLOWABLE
C
C                       SYMBOL LIST AND STORAGE LOCATION
C
C  NAME      TYPE DIMENSION STORAGE    DEFINITION AND SUBROUTINE USED IN
C  ----      ---- --------- -------    ---------------------------------
C  ANGD        R  3(N+1)    EQIV(XMN)  EULER ANGLE DOT (ANGLE,SETUP)
C  AWORK       I  200       /INTG/     LOCAL WORK AREA TO SAVE STORAGE
C  CA          R  3,N       /REAL/     CM VECTOR BFC (INBS)
C  CAC         R  3,N       /REAL/     CM VECTOR CFC (VDIV,TRANVD)
```

Table 1 (Continued)
Symbol Names Used in N-BOD2

```
C   CAO           R   3,N        /REAL/      ZERO DEF CM VECT BFC (VDIV)
C   CBDUM,CB      R   3,0:N      /REALZ/     HINGE VECTOR BFC (INBS)
C   CBCDUM,CBC    R   3,0:N      /REALZ/     HINGE VECTOR CFC (VDIV,TRANVD)
C   CBN           R   3          /REALZ/     HINGE VECTOR PART (INBS)
C   CLM           R   N          /REAL/      SCALAR TORQUE ON WHEEL (TORQUE)
C   CNF           R   3,N        EQIV(XMN)   FORCE CENTRIPETAL + CORIOLIS(ETA)
C   COMC          R   3,N+1      /REAL/      ANG RATE TO COMP FRAME (RATE,DCT)
C   CT1           I              /INTG/      COUNTER (INOPT) UNUSED AFTER
C   CT2           I              /INTG/      COUNTER (INOPT) UNUSED AFTER
C   CT3           I              /INTG/      COUNTER (INOPT) UNUSED AFTER
C   CT4           I              /INTG/      COUNTER (INOPT)+PASSES THRU (DYN)
C   CT5           I              /INTG/      COUNTER (INOPT) UNUSED AFTER
C   DOMC          R   3,N+1      /REAL/      PART OF ANG. ACC. VEC. (RATE)
C   DUMMY         R   1000       /SATELL/    STORAGE AREA FOR USER
C   ETC           R   3,N+1      /REAL/      GYRO+EXT.TORQ.ON NEST (ETA,QFDOT)
C   ETIC          R   3,N        EQIV(XMN)   INERT X-COUP TORQ. (ETA)
C   ETM           R   33         /REAL/      SCALAR,GENEALIZED TORQUES (QFDOT)
C   ETMC          R   3,N        EQIV(XMN)   WHEEL X-COUP TORQ. (ETA)
C   FCF           R   3,3,4N     /REAL/      MODAL CENTRIP X-COUP(INOPT,QFDOT)
C   FCK           R   3,4N       /REAL/      MODAL CORIOLIS X-COUP(INOPT,QFDOT)
C   FCON          I   3(N+1)     /INTG/      CODE,FREE VECTORS (INBS)
C   FG1           L              /LOGIC/     END OF RUN FLAG (MAIN,DYN,OUTPSP)
C   FG2           L              /LOGIC/     ERROR INPUT DATA(MAIN,INEROR)
C   FG3           L              /LOGIC/     ERROR INPUT DATA(MAIN,INOPT)
C   FG4           L              /LOGIC/     UNUSED
C   FG5           L              /LOGIC/     OUTPUT DATA ? FLAG (MAIN,TORQUE)
C   FLA           R   3,2N       /REAL/      MODAL CM VECTOR BFC (INOPT)
C   FLAC          R   3,2N       /REAL/      MODAL CM VECTOR CFC (VDIV,TRANVD)
C   FLB           R   3,2N       /REAL/      MODAL MOMENT VECTOR BFC (INOPT)
C   FLC           R   3,2N       /REAL/      MODAL ROTATION MOMENT BFC (INOPT)
C   FLCRC         R   3,N        /REAL/      GYRO FLEXIBLITY FORCE (ETA)
C   FLD           R   3,3,2N     /REAL/      MODAL INERTIA DYAD BFC (INOPT)
C   FLE           R   3,3,2N     EQIV(FLD)   FLD + FLD**T (VDIV)
C   FLH           R   3,3,2N     EQIV(FLJ)   FLD + FLH (VDIV)
C   FLIRC         R   3,N        /REAL/      GYRO FLEXIBLITY TORQUE (ETA)
C   FLJ           R   3,3,2N     /REAL/      MODAL ROTATION DYAD BFC (INOPT)
C   FLQ           R   3,2N       EQIV(FLB)   MODAL MOMENTUM VECTOR BFC (VDIV)
C   FLQC          R   3,2N       /REAL/      FLQ IN CFC (VDIV,TRANVD)
C   FLOM          R   2N         /REAL/      MODAL FREQUENCY (INOPT)
C   FOMC          R   3,N+1      /REAL/      INERTIAL RATE VECTOR (RATE)
C   GAM           R   3,IDEM2    /REAL/      HINGE TO CM VECTOR (VDIV,XDY)
C   H             R              /REAL/      INTEGRATION STEP SIZE (RUNGE,INBS)
C   HM            R   3,N        /REAL/      WHEEL SPIN AXIS BFC (INBS)
C   HMC           R   3,N        /REAL/      WHEEL SPIN AXIS CFC (VDIV,TRANVD)
C   HMOM          R   N          /REAL/      WHEEL ANGULAR MOMENTUM(INBS,SETUP)
```

Table 1 (Continued)
Symbol Names Used in N-BOD2

C	IDEM4	I		LOCAL	SIZE OF /LOGIC/ (RSTART)
C	IDEM5	I		LOCAL	SIZE OF /INTG/ (RSTART)
C	IDEM6	I		LOCAL	SIZE OF /INTGZ/ (RSTART)
C	IDEM7	I		LOCAL	SIZE OF /REAL/ (RSTART)
C	IDEM8	I		LOCAL	SIZE OF /REALZ/ (RSTART)
C	IDEM9	I		LOCAL	SIZE OF /SATELL/(RSTART)
C	IINIT	I	IDEM5	EQIV(AWORK)	ZERO OUT /INTG/ (RSTART)
C	INERF	L		/LOGIC/	FLAG,BFC OR IFC FOR CFC (INOPT)
C	IZINIT	I	IDEM6	EQIV(SCNDUM)	ZERO OUT /INTGZ/ (RSTART)
C	JCON	I	N	/INTG/	BODY CONNECTION MATRIX (INBS)
C	LCON	I	2(N+1)	/INTG/	CODE,LOCKED VECTORS (INBS)
C	LANGLE	L		/LDEBUG/	PRINT EQUATIONS IN ANGLE? (MAIN)
C	LDCT	L		/LDEBUG/	PRINT EQUATIONS IN DCT? (MAIN)
C	LEQU	L		EQIV( )	EQIV IN EACH SUB TO PRINT FLAG
C	LEQUIV	L		/LDEBUG/	PRINT EQUATIONS IN EQIV? (MAIN)
C	LETA	L		/LDEBUG/	PRINT EQUATIONS IN ETA? (MAIN)
C	LINIT	L	IDEM4	EQIV(FG1)	ZERO OUT /LOGIC/ (RSTART)
C	LQFDOT	L		/LDEBUG/	PRINT EQUATIONS IN QFDOT? (MAIN)
C	LRATE	L		/LDEBUG/	PRINT EQUATIONS IN RATE? (MAIN)
C	LRTAPE	L		/CHEKS/	CREATE RESTART TAPE? (MAIN)
C	LRUNGE	L		/LDEBUG/	PRINT EQUATIONS IN RUNGE? (MAIN)
C	LSETUP	L		/LDEBUG/	PRINT EQUATIONS IN SETUP? (MAIN)
C	LSIMQ	L		/LDEBUG/	PRINT EQUATIONS IN SIMQ? (MAIN)
C	LTORQU	L		/LDEBUG/	PRINT EQUATIONS IN TORQUE? (MAIN)
C	LTRAN	L		/LDEBUG/	PRINT EQUATIONS IN TRAN? (MAIN)
C	LTRANV	L		/LDEBUG/	PRINT EQUATIONS IN TRANVD? (MAIN)
C	LTRNSI	L		/LDEBUG/	PRINT EQUATIONS IN TRNSIV? (MAIN)
C	LVDIV	L		/LDEBUG/	PRINT EQUATIONS IN VDIV? (MAIN)
C	LXDY	L		/LDEBUG/	PRINT EQUATIONS IN XDY? (MAIN)
C	MO	I	N	/INTG/	BODY IN WHICH WHEEL IS IN (INBS)
C	NBOD	I		/INTG/	NUMBER OF BODIES (INBS)
C	NB1	I		/INTG/	NUMBER OF BODIES + 1 (INBS)
C	NCTC	I		/INTG/	(INOPT,NA) CONSTRAINT TORQUES
C	NEQ	I		LOCAL	# EQUATIONS SETUP BY N-BOD2 (EQIV)
C	NFER	I		/INTG/	# FREE COORD VECTORS (INBS)
C	NFKC	I		/INTG/	(INOPT,NA) CONSTRAINT FORCES
C	NFLXB	I		/INTG/	# FLEXIBLE BODIES (INOPT)
C	NFRC	I		/INTG/	# RELATIVE ANGLES COMPUTED (INOPT)
C	NLOR	I		/INTG/	# LOCKED COORD VECTORS(INBS)
C	NMO	I		/INTG/	TOTAL NUMBER OF WHEELS (INBS)
C	NMOA	I		/INTG/	# WHEELS TO COMP REL ANGLE (INOPT)
C	NMODS	I		/INTG/	TOTAL # MODES FOR SYSTEM (INOPT)
C	NMV	I		/INTG/	# VARIABLE SPEED WHEELS (INOPT)
C	NSTART	L		/CHEKS/	NEW OR RESTART RUN? (MAIN)
C	NSVP	I		/INTG/	# LOCKED VECTORS TRANSFORM(VDIV)

Table 1 (Continued)
Symbol Names Used in N-BOD2

C	NSVQ	I		/INTG/	# FREE VECTORS TRANSFORM (VDIV)
C	NTQ	I		/INTG/	# DIFF EQS.IN SUB TORQUE (TORQUE)
C	PCON	I	N+1	/INTG/	# CONSTRAINED AXES AT HINGES(INBS)
C	PHI	R	3,N+1	/REAL/	EXTERNAL TORQUE ON NEST (TORQUE)
C	PLM	R	N	/REAL/	WHEEL SPIN INERTIA (INBS)
C	QF	R	3,3(N+1)	/REAL/	FREE VECTOR BFC (INBS)
C	QFC	R	3,3(N+1)	/REAL/	FREE VECTOR CFC (VDIV,TRANVD)
C	QL	R	3,2(N+1)	/REAL/	LOCKED VECTOR BFC (INBS)
C	QLC	R	3,2(N+1)	/REAL/	LOCKED VECTOR CFC (VDIV,TRANVD)
C	RBLO	L	N	/LOGIC/	RIGID BODY OR POINT MASS? (INBS)
C	RINIT	R	IDEM7	EQIV(CA)	ZERO OUT /REAL/ (RSTART)
C	ROMC	R	3,N+1	/REAL/	RELATIVE RATE VECTOR (RATE)
C	RZINIT	R	IDEM8	EQIV(CBDUM)	ZERO OUT /REALZ/ (RSTART)
C	SC	I	3(N+1)	/INTG/	FREE VECTORS CAGED (INOPT,UNCAGE)
C	SCC	I	N	/INTG/	UNUSED
C	SCG	I		/INTG/	# CAGED DEGREES (INOPT,UNCAGE)
C	SCNDUM,SCN	I	0:N-1	/INTGZ/	CODE,CENTRIPETAL EFFECTS (INOPT)
C	SCRDUM,SCR	I	0:N-1	/INTGZ/	CODE,CORIOLIS EFFECTS (INOPT)
C	SCXC	I	2N	EQIV(TORQ)	CODE,X-COUP MODES (INOPT,QFDOT)
C	SD	I		/INTG/	CODE,DIRECTION COSINES (INOPT)
C	SEU	I		/INTG/	CODE,EULER ANGLES (INOPT)
C	SFCC	I		/INTG/	CODE,BODIES FLEX X-COUPLING (INOPT
C	SFKDUM,SFK	I	0:N-1	/INTGZ/	CODE,CONSTRAINT FORCE(INOPT,NA)
C	SFLX	I		/INTG/	CODE,ALL FLEXIBLE BODIES (INOPT)
C	SFR	I	3(N+1)	/INTG/	CODE,COMPUTE FREE VEC ANGLE(INOPT)
C	SFXM	I	N	/INTG/	# MODES EACH BODY (INOPT)
C	SG	I		/INTG/	CODE,ALL GYROSTATS(SETS)
C	SI	I	IDEM3	/INTG/	CODE,BODIES HINGE TO CM (SETS)
C	SIG	I		/INTG/	UNUSED
C	SIXDUM,SIX	I	0:N-1	/INTGZ/	CODE,INERTIA EFFECTS (INOPT)
C	SKDUM,SK	I	0:N-1	/INTGZ/	CODE,BODIES IN EACH NEST (SETS)
C	SL	I		/INTG/	CODE,ALL POINT MASSES (SETS)
C	SLK	I	3(N+1)	/INTG/	CODE,CONSTRAINT TORQUE(INOPT,NA)
C	SMA	I	N	/INTG/	CODE,WHEEL ANGLE COMPUTE(INOPT)
C	SMAL	I		/INTG/	CODE,SMALL ANGLES (INOPT)
C	SMCDUM,SMC	I	0:N-1	/INTGZ/	CODE,ALL WHEELS IN NEST (INOPT)
C	SMV	I		/INTG/	CODE,VARIABLE SPEED WHEELS (INOPT)
C	SOK	I	N+1	/INTG/	CODE,BODIES HINGE O - CM (VDIV)
C	SPIDUM,SPI	I	0:N-1	/INTGZ/	CODE,PSUEDO INERTIA TENSORS(INOPT)
C	SQF	I	N+1	/INTG/	CODE,FREE VECTOR AT HINGE (SETS)
C	SQL	I	N+1	/INTG/	CODE,LOCKED VECTOR AT HINGE (SETS)
C	SR	I		/INTG/	CODE,ALL RIGID BODIES (SETS)
C	SSCN	I		/INTG/	CODE,UNION OF ALL SCN (VDIV)
C	SSIX	I		/INTG/	CODE,UNION OF ALL SIX (VDIV)
C	SVA	I		/INTG/	CODE,CM VECTORS TRANSFORM (VDIV)

Table 1 (Continued)
Symbol Names Used in N-BOD2

```
C   SVB             I                   /INTG/      CODE,HINGE VECTORS TRANSFORM(VDIV)
C   SVD             I                   /INTG/      CODE,DON'T TRANSFORM (INOPT)
C   SVI             I                   /INTG/      CODE,INERTIA DYAD TRANSFORM (VDIV)
C   SVM             I                   /INTG/      CODE,SPIN VECTORS TRANSFORM(VDIV)
C   SVP             I   2(N+1)          /INTG/      CODE,LOCKED VECTORS TRANSFORM(VDIV
C   SVQ             I   3(N+1)          /INTG/      CODE,FREE VECTORS TRANFORM(VDIV)
C   SXM             I   3,N             /INTG/      CODE,SMALL ANGLE KINEMATICS(INOPT)
C   SXT             I                   /INTG/      CODE,TIME VARY COL INER MAT(INOPT)
C   T               R                   /REAL/      TIME (MAIN)
C   TEM             R   2,160           LOCAL       TEMP STORAGE AREA (RUNGE)
C   THA             R   33              /REAL/      GENEALIZED COORDINATES(INBS,SETUP)
C   THAD            R   33              /REAL/      GENERALIZE COORD RATE (INBS,SETUP)
C   THADD           R   33              EQIV(ETM)   GENERALIZE COORD ACC (SETUP,SIMQ)
C   THADW           R   N               /REAL/      WHEEL RATE (INBS,SETUP)
C   THAW            R   N               /REAL/      WHEEL ANGLE (INBS,SETUP)
C   TIMEND          R                   /REAL/      TIME TO END RUN (INBS,DYN)
C   TORQ            I   97              /INTG/      UNUSED STORAGE AREA FOR USER
C   TUG             R   3(N+1)          /REAL/      TIME TO UNCAGE (INOPT,UNCAGE)
C   XDIC            R   3,3,IDEM2       /REAL/      MATRIX OF INERTIA TENSORS(VDIV,XDY
C   XI              R   3,3,N           /REAL/      INERTIA DYAD BFC (INBS)
C   XIC             R   3,3,N           /REAL/      INERTIA DYAD CFC (VDIV,TRANVD)
C   XIO             R   3,3,N           /REAL/      ZERO INERTIA DYAD BFC(VDIV)
C   XMAS            R   N               /REAL/      BODY MASS (INBS)
C   XMCDUM,XMC      R   3,3,0:N         /REALZ/     TRANSFORM BFC TO CFC (TRNSIV,TRAN)
C   XMN             R   33,33           /REAL/      SCALAR INERTIA MATRIX (VDIV,QFDOT)
C   Y               R   160             LOCAL       SYSTEM STATE (EQIV,SETUP,TORQUE)
C   YD              R   160             LOCAL       SYSTEM STATE DERIV (SETUP,TORQUE)
C   YMCD            R   3,2,N+1         EQIV(XMN)   DIRECTION COSINE RATES (DCT)
C   XMT             R   3,3,N           /REAL/      ZERO STATE TRANSFORMTION MAT(INBS)
C   ZETA            R   2N              /REAL/      MODAL DAMPING RATIO (INOPT)
C
C
```

FG2 – input error flag. If .TRUE., the input data defines a physically realizable system. If .FALSE., a criteria for physical realizability in subroutine INEROR has not been satisfied and the job will terminate with an error message.

FG3 – input error flag. If .FALSE., a code word on an option card read by subroutine INOPT has not been recognized and the job will terminate with an error message.

FG5 – output data flag. Its status is checked at the end of each integration step. If .TRUE., a call to the data output subroutine OUTPSP is made. If .FALSE., it is not. To change the frequency of data output, the user may change the status of FG5 in subroutine TORQUE. The default status of FG5 is .TRUE..

LANGLE –
LDCT –
LEQUIV –
LETA –
LQFDOT –
LRATE –
LRTAPE –
LRUNGE –
LSETUP –
LSIMQ –
LTORQU –
LTRAN –
LTRANV –
LTRNSI –
LVDIV –
LXDY –
NSTART –

> see section "Data Input, Input/Output Control."

T – independent variable, defines actual simulation time. Evaluated in subroutine RUNGE.

TIMEND – termination time. At T.GE.TIMEND, the flag FG1 is reset to .FALSE. in subroutine DYN.

For additional discussion, see comment cards inserted into the MAIN program and subroutine DYN.

DATA INPUT, INPUT/OUTPUT CONTROL

The first card of the data deck is called from the MAIN program by the statement:

```
READ 102, NSTART, LRUNGE, LTRNSI, LVDIV, LEQUIV, LTRAN, LTRANV, LRATE, LXDY
     LETA, LTORQU, LQFDOT, LDCT, LANGLE, LSETUP, LSIMQ, LRTAPE
```

```
102 FORMAT (4X, 17L1)
```

The data on the card set 17 logic flags that are used to control the mode of input and output. The following definitions apply to the logic variables set by the data on this card:

The first logic variable controls the mode of data input.

NSTART = .FALSE. A new job: data input will be from cards. The starting time for the simulation run will be time zero (T.EQ.0).

= .TRUE. A restart job: data input will be from tape. The input tape for a restart job is the output tape of a previous job. It is generated by N-BOD2. The starting time for the simulation will be the termination time of the job for which the restart tape was written.

The next 15 logic variables control the print commands programmed in the various subroutines. For example:

LRUNGE = .FALSE. All print statements coded in subroutines RUNGE will be bypassed.

= .TRUE. All equations coded in subroutine RUNGE will be printed along with the numerical magnitudes of the associated variables at the time of execution.

Table 2 lists the subroutines in which the system equations of motion are coded and the associated logic variables that control whether or not the equations coded therein should be outputted on the line printer.

The last logic variable of card 1 controls whether or not a restart tape should be written.

LRTAPE = .FALSE. Create a restart tape at the termination time of this job.

= .TRUE. Do not create a restart tape at the termination time.

If a restart tape is created, the restart data are put into file 2 of the output tape referred to as data set 11 by the WRITE statements.

DATA INPUT, SUBROUTINE RSTART

After the first data card is read, a CALL to subroutine RSTART is made. The precautionary step of zeroing all storage locations, except those already loaded by the data on the first input

Table 2

Logic Variables Controlling the Printing
of the Equations of Motion Coded in the Subroutines

Logic Variable	Subroutine
LRUNGE	RUNGE
LTRNSI	TRNSIV
LVDIV	VDIV
LEQUIV	EQIV
LTRAN	TRAN
LTRANV	TRANVD
LRATE	RATE
LXDY	XDY
LETA	ETA
LTORQU	TORQUE
LQFDOT	QFDOT
LDCT	DCT
LANGLE	ANGLE
LSETUP	SETUP
LSIMQ	SIMQ

card, is performed upon entry. A check on the status of the logic variable NSTART is then made:

IF (NSTART .EQ. FALSE),

a return to the main program is immediately made. The data to be read in will be from cards and will define a new job.

IF (NSTART .EQ. TRUE),

the data to restart an old job will be read according to the read statements coded in subroutine RSTART. The restart tape is read according to the following statements:

```
READ (10, 102) Y
READ (10, 102) YD
READ (10, 101) NEQ
READ (10, 102) (DUMMY (I), I = 1, IDEM9)
READ (10, 102) (RINIT (I), I = 1, IDEM7)
```

```
      READ (10, 102) (RZINIT (I), I = 1, IDEM8)
      READ (10, 101) (IINIT (I), I = 1, IDEM5)
      READ (10, 101) (IZINIT (I), I = 1, IDEM6)
      READ (10, 101) (LINIT (I), I = 1, IDEM4)
101   FORMAT (16 Z 8)
102   FORMAT (8 Z 16)
```

These statements read the restart tape exactly as it was written. The user has no control over the writing or reading of the restart tape other than by the logic control variable LRTAPE that defines whether or not it should be created at the time of run termination.

At the end of any job that calls for a restart tape to be generated, the restart data are put in file 2 of the output tape which is data set 11. To execute a restart job, the program assumes that the appropriate job control statements have been provided that will position the restart tape (now data set 10) at the start of file 2 before the execution of the first READ input tape instruction. (See the comment cards in subroutine RSTART and the job control requirements discussion in the following section.)

After the restart tape has been read, the last input card required for a restart run is read by the statement:

```
      READ 103, TIMEND
103   FORMAT (D15.5)
```

in RSTART where

TIMEND = termination time for present job.

This read statement is executed only on a restart job. Upon exit from RSTART, return is made to the start of the numerical integration loop in the MAIN program. The job then proceeds as if it had never been previously terminated. There are no more data input cards required for a restart job.

PROGRAMMER'S GUIDE TO SUBROUTINE RSTART

There is no actual computation carried out in subroutine RSTART. Its functions are:

- Zero all storage locations used by N-BOD2 before any computation is done.
- Read the restart tape so that computation may be reinitiated at the termination time of a previous run.
- Write the restart tape so that computation may be restarted at a later data.

For further detail, see the comment cards inserted in the subroutine RSTART.

To zero out all common blocks, the size of each must be defined. Thus:

IDEM4 = number of logical words in /LOGIC/

IDEM5 = number of integer words in /INTG/

IDEM6 = number of integer words in /INTGZ/

IDEM7 = number of real words in /REAL/

IDEM8 = number of real words in /REALZ/

IDEM9 = number of real words in /DUMMY/

These parameters are defined in subroutine RSTART and must be changed if any of the primary labeled common blocks are extended or modified. The user is referred to the program listing to note the method used to zero the common blocks.

To make use of the restart capability, the user is required to provide appropriate machine-dependent job control statements.

N-BOD2 is programmed to put all user-desired output data in file 1 of data set 11 and to put all of the restart data in file 2 of data set 11. Data set 11 becomes the restart tape.

To restart the job, the restart tape becomes data set 10. The appropriate job control statements must be provided to position it at the start of file 2 prior to the execution of the first read tape instruction of the program.

For a restart job, the author has found it convenient to provide the job control statements that copy, prior to job execution, file 1 of data set 10 onto file 1 of data set 11. Data set 10 will then be positioned at the start of file 2 and, at the end of the restart job, file 1 of data set 11 will contain a continuous record of the output data from time zero. Furthermore, the old restart tape, data set 10, will not have been destroyed.

DATA INPUT, SUBROUTINE INBS

IF (NSTART .EQ. FALSE)*,

a call to the subroutine INBS is made immediately after the program returns to the MAIN program from RSTART. In INBS, the input data required to describe the basic simulation model are read in from cards.

As in References 1 and 2, the labeling of a system of N bodies (rigid bodies + flexible bodies + point masses) and M symmetric wheels must adhere to the following rules:

- Bodies must be given distinct integer labels ranging from 1 to N inclusive. The labeling must be such that, along any topological path extending from body 1 to a limb end, the body labels are of increasing magnitude.

*Condition code for a new job.

- Symmetric wheels must be given distinct integer labels ranging from 1 to M inclusive. These may be randomly assigned.
- Body 1 is treated as the principal body of the system.

The labeling of hinge points between contiguous bodies is done by the computer according to the following rule:

- Hinge point K-1 is the point of connection between body K and the chain of bodies extending to it from body 1.

Figures 3 and 4 are provided as visualization aids. Figure 3 is the example used in Reference 1 to illustrate the defined labeling convention. Figure 4 shows the exact position and labeling of each of the position vectors defined for the system.

The following definitions are used in the coding of subroutine INBS. The total number of bodies and wheels that make up the simulation model are defined by:

NBOD – total number of rigid bodies, flexible bodies, and point masses

NMO – total number of symmetric wheels

also defined is the constant

NB1 = NBOD + 1

As a direct consequence of the established labeling convention, the system topology can be uniquely defined by the one dimensional arrays:

JCON(K) for K = 2,3, . . . , NBOD	Body label of the body connected to body K at hinge point K-1. Relative to body 1, it is the body inboard of body K.
JCON(1) = 0	implies body 1 is contiguous to the inertial frame of reference.
MO(M) for M = 1,2, . . . , NMO	Body label of the body in which symmetric wheel M is is imbedded.

The physical nature of each body, its mass and the number of kinematic constraints between contiguous bodies, are defined by:

RBLO(K) = .TRUE.	Body K is either a rigid or a flexible body. For N-BOD2, body 1 and all bodies not at a limb end must be rigid.
= .FALSE.	Body K is a point mass.
for K = 1,2, . . . , NBOD	
XMAS(K) for K = 1,2, . . . , NBOD	Total mass of body K including mass of all symmetric wheels which are imbedded within it.

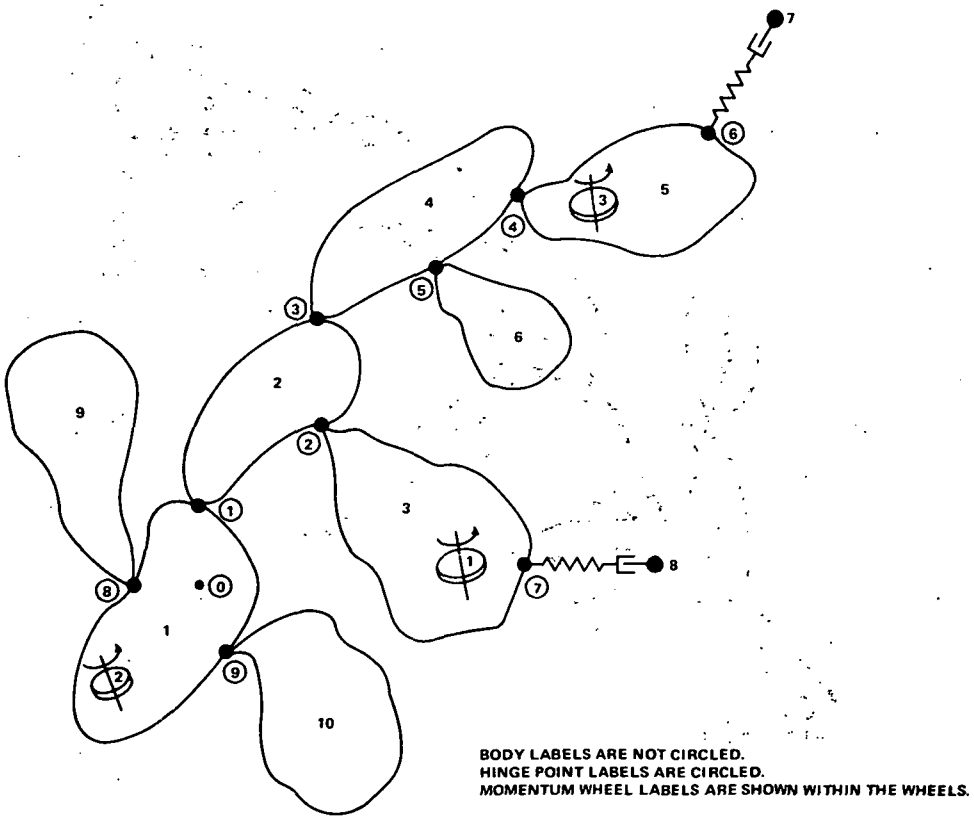


Figure 3. Labeling scheme for 10-body example.

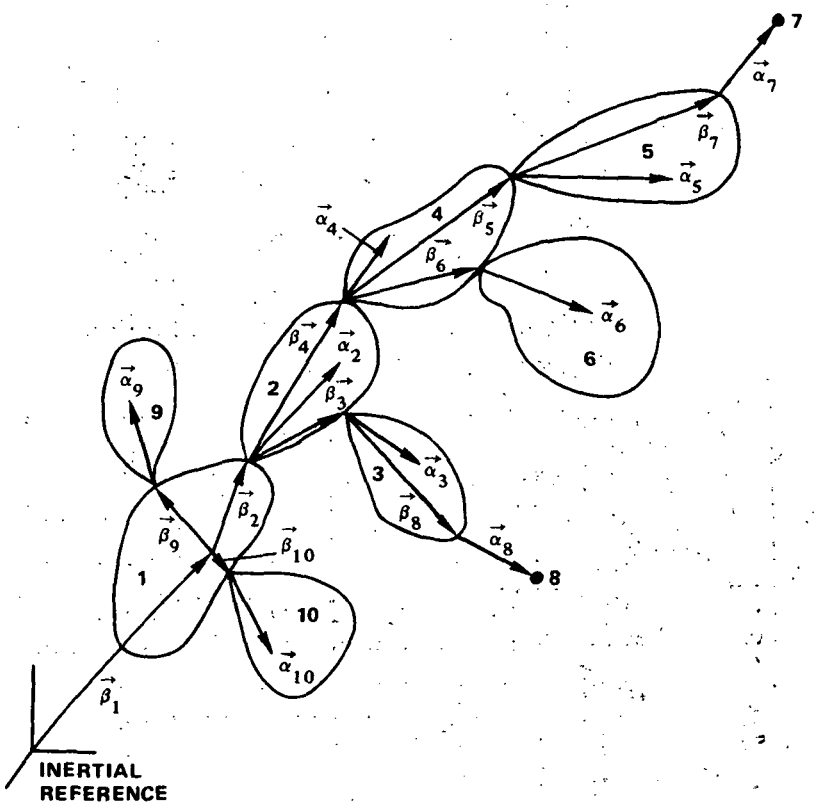


Figure 4. Positions and labeling of hinge-point and center-of-mass location vectors.

PCON(K)

for K = 1	Number of kinematically constrained rotational degrees of freedom between body 1 and an inertially fixed frame of reference.
for K = 2,3, . . . , NBOD	Number of kinematically constrained rotational or translational degrees of freedom between body K and body JCON(K).
for K = NBOD+1	Number of kinematically constrained translational degrees of freedom between body 1 and the inertially fixed frame of reference.

$$\text{PCON(K) for K} = 1,2,\ldots,\text{NBOD+1} = \begin{cases} 0 & \text{implies 3 degrees of relative freedom} \\ 1 & \text{implies 2 degrees of relative freedom} \\ 2 & \text{implies 1 degree of relative freedom} \\ 3 & \text{implies no relative motion allowed} \end{cases}$$

The equations of motion have been derived in a vector-dyadic format and are valid for all frames of reference. To numerically solve these equations, it must be possible to express any vector or tensor relative to any desirable frame of reference.

The equations of motion are derived relative to an origin that is fixed in an inertially fixed-reference system. The user is required to pick an inertial origin and to define an orthogonal reference frame there. All translational and rotational motion of body 1 is computed relative to this inertially fixed-reference frame.

Let

(X_0, Y_0, Z_0) = coordinate axes of the inertially fixed-reference frame

N-BOD2 *assumes* that if:

PCON (NBOD + 1) = 0	Body 1 unconstrained in translation
PCON (NBOD + 1) = 1	Body 1 constrained to translate in the X_0, Y_0 plane
PCON (NBOD + 1) = 2	Body 1 constrained to translate in the X_0 direction
PCON (NBOD + 1) = 3	Body 1 totally constrained in translation

In addition to the inertial reference frame, body fixed-reference frames must also be defined. Consequently, a body K fixed-reference frame is defined at hinge point K-1; K = 1, 2, . . . NBOD. The orientation of the body K fixed-coordinate axes is arbitrary and to be defined by the user. To simplify the input of initial kinematic conditions, it is desirable to define a zero state for the system. All relative motion between contiguous bodies can then be measured relative to this state in terms of a few physically meaningful relative displacement parameters.

The following set of transformation matrices completely defines the relative orientation of all reference frames when the system is in the zero relative displacement state.

[XMT (I, J, K)] – transformation matrix used to transform vectors from body K to body JCON(K) fixed coordinates when the system is in its zero relative displacement state

$$[XMT(I,J,K)] = \begin{bmatrix} XMT(1,1,K) & XMT(1,2,K) & XMT(1,3,K) \\ XMT(2,1,K) & XMT(2,2,K) & XMT(2,3,K) \\ XMT(3,1,K) & XMT(3,2,K) & XMT(3,3,K) \end{bmatrix}$$

where

$$K = 1, 2, \ldots, NBOD$$

If K = 1, the inputted transformation matrix transforms vectors from body 1 fixed coordinates to inertial fixed coordinates.

The origin of the body K fixed-reference frame is always at hinge point K-1; K = 1, 2, . . . , NBOD. If body K, K = 2, 3, . . . , NBOD, is a point mass, a transformation matrix is not inputted. The axes of the body fixed-reference frames are defined by the computer to be respectively parallel to the axes of the body JCON(K) fixed-reference frame.

The components of all inertia tensors and position vectors required to describe the physical size and rotational inertia characteristics of the simulation model are inputted relative to the body fixed-reference frame in which they are fixed. That is:

[XI (I, J, K)] for K = 1, 2, . . . , NBOD – components of the inertia tensor for body K and all despun wheels contained therein about the body's center of mass, relative to the body K fixed-reference frame (assumed to be zero if body K is a point mass).

$$[XI(I,J,K)] = \begin{bmatrix} XI(1,1,K) & XI(1,2,K) & XI(1,3,K) \\ XI(2,1,K) & XI(2,2,K) & XI(2,3,K) \\ XI(3,1,K) & XI(3,2,K) & XI(3,3,K) \end{bmatrix}$$

CA (I, K) for K = 1, 2, . . . , NBOD – components of the vector from hinge point K-1 to the center of mass of body K (zero state location if body K is a point mass), relative to the body K fixed-reference frame.

$$\{CA(I,K)\} = \begin{Bmatrix} CA(1,K) \\ CA(2,K) \\ CA(3,K) \end{Bmatrix}$$

CB (I, K)

for K = 0 – reserved for the components of the vector from the inertial origin to the composite system center of mass. (Not needed for computation and hence not computed, may be user-defined if desirable.)

for K = 1, 2, . . ., NBOD – components of the vector from hinge point JCON(K)-1 to hinge point K-1, (from inertial origin if K = 1), relative to the body JCON(K) fixed-reference frame

$$\{CB(I,K)\} = \begin{Bmatrix} CB(1,K) \\ CB(2,K) \\ CB(3,K) \end{Bmatrix}$$

Note: Special provision has been made in the coding of N-BOD2 to accept the subscript 0 in certain arrays such as CB. Furthermore, to avoid numerical computation problems, it is advisable to avoid placing the inertial origin at a distance that is many orders of magnitude greater than the diameter of a sphere which could contain the entire simulation model.

The previously defined integer array PCON(K), K = 1, . . . , NBOD+1, specifies the number of kinematic constraints that exist between contiguous bodies and between body 1 and the inertial reference. The logical array RBLO(K), K = 1, 2, . . . , NBOD, can be used to determine whether relative motion between the bodies K and JCON(K) is rotational or translational. Body 1 may have both rotational and translational degrees of freedom relative to the inertial reference. Rigid and flexible bodies may have only relative rotational degrees of freedom while point masses may have only relative translational degrees of freedom. N-BOD2 *assumes* that relative rotational motion may be adequately modeled by either a 0-, 1-, 2-, or 3-degree-of-freedom gimbal, while relative translational motion may be either 0, 1, 2, or 3 dimensional. Both the directions about which, or along which, motion is free of kinematic constraints and the directions about which, or along which, motion is kinematically constrained must be defined.

These directions are defined by a set of exactly NBOD+1 triads of unit coordinate vectors. These coordinate vectors are defined to be either free coordinate vectors (implying that motion is free about or along the directions defined by them) or locked coordinate vectors (implying that motion is locked or kinematically constrained about or along the directions defined by them). The user will be required to define two of the three coordinate vectors; the third will be internally generated by a vector cross product based upon conditions of orthogonality.

The integer array PCON(K), K = 1, 2, . . . , NBOD+1, defines the mix of free and locked coordinate vectors in each of the NBOD+1 triads. It also provides a convenient tool for creating a labeling sequence for the free and locked coordinate vectors. The labeling will be done by the computer.

It is a sequential numbering algorithm that starts with the coordinate vectors associated with relative rotational motion of body 1, proceeds sequentially up to body NBOD, and ends with the coordinate vectors associated with the description of the translational motion of body 1.

Let

$\{QF(I, M)\}$ = components of the unit free coordinate vector M relative to the reference frame in which it is fixed.

$\{QL(I, L)\}$ = components of the unit locked coordinate vector L relative to the reference frame in which it is fixed.

The mix of free and locked coordinate vectors at hinge point K-1 and their labeling sequence are defined as follows for K = 1, 2, . . . , NBOD+1:

- For PCON (K) = 0 the mix of coordinate vectors at hinge point K-1 is

$$\{QF(I, M)\}, \{QF(I, M+1)\}, \{QF(I, M+2)\}$$

- For PCON (K) = 1 the mix of coordinate vectors at hinge point K-1 is

$$\{QF(I, M)\}, \{QF(I, M+1)\}, \{QL(I, L)\}.$$

- For PCON (K) = 2 the mix of coordinate vectors at hinge point K-1 is

$$\{QF(I, M)\}, \{QL(I, L)\}, \{QL(I, L+1)\}.$$

- For PCON (K) = 3 the mix of coordinate vectors at hinge point K-1 is

$$\{QL(I, L)\}, \{QL(I, L+1)\}, \{QL(I, L+2)\}$$

where for any K = 1, 2, . . ., NBOD + 1

$$M = 1 + \sum_{\substack{i=1 \\ K \neq 1}}^{K-1} (3 - PCON(i))$$

$$L = 1 + \sum_{\substack{i=1 \\ K \neq 1}}^{K-1} PCON(i)$$

One of several different kinematic conditions can exist at each hinge point. The user must adhere to the following rules when specifying the coordinate of the free and locked coordinate vectors in each of the NBOD+1 triads:

- If RBLO (K) = .TRUE. for any K = 1, 2, . . . , NBOD, the coordinate vectors are associated with gimbal axes. Table 3 defines for this case the coordinate frames in which the free and locked coordinate vectors are defined.

- If RBLO (K) = .FALSE. for any K = 2, 3, . . . , NBOD, the coordinate vectors define the directions along which relative translational motion of point mass K is free and along which it is constrained. The origin of this triad of coordinate vectors is the zero position of the point mass K that is defined by the components of the center of mass vector $\{CA(I, K)\}$. Table 4 defines the coordinate frame in which the free and locked coordinate vectors are defined.
- If K = NBOD+1, the coordinate vectors define the direction along which the center of mass of body 1 is free to translate. These directions are defined by the computer based upon the number of kinematic constraints defined by PCON(NBOD+1). Table 5 defines the coordinates of the unit coordinate vectors relative to the inertially fixed reference. These are set by the computer.

The initial state of the simulation model can now be defined by specification of initial displacements and rates about or along the free coordinate vectors whose direction defines the directions of positive displacement. Let

THA (M) = initial relative displacement about or along the free coordinate vector M

THAD (M) = initial relative rate about or along the free coordinate vector M

The user need not be concerned with defining initial values for transformation matrices since it is done by an internal computer computation.

Table 3
Coordinate Frames in Which Free and Locked
Coordinate Vectors are to be Defined for
RBLO (K) = .TRUE.; K = 1, 2, . . . , NBOD

Vector	PCON (K)			
	0	1	2	3
$\{QF(I, M)\}$	JCON (K)	JCON (K)	JCON (K)	
$\{QF(I, M+1)\}$	Intermediate	K		
$\{QF(I, M+2)\}$	K			
$\{QL(I, L)\}$		Intermediate	K	K
$\{QL(I, L+1)\}$			K	K
$\{QL(I, L+2)\}$				K

Table 4
Coordinate Frames in Which Free and Locked
Coordinate Vectors are to be Defined for
RBLO (K) = .FALSE.; K = 2, 3, . . . , NBOD

Vector	PCON (K)			
	0	1	2	3
{QF (1, M)}	K	K	K	
{QF (1, M + 1)}	K	K		
{QF (1, M + 2)}	K			
{QL (1, L)}		K	K	K
{QL (1, L + 1)}			K	K
{QL (1, L + 2)}				K

Table 5
Coordinates of the Unit Coordinate Vectors Defined
at the Inertial Origin Along Which Translational
Motion of Body 1 is Measured;
K = NBOD + 1

Vector	PCON (K)			
	0	1	2	3
{QF (I, M)}	⌊ 1, 0, 0 ⌋	⌊ 1, 0, 0 ⌋	⌊ 1, 0, 0 ⌋	
{QF (I, M + 1)}	⌊ 0, 1, 0 ⌋	⌊ 0, 1, 0 ⌋		
{QF (I, M + 2)}	⌊ 0, 0, 1 ⌋			
{QL (I, L)}		⌊ 0, 0, 1 ⌋	⌊ 0, 1, 0 ⌋	⌊ 1, 0, 0 ⌋
{QL (I, L + 1)}			⌊ 0, 0, 1 ⌋	⌊ 0, 1, 0 ⌋
{QL (I, L + 1)}				⌊ 0, 0, 1 ⌋

To define the characteristics of each of the symmetric wheels imbedded within the simulation model, the following definitions are made:

{HM (I, M)} – components of a unit vector along the spin axis of symmetric wheel M relative to the body MO(M) fixed-reference frame. Its direction defines the direction of positive rotation.

PLM (M) – rotational inertia of the symmetric wheel M about its spin axis.

THAW (M) – initial angular position of the wheel M about its spin axis.

THADW (M) – initial angular rate of the wheel M about its spin axis relative to body MO(M).

The integration step size for the fourth order Runge-Kutta integration package and the time for job termination are defined by:

H – integration step size

TIMEND – job termination time

To completely specify the initial state of the simulation model, the above parameters must be defined by the user or be internally generated. The following is a listing of the READ statements coded in subroutine INBS in order of execution. The data deck must be prepared accordingly.

```
********** READ STATEMENTS FOR SUBROUTINE INBS IN ORDER OF EXECUTION **********
C
C     INITIALIZE FREE AND LOCKED COORDINATE VECTOR LABELS
      M = 1
      L = 1
********** FIRST CARD READ BY INBS IS SECOND CARD OF DATA DECK
      READ 100, NBOD
C
********** READ EXACTLY NBOD SETS OF BODY DATA, ONE PER BODY
      DO 1 K=1,NBOD
      READ 105, N,(MESS(J),J=1,18)
      IF(N.NE.K) ERROR, CARD OUT OF SEQUENCE, PRINT MESSAGE, HALT
C     MESS(J),J=1,18  ALPANUMERIC DESCRIPTION OF BODY N, PRINTED WITH DATA ECHO
      READ 101, RRLO(K),JCON(K),PCON(K),XMAS(K)
      IF(RRLO(K)) GO TO 2  BODY K IS A RIGID OR FLEXIBLE BODY
      GO TO 8  BODY K IS A POINT MASS
    2 READ 102, ((XI(I,J,K),J=1,3),I=1,3)
      READ 102, ((XMT(I,J,K),J=1,3),I=1,3)
    8 READ 102, (CA(J,K),J=1,3)
      READ 102, (CB(J,K),J=1,3)
C     COMPUTE LABELS FOR FREE AND LOCKED COORDINATE VECTORS INTERNALLY
      IF(K.EQ.1) GO TO 12
      M = M + 3 - PCON(K-1)
      L = L + PCON(K-1)
   12 GO TO (13,14,15,17), PCON(K)+1
   13 CONTINUE     THREE DEGREES OF RELATIVE FREEDOM FOR BODY K
      READ 102, (QF(J,M),J=1,3)
      READ 102, (QF(J,M+2),J=1,3)
      READ 102, (THA(J),J=M,M+2)
      READ 102, (THAD(J),J=M,M+2)
      GO TO 1  END OF DATA FOR BODY K
   14 CONTINUE     TWO DEGREES OF RELATIVE FREEDOM FOR BODY K
      READ 102, (QF(J,M),J=1,3)
      READ 102, (QF(J,M+1),J=1,3)
      READ 102, (THA(J),J=M,M+1)
      READ 102, (THAD(J),J=M,M+1)
      GO TO 1  END OF DATA FOR BODY K
```

```
   15 CONTINUE      ONE DEGREE OF RELATIVE FREEDOM FOR BODY K
      READ 102, (QF(J,M),J=1,3)
      READ 102, (QL(J,L),J=1,3)
      READ 102, THA(M)
      READ 102, THAD(M)
      GO TO 1  END OF DATA FOR BODY K
   17 CONTINUE      ZERO DEGREES OF RELATIVE FREEDOM FOR BODY K
      READ 102, (QL(J,L),J=1,3)
      READ 102, (QL(J,L+1),J=1,3)
      GO TO 1  END OF DATA FOR BODY K
    1 CONTINUE
********** ALL BODY DESCRIPTIVE DATA IS IN MEMORY NOW
C
********** READ TRANSLATION CONDITIONS FOR BODY 1
      READ 100, PCON(NBOD+1)
      M = M + 3 - PCON(NBOD)
      READ 102, (THAD(J),J=M,M+2-PCON(NBOD+1))
C
********** READ ALL SYMMETRIC WHEEL DATA
      READ 100, NMO
      DO 16 I=1,NMO     GO TO 16 IF NMO.EQ.0
      READ 104, MO(I),(HM(J,I),J=1,3)
      READ 106, PLM(I),(MESS(J),J=1,16)
      READ 102, THAW(I),THADW(I)
   16 CONTINUE
********** ALL WHEEL DATA IS IN MEMORY
C
********** READ STEP SIZE AND TIME TO END SIMULATION
      READ 102, H,TIMEND
C     END OF DATA TO BE READ BY SUBROUTINE INBS
C
********** FORMAT STATEMENTS USED
  100 FORMAT (I5)
  101 FORMAT (L5,2I5,D15.5)
  102 FORMAT (3D15.5)
  104 FORMAT (I5,3D15.5)
  105 FORMAT (I5,18A4)
  106 FORMAT (D15.5,16A4)
```

PROGRAMMER'S GUIDE TO SUBROUTINE INBS

The primary function of subroutine INBS is to read the data cards that define the basic simulation model and its initial kinematic conditions. This has been thoroughly discussed in the section "Data Input, Subroutine INBS." A cursory glance at the subroutine listing will show that it is also programmed to provide an echo of the input data and to compute a few additional variables that will be needed for the succeeding computations.

The following variables defined in table 1 are either first introduced or used extensively in subroutine INBS. The programmer is referred to the comment cards inserted therein and the following for detailed definitions:

CA (I, K) – see section "Data Input, Subroutine INBS."

CB (I, K) – see section "Data Input, Subroutine INBS."

CBN (I) – computed variable. Used to compute body 1 center of mass location relative to inertial origin

$$\begin{Bmatrix} CBN(1) \\ CBN(2) \\ CBN(3) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad \text{if PCON (NB1) = 0}$$

$$= \begin{Bmatrix} 0 \\ 0 \\ CB(3,1) \end{Bmatrix} \quad \text{if PCON (NB1) = 1}$$

$$= \begin{Bmatrix} 0 \\ CB(2,1) \\ CB(3,1) \end{Bmatrix} \quad \text{if PCON (NB1) = 2}$$

$$= \begin{Bmatrix} CB(1,1) \\ CB(2,1) \\ CB(3,1) \end{Bmatrix} \quad \text{if PCON (NB1) = 3}$$

FCON (M) – code used to define the body label of the body in which free coordinate vector M is fixed.

FCON (M) = K — if free coordinate vector M is fixed in body K

FCON (M) = -(M-1) — if free coordinate vector M is the intermediate axis of a three axis gimbal

H – see section "Data Input, Subroutine INBS."

{HM (I, M)} – see section "Data Input, Subroutine INBS."

HMOM (M) – relative angular momentum of momentum wheel M

HMOM (M) = PLM (M) * THADW (M)

JCON (K) – see section "Data Input, Subroutine INBS."

LCON (L) – code used to define the body label of the body in which locked coordinate ordinate vector L is fixed

LCON (L) = K — if locked coordinate vector L is fixed in body K

= -M — if locked coordinate vector L is the axis about which relative motion is kinematically constrained on a two axis gimbal defined by the free vector M and M+1

MO (M) –
NBOD –
NB1 – } see section "Data Input, Subroutine INBS."

NFER – total number of free coordinate vectors

NLOR – total number of locked coordinate vectors

NMO –
PCON (K) –
PLM (M) – } see section "Data Input, Subroutine INBS."

{QF (I, M)} –
{QL (I, L)} –
RBLO (K) –
THA (M) –
THAD (M) –
THAW (M) –
THADW (M) –
TIMEND –
[XI (I, J, K)] –
XMAS (K) –
[XMT (I, J, K)] –

} see section "Data Input, Subroutine INBS"

PROGRAMMER'S GUIDE TO SUBROUTINE INEROR

The main function of subroutine INEROR is to check the data inputted by subroutine INBS for obvious errors.

The following error checks are made:

- All bodies must have positive mass greater than zero.
- The topological tree must be properly labeled; i.e., JCON (K) must be less than K for all K where K = 1, 2, . . . , NBOD.
- The number of kinematic constraints must be 0, 1, 2, or 3.
- Inertia tensors must be symmetric and have a positive nonzero determinant.
- Zero-state transformation matrices must be orthonormal.

If any of the above tests are not satisfied, the flag FG2 is set .FALSE.. Appropriate error messages are printed to assist the user in locating the error source, and the job will terminate after a return to the MAIN program.

PROGRAMMER'S GUIDE TO SUBROUTINE SETS

The main function of subroutine SETS is to interrogate the data inputted by subroutine INBS and compute various sets of integers which will enable the computer to avoid redundant and trivial computation. To save computer storage, code words are created that compact the desired integer arrays into a single integer code word. Triangular arrays are stored as one-dimensional arrays in the memory of the computer. The following variables, defined

in table 1, are either first introduced or used extensively in subroutine SETS. The programmer is referred to the comment cards inserted therein and the following for detailed definitions.

SG — code word used to define the body labels of all gyrostats; i.e., all bodies that have one or more imbedded symmetric wheels.

SI (KN) — one dimensional array of code words used to define the body labels of all bodies on the topological path extending from hinge point K-1 outward to the center of mass of body N, where

$$KN = KT0(NBOD, K-1, N)$$

$$K = 1, 2, \ldots, NBOD$$

$$N \geqslant K$$

The utility function KT0 is used to store the triangular array of code words in a one-dimensional array.

SK (K-1) — code word used to define the body labels of those bodies contained in the nest of bodies K-1; K = 1, 2, . . . , NBOD. Due to the setup of labeled common block /INTGZ/, SK(0) is stored in the location SKDUM.

SL — code word used to define the body labels of all point masses.

SQF (K) — lowest magnitude free coordinate vector label at hinge point K-1; K = 1, 2, . . . , NBOD+1, where

$$K = NBOD+1 \qquad \text{implies the inertial origin}$$

and

$$SQF(K) = 0 \qquad \text{if } PCON(K) = 3$$

SQL (K) — lowest magnitude locked coordinate vector label at hinge point K-1; K = 1, 2, . . . , NBOD+1, where

$$K = NBOD+1 \qquad \text{implies the inertial origin}$$

and

$$SQL(K) = 0 \qquad \text{if } PCON(K) = 0$$

SR — code word used to define the body labels of all rigid and flexible bodies.

DATA INPUT, SUBROUTINE INOPT

Subroutine INOPT is the second input subroutine called by the MAIN program. The input cards to be read by it follow immediately after those read by subroutine INBS in the data deck.

The program N-BOD2 has been coded to accept a description of the basic simulation model by subroutine INBS. Amplifications or modifications to the model or to the equations of motion that describe it are defined by subroutine INOPT.

In many problems of practical interest, the user will not have to exercise any of the available options. In such cases all parameters required for computation, which could have been user defined, are set by default to the values that, in the author's opinion, are most commonly desired. If no options are to be used, INOPT will read only one data card.

INOPT is programmed to recognize different code words that are used to define which of the available options is to be used. The code words are punched on what will be referred to as "option cards." All option cards are read at program point 12 in INOPT by the statement

```
12 READ 100, ICD1, ICD2, ICD3, I, NSET, (S1 (J), J=1, 10)
100 FORMAT (3A4, I3, 1115)
```

Columns 1 through 12 of each option card contain the code word. Table 6 contains a listing of all code words, the columns in which the alphanumeric characters must be punched, and a brief description of the function of the option.

Upon recognition of a code word, the program branches to the appropriate point in the program where all required computational and additional READ data operations are carried out. Upon completion, the program branches back to program point 12 to read the next option card. The cycling back to program point 12 continues until an END OPTIONS option card is recognized. The END OPTIONS card is the last card to be read by INOPT.

If a code word is not recognized, the flag FG3 will be set .FALSE., an error message will be printed, and return to the MAIN program will be immediately made.

The following subsections define in detail the various option codes recognized by INOPT and their purpose.

***** Terminate Option Input *****

1. No Option Required

If no options are required for the simulation model, the code word NO OPTIONS must be punched on the data card immediately following the data cards read by subroutine INBS. In this case, all modifiable parameters are set by default. The NO OPTIONS card would, in this case, be the only card read by INOPT.

Table 6.
Option Codes Recognized by INOPT and Function

GODDARD SPACE FLIGHT CENTER
FORTRAN CODING RECORD

PROGRAM | PUNCHING INSTRUCTIONS | GRAPHIC | PAGE OF
PROGRAMMER | DATE | PUNCH | CARD ELECTRO NUMBER*

OPTION CODE	FUNCTION OF OPTION
NO OPTIONS	NO OPTION CARDS TO BE READ
END OPTIONS	END OF OPTION CARDS
INERTIAL	USE INERTIAL COORDINATES FOR COMPUTATION
BODY I	USE BODY I COORDINATES FOR COMPUTATION
ASETX	MODIFY ARRAY SPI
ASETIXC	MODIFY ARRAY SIX
ASETCEN	MODIFY ARRAY SCN
ASETCOR	MODIFY ARRAY SCR
EULERANG	MODIFY ARRAY SEU
SMALLANG	MODIFY ARRAY SMAL
VTINBOD	MODIFY ARRAY SVO
FREE	MODIFY ARRAY SFR
LOCK	MODIFY ARRAY SLK
MOMANGLE	MODIFY ARRAY SMA
CFORCE	MODIFY ARRAY SFK
YARWHEEL	MODIFY ARRAY SMV
COLX	MODIFY ARRAY SXT
CAGE	SOME DEGREES OF FREEDOM CAGED
FLEXIBLE	SOME BODIES ARE TO BE FLEXIBLE
MODE CUP	SOME VIBRATION MODES CROSS-COUPLED

GSFC 6-1(12/69)

2. End Options

If the user is to make use of one or more options, the last card of the cards to be read by INOPT must have the code word END OPTIONS punched on it. If it does not, an input error will result and the program will terminate with an appropriate error message.

*** Optimize Computation Efficiency ***

In order to optimize computation speed, all unnecessary, trivial, and redundant computation must be avoided. By an interrogation of the input data, numerous sets of integers can be defined that can be used to carry out the above task. It is not possible for the computer to pick out "almost trivial" computations; this the engineer must do. By use of particular option cards, he can literally insert engineering judgment into N-BOD2.

In practical application, the author has found that usually there are not enough "almost trivial" terms that can be justifiably deleted to make a significant impact on computation speed. Consequently, while the options are available for the user, the author has rarely made practical use of most of them.

3. Choice of Frame of Computation

The equations of motion, as derived, are in a vector-dyadic form and are valid for all frames of reference. The user may choose the frame of computation to be either the inertially fixed-reference frame or the body 1 fixed-reference frame. All vectors and tensors required for computation are stored in memory in computing frame coordinates. The user should define the frame of computation to be the reference frame in which he wishes to output most of the computed system state variable vectors.

The logic variable INERF is used to define which frame of computation is to be used for internal computation.

INERF = .TRUE. if the frame of computation is to be the inertially fixed-reference frame. Recognition of the code word INERTIAL on an option card will set INERF equal to .TRUE.

INERF = .FALSE. if the frame of computation is to be the body 1 fixed-reference frame. Recognition of the code word BODY 1 on an option card will set INERF equal to .FALSE.

Default INERF = .FALSE.

Body 1 will be the frame of computation unless otherwise directed.

4. Truncate Computation of Pseudo-inertia Tensors

The vector-dyadic form of the simulation model's equations of motion is given in Reference 1 by equation 88 and in Reference 2 by equation 111. The coefficient matrix on the left-hand side of the equation contains numerous inertia and pseudo-inertia tensors that are

defined by equations 63 through 70 of Reference 1. If the simulation model has both large and small bodies, some of the small bodies may contribute a nearly trivial amount to the pseudo-inertia tensors associated with the nests that contain both the large and the small bodies.

To compute the inertia and pseudo-inertia tensors defined by equations 63 through 70 of Reference 1, the integer code words SPI(I), I = 0, 1, . . . , NBOD-1, are used in subroutine XDY.

SPI(I) – code word. Used to define the set of bodies contained in the nest I that contribute significantly in the computation of the inertia and pseudo-inertia tensors associated with the nest I; I = 0, 1, . . . , NBOD-1.

Upon recognition of the code word ASETX, INOPT makes use of the data in columns 13 through 70 of the option card to redefine the code word SPI(I). The following definitions apply for this option:

I = nest number

NSET = number of body labels to be defined to be in the set defined by SPI(I)

(S1 (J), J = 1, NSET) = the body labels defined to be in the set defined by SPI(I).

The code word SPI(I) is created by the operation

CALL COMPAC (S1, NSET, SPI(I)).

Default SPI(I) = SK(I); I = 0, 1, . . . , NBOD-1

Unless otherwise directed, all bodies contained in the nest I are assumed to contribute significantly to the inertia and pseudo-inertia tensors associated with the nest I.

5. Truncate Computation of Inertia Cross-Coupling Torques

In Reference 1, the inertia cross-coupling torque acting on the nest I can be found from equations 79 and 80. In Reference 2, it can be found in equation 92. It is given in the notation of these references as

$$\sum_{\lambda \epsilon S_I} \left[\mathring{\Phi}_\lambda \cdot \vec{\omega}_\lambda + \vec{\omega}_\lambda \times \left(\Phi_\lambda \cdot \vec{\omega}_\lambda + \Delta \vec{L}_{\lambda,\lambda} \right)_i \right]$$

where the summation extends over all bodies contained in the nest I. In subroutine ETA, the integer code words SIX(I), I = 0, 1, . . . , NBOD-1, are used to define which bodies contribute significantly to the above summation.

SIX(I) – code word. Used to define the set of bodies contained in the nest I that contribute significantly to the computation of the inertia cross-coupling torque acting on the nest I; I = 0, 1, . . . , NBOD-1.

Upon recognition of the code word ASETIXC, INOPT makes use of the data in columns 13 through 70 of the option card to redefine the code word SIX(I). The following definitions apply for this option:

I = nest number

NSET = number of body labels to be defined to be in the set defined by SIX(I).

(S1 (J), J = 1, NSET) = the body labels defined to be in the set defined by SIX(I).

The code word SIX(I) is created by the operation

CALL COMPAC (S1, NSET, SIX(I)).

Default Unless otherwise directed, all rigid and flexible bodies contained in the nest I are assumed to contribute significantly to the inertia cross-coupling torque acting on the nest I.

NOTE: Point masses have zero contribution and hence their body labels are not contained in any of the sets defined by the code words SIX(I); I = 0, 1, . . . , NBOD-1.

6. Truncate Computation of Centripetal and some of the Coriolis Cross-Coupling Torques

In Reference 1, the centripetal and Coriolis cross-coupling forces and torques appearing on the right-hand side of equation 88 are defined by equations 75 through 78. In Reference 2, the analogous terms are found in equations 100, 107, and 108.

In the notation of Reference 2, the gyroscopic torque associated with these effects acting on the nest I is given by

$$\sum_{\lambda \epsilon S_I} \vec{\gamma}_{I,\lambda} \times \vec{C}_\lambda \qquad ; I = 0, 1, \ldots, NBOD\text{-}1$$

where

$$\vec{C}_\lambda = m_\lambda \left[\sum_{\substack{i \epsilon S_{0,\lambda-1} \\ i \neq 1}} \left(\vec{\omega}_{J(i)} \times \left(\vec{\omega}_{J(i)} \times \vec{\beta}_i \right) + 2\vec{\omega}_{J(i)} \times \overset{\circ}{\vec{\beta}}_i \right) \right.$$

$$\left. + \vec{\omega}_\lambda \times \left(\vec{\omega}_\lambda \times \vec{\alpha}_\lambda \right) + 2\vec{\omega}_\lambda \times \overset{\circ}{\vec{\alpha}}_\lambda \right]$$

and the summation extends over all bodies contained in the nest I.

In subroutine ETA, the integer code words SCN(I) and SCR(I), I = 0, 1, . . . , NBOD-1, are used to define which bodies contribute significantly to the above summation.

SCN(I) – code word. Used to define the set of bodies contained in the nest I which contribute significantly to the computation of the centripetal and Coriolis cross-coupling torques acting in the nest I; I = 0, 1, . . . , NBOD-1.

SCR(I) – code word. Used to define the set of point masses contained in the nest I for which their Coriolis acceleration effects contribute significantly to the computation of the centripetal and Coriolis cross-coupling torques acting on the nest I; I = 0, 1, . . . , NBOD-1.

Upon recognition of the code words ASETCEN or ASETCOR, INOPT makes use of the data in columns 13 through 70 of the option card to redefine the code words SCN(I) or SCR(I) respectively. The following definitions apply for these options:

I = nest number

NSET = number of body labels to be defined by SCN(I) or SCR(I), respectively.

(S1 (J), J = 1, NSET) = the body labels defined to be in the set defined by SCN(I) or SCR(I), respectively.

The code word SCN(I) is created by the data on the ASETCEN option card by the operation

CALL COMPAC (S1, NSET, SCN(I)).

The code word SCR(I) is created by the data on the ASETCOR option card by the operation

CALL COMPAC (S1, NSET, SCR(I)).

Default Unless otherwise directed, all bodies contained in the nest I are assumed to contribute to the centripetal and Coriolis cross-coupling torques acting on the nest I; I = 0, 1, . . . , NBOD-1.

NOTE: The elements of sets defined by the code words SCR(I), I = 0, 1, . . . , NBOD-1, are only point-mass body labels.

7. Choice of Kinematic Techniques

The user has the choice of which type of kinematic formulation should be used to define the transformation matrices. In the first version of N-BOD2, direction cosine techniques were used exclusively. In an attempt to improve computation speed, it was noted that the

integration of the equations of motion yielded Euler angles directly. These could be used to directly obtain the required transformation matrices and eliminate the need for setting up and integrating six direction cosine equations per transformation matrix. It was further reasoned that time-consuming trigometric operations could also be dropped if it was known that the Euler angles would be small angles. Practical application has shown that the computational operations required to go from Euler angles or small Euler angles to the transformation matrices takes about the same amount of computer time as the operations required to set up and integrate the direction cosine equations.

These three approaches are available for the user in N-BOD2. The author personally uses direction cosine techniques for most simulation work; however, if it is critical that the transformation matrices maintain orthonormality to a high degree of accuracy, Euler angle techniques are preferable. There is no significant computational speed advantage associated with any of the techniques. The Euler angle and small angle approach requires that the body fixed free vectors, about which the Euler angles are measured, be aligned with the coordinate axes of their respective body fixed-reference frames; a restriction not imposed in the direction cosine approach.

In subroutine DCT, the direction cosine equations are set up and in subroutine TRAN, transformation matrices are computed. The code words SD, SEU, and SMAL are used in these subroutines to define the particular kinematic approach that is to be used.

SD = code word. Used to define all contiguous pairs of bodies (JCON(I), I) for which direction cosine techniques are used to compute transformation matrices.

SMAL = code word. Used to define all contiguous pairs of bodies (JCON(I), I) for which small angle techniques are used to compute transformation matrices.

SEU = code word. Used to define all contiguous pairs of bodies (JCON(I), I) for which Euler angle techniques are used to compute transformation matrices.

Upon recognition of the code word EULERANG, the data in columns 13 through 70 are used to create the code word SEU by the operation

CALL COMPAC (S1, NSET, SEU).

Upon recognition of the code word SMALLANG, the data in columns 13 through 70 are used to create the code word SMAL by the operation

CALL COMPAC (S1, NSET, SMAL).

NOTE: Redundancy is not permitted. Pairs of bodies defined by the user to be in either of the sets defined by SEU or SMAL will be subtracted from the set defined by SD

In columns 13 through 17 of the option cards,

I = unused

NSET = the number of pairs of bodies for which the special kinematic techniques are to be used.

(S1 (J), J = 1, NSET) = the labels defining the pairs. The *Jth* pair is defined as body JCON(SI(J)) and body SI(J).

Default Unless otherwise directed, all contiguous pairs of rigid or flexible bodies are contained in the set defined by SD. Both SEU and SMAL are empty sets unless otherwise directed.

8. Suppress Selected Coordinate Transformations

All nontrivial body fixed vectors and tensors are transformed to computing frame coordinates four times per integration step. Under certain conditions, this can be a wasteful operation. If it can be determined that the components of all body fixed vectors and tensors of certain bodies are essentially time invariant in computing frame coordinates, it makes no sense to retransform them at every integration step.

The avoidance of these redundant transformations is controlled by the code word SVD in VDIV and TRANVD.

SVD – code word. Used to define the set of all bodies for which body-fixed vectors and tensors are time varying in the frame of computation.

Upon recognition of the code word VTINBOD, the data in columns 13 through 70 are used to create the code word SVD by the operation

CALL COMPAC (S1, NSET, SVD).

In columns 13 through 70 of the option card,

I = unused

NSET = the number of bodies for which body-fixed vectors and tensors are time varying in the frame of computation.

(S1 (J), J = 1, NSET) = the body labels defined to be in the set defined by SVD.

Default Unless otherwise defined, all body labels are contained in the set SVD. If the computing frame is the body 1 fixed-coordinate frame, body label 1 is automatically subtracted from SVD.

NOTE: Body labels associated with point masses are included in SVD since their nominal center-of-mass positions are given in their respective body-fixed coordinate system.

9. Suppress Recomputation of Elements of Coefficient Matrix of Inertia Tensors

In References 1 and 2, the matrix **X** is a symmetric (NB1) × (NB1) matrix of inertia and pseudo-inertia tensors, the elements of which are given by equations 63 through 70 in Reference 1. Under certain conditions, one or more of the columns (down to the diagonal) of this matrix will have nonzero entries that are nearly constant in time. Unless otherwise directed, the program recomputes these terms four times per integration step. The avoidance of these redundant computations in subroutine XDY is controlled by the code word SXT.

SXT – code word. Used to define the columns of the matrix **X** of psuedo-inertia tensors that have time-varying elements. **X** is symmetric and columns referred to extend down only to the diagonal element. Column NB1 always is taken as having time-varying elements. The elements of column K, K = 2, . . . , NBOD, give the inertia contributions of the nest K-1 to the system's equations of motion.

Upon recognition of the code word COLX, the data in columns 13 through 70 are used to create the code word SXT by the operation

CALL COMPAC (S1, NSET, SXT).

In columns 13 through 70 of the option card

I = unused

NSET = the number of columns to have time-varying elements (not including column NB1, NB1 = NBOD + 1).

(S1 (J), J = 1, NSET) = the column numbers of the columns of **X** that will have time-varying elements.

Default Unless otherwise defined, all body labels are contained in the set SXT. This implies that all nonzero elements of **X** will be assumed to be time varying.

10. Suppress Computation of Relative Displacements

A redundancy exists in the computation of kinematics. All transformation matrices may be computed by a solution of direction cosine equations. Relative displacement angles, which provide basically the same information, are found by an integration of Euler rate equations. If direction cosine methods are used, it usually is not necessary to compute all relative displacement angles since their only use is for a description of torques acting between bodies that are a function of relative displacement angle (springs) and for output display. Accordingly, the user may selectively supress their computation.

The avoidance of the computation of relative displacement angles is controlled by the integer array SFR in subroutines SETUP and ANGLE.

NFRC – total number of free coordinate vectors about or along which relative displacement is to be computed.

SFR (I) – the labels associated with the free coordinate vectors about or along which relative displacement is to be computed. I = 1, 2, . . . , NFRC

Upon recognition of the code word FREE, the data in columns 13 through 70 are used to define the elements of the set SFR. In columns 13 through 70 of the option card,

I = unused

NSET = the number of free coordinate labels to be defined on this card for inclusion is the set SFR.

(SI (J), J = 1, NSET) = the free coordinate labels to be defined to be in the set SFR.

The input data format statement allows a maximum of 10 free coordinate vector labels to be defined per option card. Several option cards with the code word FREE may be used if required.

Default Unless otherwise defined, all relative displacement rate equations will be set up and integrated.

NOTE: INOPT is programmed to prohibit the user from supressing the computation of angles required for use in conjunction with either the Euler angle or the small angle option. It also prohibits the suppression of the computation of the coordinates that define body 1 translational displacement or point mass displacement.

11. Suppress Computation of Symmetric Wheel Angular Position

In spacecraft application, the integration of the symmetric wheel rate equation to ascertain the angular position of a mark on a symmetric wheel is usually a time-consuming and wasteful practice since its rates are usually high relative to the spacecraft frequencies of interest. At times, however, it may be important to know the wheel's angular position.

The determination as to which wheel rate equation should be integrated is controlled in subroutine SETUP by the integer array SMA.

NMOA – total number of symmetric wheel rate equations to be integrated.

SMA (I) – the labels of the symmetric wheels for which wheel angular position is to be computed. I = 1, 2, . . . , NMOA

Default Unless otherwise directed, none of the wheel rate equations will be integrated.

Upon recognition of the code word MOMANGLE, the data in columns 13 through 70 are used to define the elements of the set SMA. In columns 13 through 70 of the option card,

I = unused

NSET = the number of symmetric wheel labels to be defined on this card for inclusion in the set SMA.

(S1 (J), J = 1, NSET) = the symmetric wheel labels to be defined to be in the set

The input data format statement allows a maximum of 10 wheel labels to be defined per option card. Several option cards with the code word MOMANGLE may be used if desired.

12. Computation of Forces and Torques of Constraint

The code words CFORCE and LOCK are recognized and may be used to redefine the elements in the integer array SFK and SLK, respectively. These arrays were to be used to define which forces and torques of constraint should be computed. This option has not been coded and is not available for users of N-BOD2.

Default

NCTC = 0

NFKC = 0

SFK = 0

SLK = 0

***** Expand Modeling Capability *****

13. Specification of Constant Speed Wheels

Wheels that are controlled to spin at a constant speed should be modeled as constant speed wheels. Computationally, this reduces the order of the equations of motion of the system and eliminates the need for defining a control system that will maintain constant wheel rate.

The code word SMV is used in subroutines QFDOT and SETUP to define which wheels are to be treated as variable speed wheels.

SMV – code word. Used to define which of the symmetric wheels are to be treated as variable speed wheels; all others are treated as constant speed wheels.

NMV – total number of variable speed wheels.

Upon recognition of the code word VARWHEEL, the data in columns 13 through 70 are used to create the code word SMV by the operation

CALL COMPAC (S1, NSET, SMV).

In columns 13 through 70 of the option card,

I = unused

NSET = number of wheels to be defined as variable speed wheels.

(S1 (J), J = 1, NSET) = the labels of the variable speed wheels.

Only one VARWHEEL option card is permitted.

Default Unless otherwise directed, all symmetric wheels are assumed to be variable speed wheels.

14. Caging and Uncaging of Degrees of Freedom

In order to simulate the deployment of spacecraft appendages, it is desirable to have the capability to increase the number of allowable degrees of freedom during a simulation run. A "caged degree of freedom" is a degree of freedom that is not allowed for a predefined time. The uncaging of degrees of freedom at predefinable times allows the user to simulate deployment transients.

The following definitions are used:

SCG – total number of caged degrees of freedom at time zero.

SC (I) – labels of the free coordinate vectors about or along which relative motion will be constrained for a predefined time. I = 1, 2, . . ., SCG

TUG (I) – time at which motion about or along free coordinate vector SC(I) will be allowed (uncaged). I = 1, 2, . . . , SCG

NOTE: All uncaging takes place with zero initial relative rate. The coding to account for the impulse effect that is associated with uncaging with an initial relative rate has not been included in N-BOD2.

Upon recognition of the code word CAGE, the procedure for inputting the data required to use this option is started. In columns 13 through 70 of the option card,

I = unused

NSET = SCG, the total number of degrees of freedom caged at time zero.

(S1 (J), J = 1, NBOD) = unused.

Their labels and times for uncaging are then read according to the following:

```
READ 101, (SC(I), TUG(I), I=1, SCG)
101  FORMAT (I5, D 15.5)
```

After these data cards are read, a check is made to make sure that the initial rates associated with the caged degrees of freedom are zero. If not, they are reset to zero.

Default Unless otherwise directed, all degrees of freedom are assumed uncaged at time zero.

15. Flexible Body Option

The user is encouraged to first refer to the section entitled "Flexible versus Rigid Body Modeling" in Reference 2 before making use of the flexible body option. Contained therein are equations that provide a guide for estimating the required number of vibration modes and a discussion of the assumptions implicit in the formulation of the coupled flexible body equations of motion.

N-BOD2 accepts only resultant mode dependent parameters. Their description is based upon the assumption that a finite element model of the flexible body can be defined.*

The following definitions are taken directly from Reference 2 and are used to define the required resultant mode dependent parameters.

$m_{i,\lambda}$ = mass of element i of body λ

${}^{o}\Phi_{i,i,\lambda}$ = inertia tensor of element i of body λ about its own center of mass in the undeformed state of body λ

$\vec{{}^{o}r}_{i,\lambda}$ = position vector from hinge point λ-1 to the undeformed center-of-mass position of the element i of body λ

$\vec{{}^{o}\alpha}_{\lambda}$ = position vector from hinge point λ-1 to the undeformed center-of-mass position of body λ

$\vec{\phi}^{T}_{n,i,\lambda}$ = *nth* normal mode displacement vector for the element i of body λ

$\vec{\phi}^{R}_{n,i,\lambda}$ = *nth* normal mode rotation vector for the element i of body λ

m_{λ} = total mass of body λ

$\delta_{m,n}$ = Kronecker delta function

$N_{i,\lambda}$ = total number of elements i used in the finite element model of body λ

*The user is also referred to Appendix B in which a NASTRAN DMAP program and a preprocessor program are listed. The output of these two programs define the resultant mode dependent parameter required.

The normalization condition imposed upon the modes is

$$\sum_{i\epsilon\lambda} \left[m_{i,\lambda} \vec{\varphi}^{T}_{m,i,\lambda} \cdot \vec{\varphi}^{T}_{n,i,\lambda} + \vec{\varphi}^{R}_{m,i,\lambda} \cdot \left({}^{\circ}\Phi_{i,i,\lambda} \cdot \vec{\varphi}^{R}_{n,i,\lambda} \right) \right] = m_{\lambda} \delta_{m,n}$$

where

$$\lim_{N_{i,\lambda} \to \infty} \sum_{i\epsilon\lambda} \vec{\varphi}^{R}_{m,i,\lambda} \cdot \left({}^{\circ}\Phi_{i,i,\lambda} \cdot \vec{\varphi}^{R}_{n,i,\lambda} \right) = 0$$

The resultant mode dependent parameters required as input data to N-BOD2 are given by equations 48, 50, 51, 54, 55, 67, 70, and 72 in Reference 2. They are

$$\vec{A}_{n,\lambda} = \frac{1}{m_{\lambda}} \sum_{i\epsilon\lambda} m_{i,\lambda} \vec{\varphi}^{T}_{n,i,\lambda}$$

$$D_{n,\lambda} = \sum_{i\epsilon\lambda} m_{i,\lambda} \left[\left({}^{\circ}\vec{r}_{i,\lambda} - {}^{\circ}\vec{\alpha}_{\lambda} \right) \cdot \vec{\varphi}^{T}_{n,i,\lambda} \mathbf{1} - \left({}^{\circ}\vec{r}_{i,\lambda} - {}^{\circ}\vec{\alpha}_{\lambda} \right) \vec{\varphi}^{T}_{n,i,\lambda} \right]$$

$$\vec{B}_{n,\lambda} = \frac{1}{m_{\lambda}} \sum_{i\epsilon\lambda} m_{i,\lambda} {}^{\circ}\vec{r}_{i,\lambda} \times \vec{\varphi}^{T}_{n,i,\lambda}$$

$$\vec{C}_{n,\lambda} = \sum_{i\epsilon\lambda} {}^{\circ}\Phi_{i,i,\lambda} \cdot \vec{\varphi}^{R}_{n,i,\lambda}$$

$$J_{n,\lambda} = \sum_{i\epsilon\lambda} \vec{\varphi}^{R}_{n,i,\lambda} \times {}^{\circ}\Phi_{i,i,\lambda}$$

$$F_{m,n,\lambda} = \sum_{i\epsilon\lambda} m_{i,\lambda} \left[\vec{\varphi}^{T}_{m,i,\lambda} \cdot \vec{\varphi}^{T}_{n,i,\lambda} \mathbf{1} - \vec{\varphi}^{T}_{m,i,\lambda} \vec{\varphi}^{T}_{n,i,\lambda} \right]$$

$$\vec{K}_{m,n,\lambda} = \sum_{i\epsilon\lambda} \left\{ m_{i,\lambda} \vec{\varphi}^{T}_{m,i,\lambda} \times \vec{\varphi}^{T}_{n,i,\lambda} + \frac{1}{2} \left({}^{\circ}\Phi_{i,i,\lambda} \cdot \vec{\varphi}^{R}_{m,i,\lambda} \right) \times \vec{\varphi}^{R}_{n,i,\lambda} \right\}$$

If mode shapes are obtained by a continuum analysis, the rotational inertia effects of the infinitesimal elements about their respective centers of mass is negligible; hence

$$\lim_{N_{i,\lambda}\to\infty} \vec{C}_{n,\lambda} = 0$$

$$\lim_{N_{i,\lambda}\to\infty} J_{n,\lambda} = 0$$

$$\lim_{N_{i,\lambda}\to\infty} \vec{K}_{m,n,\lambda} = \sum_{i\epsilon\lambda} m_{i,\lambda} \vec{\varphi}^{T}_{m,i,\lambda} \times \vec{\varphi}^{T}_{n,i,\lambda}$$

and

$$\lim_{N_{i,\lambda}\to\infty} \left[\sum_{i\epsilon\lambda} m_{i,\lambda} \vec{\varphi}^{T}_{m,i,\lambda} \cdot \vec{\varphi}^{T}_{n,i,\lambda} - m_{\lambda} \delta_{m,n} \right] = 0$$

The other mode-dependent parameters that are used or computed are:

$\omega_{n,\lambda}$ – natural frequency of mode n of body λ (rad/ . . .)

$\zeta_{n,\lambda}$ – damping ratio for mode n of body λ

$a_{n,\lambda}(t)$ – generalized displacement coordinate for mode n of body λ

$\dot{a}_{n,\lambda}(t)$ – generalized displacement coordinate rate for mode n of body λ

The position vector from the undeformed to the deformed center-of-mass position of the element i of body λ is

$$\vec{\epsilon}_{i,\lambda} = \sum_{n,\lambda} a_{n,\lambda}(t) \vec{\varphi}^{T}_{n,i,\lambda}$$

where the summation is overall modes defined for body λ.

The following parameters are defined in subroutine INOPT and used within the various subroutines to compute the effects of body flexibility:

NFLXB – total number of flexible bodies

SFXM (I) – total number of flexible body modes to be used to describe the flexible body characteristics of body I (zero if body I is a rigid body or a point mass); I = 1, 2, . . . , NBOD.

SFLX – code word used to define the body labels of those bodies that are to be treated as flexible bodies.

NMODS – total number of flexible body modes used for entire simulation model.

Upon recognition of the code word FLEXIBLE, the procedure for inputting uncoupled modal data starts. In columns 13 through 70 of the option card,

I = unused

NSET = total number of bodies to be defined to be flexible.

(S1 (I), I = 1, NBOD) = the number of flexible body modes to be used to describe the flexible body characteristics of body I; I = 1, 2, . . ., NBOD. (Zero if body I is a rigid body or point mass).

The parameters NFLXB, SFXM, SFLX, and NMODS are computed by an internal process that makes use of the above data provided on the option card.

To conserve computer storage and increase computation speed, all mode-dependent parameters are sequentially numbered from 1 to NMODS by an internally defined algorithm that makes use of the SFXM integer array. All parameters associated with mode M of body K are given the integer label MN which is computed by the algorithm

$$MN = M + \sum_{I=1}^{K-1} SFXM(I)$$

where

$$K = 2, 3, \ldots, NBOD$$

and

$$M = 1, 2, \ldots, SFXM(K)$$

MN will be referred to as the "mode number."

All uncoupled mode-dependent data required by N-BOD2 are read immediately after recognition of the code word FLEXIBLE on an option card. For mode M of flexible body K, the following data must be defined. Its mode number MN is defined as

$$MN = M + \sum_{I=1}^{K-1} SFXM(I)$$

and recall that

NFER = total number of free coordinate vectors.

Then

FLOM (MN) = $\omega_{M,K}$ the modal natural frequency (rad/. . .)

ZETA (MN) = $\zeta_{M,K}$ the modal damping ratio

THA (NFER + MN) = $a_{M,K}$ (t) generalized displacement coordinate (initial value to be inputted)

THAD (NFER + MN) = $\dot{a}_{M,K}$ (t) the generalized displacement coordinate rate (initial value to be inputted)

{FLA (I, MN)} = components of the resultant mode-dependent vector $\vec{A}_{M,K}$ relative to the body K fixed reference frame.

{FLB (I, MN)} = components of the resultant mode-dependent vector $\vec{B}_{M,K}$ relative to the body K fixed reference frame.

{FLC (I, MN)} = components of the resultant mode-dependent vector $\vec{C}_{M,K}$ relative to the body K fixed reference frame.

[FLD (I, J, MN)] = components of the resultant mode-dependent dyad $\mathbf{D}_{M,K}$ relative to the body K fixed reference frame.

[FLJ (I, J, MN)] = components of the resultant mode-dependent dyad $\mathbf{J}_{M,K}$ relative to the body K fixed reference frame.

The data required to describe the uncoupled modal data for each flexible body are read according to the following sequence of read statements.

```
************ READ STATEMENTS FOR OPTION 'FLEXIBLE' IN INOPT ************
C
C      INITIALIZE MODE LABELS
       MN = 0
C
************ READ UNCOUPLED MODE DATA FOR ALL FLEXIBLE BODIES ***********
       DO 1 K=1,NBOD
       IF(SFXM(K).EQ.0) GO TO 1 BODY K IS NOT A FLEXIBLE BODY.
       READ 103, N,(MESS(J),J=1,18)
       IF(K.NE.N) ERROR.CARDS OUT OF SEQUENCE, PRINT MESSAGE HALT
C         N = BODY LABEL FOR WHICH FLEXIBLE BODY DATA TO FOLLOW APPLIES
C      MESS = ALPHANUMERIC DESCRIPTION OF BODY N, PRINTED WITH DATA ECHO
C
************ READ EXACTY SFXM(K) SETS OF MODAL DATA FOR BODY K *********
       DO 2 M =1,SFXM(K)
       MN = MN + 1
       READ 104, FLOM(MN),ZETA(MN)
       READ 104, THA(NFER+MN),THAD(NFER+MN)
       READ 104, (FLA(I,MN),I=1,3)
       READ 104, (FLB(I,MN),I=1,3)
       READ 104, (FLC(I,MN),I=1,3)
       READ 104,((FLD(I,J,MN),J=1,3),I=1,3)
       READ 104,((FLJ(I,J,MN),J=1,3),I=1,3)
    2  CONTINUE
************ ALL UNCOUPLED MODAL DATA FOR BODY K HAS BEEN READ *********
C
C
    1  CONTINUE
```

```
************ ALL UNCOUPLED MODAL DATA HAS BEEN READ **********************
C
C
************     END OF DATA FOR OPTION 'FLEXIBLE     **********************
C
C
C      FORMAT STATEMENTS
  103 FORMAT (I5,18A4)
  104 FORMAT (3D15.5)
```

The input of the coupled modal data follows the option card 'MODE CUP.'

In the process of developing the equations that define centripital and Coriolis related loads associated with flexible bodies, certain mode-dependent, cross-coupling parameters are derived; namely $F_{m,n,\lambda}$ and $\vec{K}_{m,n,\lambda}$ (previously defined in this section). In many practical applications, most, if not all, of these terms are identically or nearly equal to zero. Computer time and storage can be conserved by developing a scheme which will only use nontrivial cross-coupling terms.

The following parameters are defined in subroutine INOPT and used in subroutine QFDOT to compute the modal cross-coupling effects.

SFCC = code word. Used to define the body labels of the flexible bodies that have flexible body modes that are cross coupled (significantly).

SCXC (MN) = code word. Used to define the modes M of body K which significantly cross couple in the generalized displacement coordinate equation for the mode having mode number MN

$$MN = N + \sum_{I=1}^{K-1} SFXM(I)$$

From equation 95 of Reference 2, the cross-coupling terms in the equation for the generalized displacement coordinate associated with mode n of body λ are

$$\vec{\omega}_\lambda \cdot \left[\sum_{m,\lambda} F_{m,n,\lambda}\, a_{m,\lambda}(t)\right] \cdot \vec{\omega}_\lambda - 2\vec{\omega}_\lambda \cdot \sum_{m,\lambda} \dot{a}_{m,\lambda}(t)\, \vec{K}_{m,n,\lambda}$$

More simply expressed, the SCXC array defines which modes m of body λ shall be used in evaluating the above summations.

In the input deck, the code word MODE CUP must follow FLEXIBLE. Upon recognition of the code word MODE CUP, the procedure for inputting coupled modal data starts. In columns 13 through 70 of the option card,

I = unused

NSET = unused

(S1 (I), I = 1, NBOD) = number of blocks of mode dependent cross coupling data to be read in for body I; a block of data consists of the three integers (M, N, K), the components of the tensor $\mathbf{F}_{M,N,K}$, and the components of the vector $\vec{K}_{M,N,K}$

The parameter SFCC is computed by an internal process which makes use of the above data provided on the option card.

To conserve computer storage and increase computational speed, all coupled mode-dependent data are sequentially numbered. Strict adherence to the following input logic is required.

```
MN = 0
KF = 0
DO 1 K=1, NBOD
DO 1 N=1, SFXM(K)
MN = MN+1
DO 1 M = 1, SFXM(K)
IF [F(M,N,K) INSIGNIFICANT .AND. K(M,N,K) INSIGNIFICANT] GO TO 1
READ M, N, K
KF = KF+1
READ F(M,N,K) and K(M,N,K)
LOAD F(M,N,K) into FCF array, starting at location (1,1,KF)
LOAD K(M,N,K) into FCK array, starting at location (1,KF)
ADD integer M to code word SCXC(MN)
1 CONTINUE
```

where

MN — mode number for mode N of body K

KF — storage location integer designator for modal cross-coupling parameters

[FCF(II,JJ,KF)] — components of the resultant coupled mode-dependent tensor $\mathbf{F}_{M,N,K}$ relative to the body K fixed reference frame (if M = N, significant terms usually exist).

[FCK(II,KF)] — components of the resultant coupled mode dependent vector $\vec{K}_{M,N,K}$ relative to the body K fixed reference frame.

All coupled mode-dependent data required by N-BOD2 are read immediately after recognition of the code word MODE CUP on an option card. If flexible bodies are defined and the 'MODE CUP' option card and associated data are not in the option deck, all computation required to define modal cross-coupling effects is skipped.

The data required to describe the coupled modal data for each flexible body are read according to the following sequence of read statements:

```
                READ STATEMENTS FOR OPTION 'MODE CUP' IN INOPT
C
C     INITIALIZE COUPLED MODE LABELS
      KF = 0
C
                READ ALL NON-TRIVIAL COUPLED MODE PARAMETERS
C
      DO 1 K = 1, NBOD
C
C     S1(K)     = NUMBER OF DATA-BLOCKS FOR BODY K
C     SFXM(K)   = NUMBER OF MODES USED FOR BODY K
      IF(SFXM(K).EQ.0) GO TO 1    (BODY K IS RIGID)
      IF(SI(K).NE.0.) GO TO 2     (SOME CROSS COUPLING FOR BODY K)

      GO TO 1
    2 DO 1 M = 1, S1(K)
      READ 105 MB, NB, KB
      KF = KF + 1
      READ 104, (FCF(II,JJ,KF),JJ = 1, 3),II = 1, 3)
      READ 104, (FCK(II,KF),II = 1, 3)
    1 CONTINUE
C
                          ALL COUPLED MODAL DATE IN
C
C
C     FORMAT STATEMENTS
  104 FORMAT (3E15.5)
  105 FORMAT (3I5)
```

Default Unless otherwise directed, the simulation model will consist only of rigid bodies, point masses, and symmetric wheels. All parameters associated with flexible bodies will be set equal to zero.

PROGRAMMER'S GUIDE TO SUBROUTINE INOPT

The main function of subroutine INOPT is to accept the data that will define the parameters associated with the various computational options available to the user. It is structured to

grow. As new modeling requirements are defined and their associated equations developed, INOPT is the subroutine in which the basic data required to define the new option are to be entered. This is done by simply defining another code word that will be recognized along with the FORTRAN coding required to set up the basic data necessary to describe the new option.

Subroutine INOPT has the following structure:

- Define by default all parameters associated with available options.
- Read option card and compare option code with library of recognizable codes.
- Branch to appropriate program point, redefine option-related parameters, input any additional data required.
- Branch back to program point 12 to read another option card.
- Upon recognition of the 'END OPTIONS' code word, check all option data for basic incompatabilities.
- Perform any final setup work that must be done after all option data have been inputted.
- Print an echo of all option related data defined in INOPT.

The following variables, defined in table 1, are either first introduced or used extensively in subroutine INOPT. The programmer is referred to the comment cards inserted therein, and the following, for detailed definition.

CT1	–	counter. Zeroed before exit from INOPT
CT2	–	counter. Zeroed before exit from INOPT
CT3	–	counter. Zeroed before exit from INOPT
CT4	–	counter. Zeroed before exit from INOPT (See DYN)
CT5	–	counter. Zeroed before exit from INOPT
[FCF(I,J,K)]	–	See section "Data Input/INOPT," subsection 15
{FCK(I,K)}	–	See section "Data Input/INOPT," subsection 15
FG3	–	input error flag. If the code word on any option card is not recognized, it is set to .FALSE.
{FLA(I,MN)}	–	See section "Data Input/INOPT," subsection 15
{FLB(I,MN)}	–	See section "Data Input/INOPT," subsection 15
{FLC(I,MN)}	–	See section "Data Input/INOPT," subsection 15
[FLD(I,J,MN)]	–	See section "Data Input/INOPT," subsection 15

[FLJ(I,J,MN)]	–	See section "Data Input/INOPT," subsection 15
FLOM(MN)	–	See section "Data Input/INOPT," subsection 15
INERF	–	See section "Data Input/INOPT," subsection 3
NCTC	–	See section "Data Input/INOPT," subsection 12
NFKC	–	See section "Data Input/INOPT," subsection 12
NFLXB	–	See section "Data Input/INOPT," subsection 15
NFRC	–	See section "Data Input/INOPT," subsection 10
NMOA	–	See section "Data Input/INOPT," subsection 11
NMODS	–	See section "Data Input/INOPT," subsection 15
NMV	–	See section "Data Input/INOPT," subsection 13
SC(I)	–	See section "Data Input/INOPT," subsection 14
SCXC(K)	–	See section "Data Input/INOPT," subsection 15
SCG	–	See section "Data Input/INOPT," subsection 14
SCN(I)	–	See section "Data Input/INOPT," subsection 6
SCR(I)	–	See section "Data Input/INOPT," subsection 6
SD	–	See section "Data Input/INOPT," subsection 7
SEU	–	See section "Data Input/INOPT," subsection 7
SFCC	–	See section "Data Input/INOPT," subsection 15
SFK(I)	–	See section "Data Input/INOPT," subsection 12
SFLX	–	See section "Data Input/INOPT," subsection 15
SFR(I)	–	See section "Data Input/INOPT," subsection 10
SFXM(I)	–	See section "Data Input/INOPT," subsection 15
SIX(I)	–	See section "Data Input/INOPT," subsection 5
SLK(I)	–	See section "Data Input/INOPT," subsection 12
SMA(I)	–	See section "Data Input/INOPT," subsection 11
SMAL	–	See section "Data Input/INOPT," subsection 7
SMC(K)	–	code word. Used to define the symmetric wheel labels associated with all wheels contained in the nest K-1; K = 1, 2, . . . , NBOD
SMV	–	See section "Data Input/INOPT," subsection 13

SPI(I) – See section "Data Input/INOPT," subsection 4

SVD – See section "Data Input/INOPT," subsection 8

SXM(I,K) – computed integer array. Used in subroutine TRAN if the transformation matrices for body K are to be computed using Euler-angle techniques. SXM(I,K) defines the coordinate axis (±1, ±2, ±3) in body K along which the free coordinate vector M+(I-1), I = 1, . . . , 3-PCON(K), is aligned in the zero state (sign implies direction). It is computed only if needed; otherwise it is set to zero.

SXT – See section "Data Input/INOPT," subsection 9

TUG(I) – See section "Data Input/INOPT," subsection 14

ZETA(MN) – See section "Data Input/INOPT," subsection 15

DATA INPUT, SUBROUTINE INTOR

The third and last input data subroutine to be entered is INTOR. Normally this routine contains no executable instruction. It is for the user. The user may insert any desired input statements and store the resultant data in common block /SATELL/. The data inputted here usually consist of all data needed to define the parameters in the user-defined description of the forces and torques acting on the simulation model.

There are no new variables introduced in this subroutine other than those that are user defined.

PROGRAMMER'S GUIDE TO SUBROUTINE TRNSIV

The primary purpose of the computation coded in subroutine TRNSIV is to compute initial values for all required transformation matrices. The user need never interface with this subroutine.

All required computation carried out in N-BOD2 is done relative to the frame of computation (see section "Data Input/INOPT," subsection 3). Consequently, transformation matrices which take vectors from body fixed coordinates to computing frame coordinates must be evaluated. TRNSIV computes the initial values of these matrices based upon the data inputted in subroutine INBS. Quarternion techniques are applied since gimbal axes need not be aligned with body fixed coordinate axes (see Reference 1, section "Initial Orientation of the N-Body System" and Appendix "Quarternion Techniques").

The following variables, defined in table 1, are first introduced in subroutine TRNSIV. All other variables used therein have been previously defined. The programmer is referred to the comment cards inserted in TRNSIV, and the following, for detailed definitions:

[XMC(I,J,K)]

for K = 0 – transformation matrix used to transform vectors from inertially fixed coordinates to the frame of computation.

for K = 1, 2, . . . , NBOD — transformation matrix used to transform vectors from body K fixed coordinates to the frame of computation,

where

$$[XMC(I,J,K)] = \begin{bmatrix} XMC(1,1,K) & XMC(1,2,K) & XMC(1,3,K) \\ XMC(2,1,K) & XMC(2,2,K) & XMC(2,3,K) \\ XMC(3,1,K) & XMC(3,2,K) & XMC(3,3,K) \end{bmatrix}$$

For any particular problem, all computation carried out in TRNSIV can be outputted on the line printer by defining

LTRNSI = .TRUE.

on card 1 of the data deck. There is a one-for-one correspondence between the line printer output and the theoretical notation in Reference 1.

PROGRAMMER'S GUIDE TO SUBROUTINE VDIV

The initial values for all body fixed vectors and tensors, relative to the frame of computation, are computed in subroutine VDIV. The user need never interface with this subroutine.

All vector-tensor operations required to define the system equations of motion are carried out in computing frame coordinates. N-BOD2 attempts to avoid redundant and trivial computation by transforming all body fixed vectors and tensors, used more than once, into computing frame coordinates at time zero. Thereafter, integer sets, computed in VDIV, are used to transform, at every integration step, only those vectors and tensors that have non-trivial time-varying components relative to computing frame coordinates.

The following variables, defined in table 1, are first introduced in subroutine VDIV. All other variables used therein have been previously defined. The programmer is referred to the comment cards inserted in VDIV, and the following, for detailed definitions:

{CAC(I,K)} for K = 1, 2, . . . , NBOD — components of the vector from hinge point K-1 to the deformed center-of-mass position of body K, relative to the frame of computation (see Reference 2, equation 47).

{CAO(I,K)} for K = 1, 2, . . . , NBOD — components of the vector from hinge point K-1 to the undeformed center-of-mass position of body K, relative to body-K fixed coordinates.

{CBC(I,K)} for K = 0 — unused. Reserved for the components of the vector from the inertial origin to the composite system center-of-mass, if needed.

for K = 1, 2, . . . , NBOD — components of the vector from hinge point JCON(K)-1 to hinge point K-1, relative to the frame of computation (N-BOD2 *assumes* zero elastic deformation at hinge points).

{FLAC(I,MN)} — components of the mode-dependent vector defined by equation 48 of Reference 2 relative to the frame of computation. It is used in equation 47 to define the contribution of mode M deformation of body K to the vector which defines the deformed center-of-mass location of body K, where mode number MN is defined as

$$MN = M + \sum_{I=1}^{K-1} SFXM(I)$$

$$K = 2, 3, \ldots, NBOD$$
$$M = 1, 2, \ldots, SFXM(K)$$

[FLE(I,J,MN)] — components of the mode-dependent tensor defined by equation 52 of Reference 2 relative to body K fixed coordinates. It is used in equation 49 to define the contribution of mode M deformation of body K to the deformed state inertia tensor of body K, where MN is the mode number defined as above.

[FLH(I,J,MN)] — components of the mode-dependent tensor defined by equation 68 of Reference 2 relative to body K fixed coordinates. It is used in equation 65 of Reference 2 to define a component of the generalized force acting on mode M of body K, where MN is the mode number defined as above.

{FLQ(I,MN)} — components of the mode-dependent vector defined by equation 56 of Reference 2 relative to the body K fixed reference. It is used in equation 53 of Reference 2 to compute the contribution of mode M deformation rate to the angular momentum of body K, where MN is the mode number as defined above.

{FLQC(I,MN)} — components of the mode-dependent vector defined by equation 56 of Reference 2 relative to the computing frame.

{GAM(I,KL)}	– unused in VDIV; however, all elements of the array GAM are set equal to zero in VDIV.
{HMC(I,M)} for M = 1, 2, . . . , MNO	– components of the unit vector directed along the spin axis of symmetric wheel M relative to the frame of computation (N-BOD2 assumes zero elastic deformation at wheel attachment points).
NSVP	– total number of locked coordinate vectors to be transformed to computing frame coordinates at every integration step.
NSVQ	– total number of free coordinate vectors to be transformed to computing frame coordinates at every integration step.
{QFC(I,M)} for M = 1, 2, . . . , NFER	– components of free coordinate vector M relative to the frame of computation.
{QLC(I,L)} for L = 1, 2, . . . , NLOR	– components of locked coordinate vector L relative to the frame of computation.
SOK(K) for K = 1, 2, . . . , NBOD	– code word used to define the body labels of those bodies on the topological path from hinge point 0 to the center-of-mass of body K.
SOK(NBI) for NBI = NBOD+1	– code word used to define the set of integers from 1 to NBOD+1 inclusive.
SSCN	– code word used to define the union of all labels contained in the sets of integers defined by the code words SCN(K); K = 0, 1, . . . , NBOD-1.
SSIX	– code word used to define the union of all labels contained in the sets of integers defined by the code words SIX(K); K = 0, 1, . . . , NBOD-1.
SVA	– code word used to define the set of center-of-mass vectors that have nontrivial time-varying components in the frame of computation.
SVB	– code word used to define the set of hinge-point vectors that have nontrivial time-varying components in the frame of computation.
SVI	– code word used to define the set of inertia tensors that have nontrivial time-varying components in the frame of computation.

SVM	–	code word used to define the set of symmetric-wheel spin vectors that have nontrivial time-varying components in the frame of computation.
SVP(L) for L = 1, 2, . . . , NSVP	–	the labels associated with the locked coordinate vectors that must be transformed at every integration step.
SVQ(M) for M = 1, 2, . . . , NSVQ	–	the labels associated with the free coordinate vectors that must be transformed at every integration step.
[XDIC(I,J,IK)]	–	unused in VDIV; however, all elements of the array XDIC are set equal to zero in VDIV.
[XIC(I,J,K)] for K = 1, 2, . . . , NBOD	–	components of the inertia tensor of body K about its center-of-mass in the deformed state relative to the frame of computation (see Reference 2, equation 49).
[XIO(I,J,K)] for K = 1, 2, . . . , NBOD	–	components of the inertia tensor of body K about its undeformed center-of-mass position in the undeformed state relative to the body K fixed reference.

For any particular problem, all computation carried out in VDIV can be outputted on the line printer by defining

LVDIV = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE EQIV

The complete description of the motion of the coupled-body system is defined by the simultaneous solution of a set of first-order, nonlinear, ordinary differential equations. Subroutine EQIV is used to compute the total number of first-order equations to be integrated (excluding those that are user defined) and to set up an array of initial conditions in a form compatible with the integration subroutine RUNGE. The user need never interface with this subroutine.

The following variables, defined in table 1, are first introduced in subroutine EQIV. All other variables used therein have been previously defined. The programmer is referred to the comment cards inserted in EQIV, and the following, for detailed definitions:

NEQ	–	total number of first-order differential equations that will be set up and numerically integrated via RUNGE (excluding those that are user defined in TORQUE).

In the coding of the equations of motion, the system state variables and their associated time derivatives are stored in various arrays which are given different symbolic names.

To effectively use the integration subroutine, these must all be reloaded into two one-dimensional arrays so that the equations to be integrated take on the general form

$$\dot{y}_N(t) = f_N\left[\dot{y}_1(t), \ldots, \dot{y}_{NEQ}(t), y_1(t), \ldots, y_{NEQ}(t), t\right]$$

where

$$N = 1, 2, \ldots, NEQ$$

Y(N) — one dimensional array in which all state variables that are determined by integration are stored for entry into the integration package. In EQIV, all initial values are defined

$$N = 1, 2, \ldots, NEQ.$$

The following sequential ordering scheme is used to define the location of each system state variable in the array Y.

- Angular and linear rates about or along all of the free-coordinate vectors

  THAD(N); N = 1, 2, . . . , NFER.

- Generalized elastic coordinate rates for all modes of vibration

  THAD(N); N = NFER + 1, . . . , NFER + NMODS.

- Angular rates for all variable-speed wheels

  THADW(M); M defined by code word SMV.

- Displacements about or along selected free-coordinate vectors

  THA(SFR(N)); N = 1, 2, . . . , NFRC.

- Generalized elastic coordinates, all of them

  THA(N); N = NFER + 1, . . ., NFER + NMODS.

- Angular position of selected wheels

  THAW(SMA(M)); M = 1, 2, . . . , NMOA.

- Elements of the first two columns of each transformation matrix that is obtained by direction-cosine technique

  XMC(I,J,M); I = 1, 2, 3
  J = 1, 2
  M defined by code word SD.

Refer to subroutine listing for exact details if needed. For any particular problem, all computation carried out in EQIV can be outputted on the line printer by defining

LEQIV = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE TRAN

Numerical solution of the vector-dyadic equations of motion requires that the orientation of all body fixed-reference frames relative to each other and to the inertially fixed frame be obtainable. A minimal set of transformation matrices is the set that defines the relative orientation of each body fixed and inertially fixed reference frame to the frame of computation. These transformation matrices are evaluated in subroutine TRAN. The user need never interface with this subroutine.

As discussed in section "Data Input Subroutine INOPT," subsection 7, three kinematic methods for obtaining transformation matrices are available for the user. TRAN makes use of the data stored in the code words SD, SMAL, and SEU to determine the kinematic technique to be used for the computation of each respective transformation matrix.

The following variable, defined in table 1, is recomputed in subroutine TRAN. All other variables used have been previously defined. The programmer is referred to the comment cards inserted in TRAN, and the following, for detailed definitions:

[XMC(I,J,K)] – see section "Programmer's Guide to Subroutine TRNSIV." In subroutine TRAN, which is entered four times per integration step, the elements of all transformation matrices that have time-varying coefficients are recomputed.

If direction-cosine techniques are used, the first two columns of the transformation matrix [XMC(I,J,K)], namely

$$\{XMC(I,1,K)\} \quad \text{and} \quad \{XMC(I,2,K)\}$$

are computed by integration of direction-cosine rate equations. The third column

$$\{XMC(I,3,K)\}$$

is computed in TRAN by a simple vector cross-product operation. No attempt is made to ensure that the resultant matrix is orthonormal.

The author is of the opinion that since it generally is impossible to determine which of the six integrated direction cosine parameters is contributing most strongly to any error buildup, it is also impossible to define an orthonormalization procedure which would be satisfactory for all possible situations. If orthonormalization of transformation matrices must be maintained to a high degree of accuracy, then Euler angle techniques should be called for in INOPT.

If Euler angles are used, the appropriate free coordinate vectors must be aligned with body fixed coordinate axes in the nominal zero displacement state. Euler angle techniques are used to obtain only the transformation matrix between pairs of contiguous bodies. The transformation matrix to the computing frame is obtained by an appropriate matrix multiplication. The same logic is used for the small-angle technique with the exception that the approximation

$$\theta_M = \sin \theta_M$$

is employed.

For any particular problem, all computation carried out in TRAN can be outputted on the line printer by defining

LTRAN = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE TRANVD

The primary purpose of TRANVD is to transform to computing frame coordinates all body fixed vectors and tensors, which are used more than once in the evaluation of the equations of motion. The user need never interface with this subroutine.

The following variables defined in table 1 are recomputed each time TRANVD is called by DYN:

{CAC (I,K)}	{HMC (I,M)}
{CBC (I,K)}	{QFC (I,M)}
{FLAC (I, MN)}	{QLC (I,L)}
{FLQC (I,MN)}	[XIC (I,J,K)]

Each of the above variables is defined in the section "Programmers Guide to Subroutine VDIV." In subroutine TRANVD, which is entered four times per integration step, the transformation matrices evaluated in TRAN are used to obtain the components of each of the above body fixed vectors and tensors in computing frame coordinates. Only those that have time-varying components in the computing frame are recomputed; all others retain the initial value that was set in VDIV.

For any particular problem, all computation carried out in TRANVD can be outputted on the line printer by defining

LTRANV = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE RATE

The primary purpose of subroutine RATE is to compute all linear and angular velocity vectors that will be required by the succeeding subroutines. The user need never interface with this subroutine.

In the coding of N-BOD2, all hinge points of the system are assumed to be either on rigid bodies or at node points of flexible bodies. Consequently, all rate-related effects associated with the deformation dependent motion of hinge points are identically equal to zero. That is, in Reference 2, equation 126, all elements of the partition φ_H^R, φ_W^R, and φ_H^T equal zero.

The following variables are defined in table 1 and recomputed each time subroutine RATE is called by DYN:

$\{ROMC(I,K)\}$ $\{COMC(I,K)\}$

$\{FOMC(I,K)\}$ $\{DOMC(I,K)\}$

In Reference 1, equation 141, the partition that defines relative rate between contiguous bodies is

$$\{\tilde{\omega}\} = [q]\{\dot{\theta}\}$$

In Reference 2, equation 126, it is

$$\{\tilde{\omega}\} = \{^{\circ}\tilde{\omega}\} + [\varphi_H^R]\{\dot{a}\}$$

where, by equation 133, Reference 2,

$$\{^{\circ}\tilde{\omega}\} = [q]\{\dot{\theta}\}$$

and under the restrictions of N-BOD2

$$[\varphi_H^R] = [0]$$

The components, relative to the computing frame, of the NB1 vectors contained in the ni 1 column matrix $\{\tilde{\omega}\}$ are stored in the array ROMC according to the following storage convention:

- Body K is a rigid or flexible body; K = 1, 2, . . . , NBOD

$$\{ROMC(I,K)\} = \{\vec{\omega}_K\}_c = \sum_{M@K-1} \dot{\theta}_M \{\vec{q}_M\}_c$$

- Body K is a point mass; K = 2, 3, . . . , NBOD

$$\{\text{ROMC (I,K)}\} = \{\mathring{\vec{\alpha}}_K\}_c = \sum_{M@K\text{-}1} \dot{\theta}_M \{\vec{q}_M\}_c$$

- K = NB1 implies linear velocity of the center-of-mass of body 1 relative to inertial origin

$$\{\text{ROMC (I,K)}\} = \{\mathring{\vec{\beta}}_1\}_c = \sum_{M@K\text{-}1} \dot{\theta}_M \{\vec{q}_M\}_c$$

Note that by applying equation 134 of Reference 1, namely

$$\vec{\tilde{\omega}}_K = \sum_{M@K\text{-}1} \dot{\theta}_M \vec{q}_M$$

all trivial multiplications by zero called for by the matrix equation

$$\{\tilde{\omega}\} = [q]\{\dot{\theta}\}$$

are circumvented.

The relative angular-rate vectors stored in the array ROMC are used to compute the inertial angular-rate vectors; the components of which, relative to the computing frame, are stored in the array FOMC. Making use of equation 37 of Reference 1, the following parameters are stored in the array FOMC:

- Body K is a rigid or flexible body; K = 1, 2, . . . , NBOD

$$\{\text{FOMC (I,K)}\} = \{\vec{\omega}_K\}_c = \sum_{i \in S_{0,K\text{-}1}} \{\vec{\tilde{\omega}}_i\}_c$$

- Body K is a point mass, since hinge point K-1 is at a node point on body JCON(K); K = 2, 3, . . . , NBOD

$$\{\text{FOMC (I,K)}\} = \{\vec{\omega}_{J(K)}\}_c$$

- K = NB1

$$\{\text{FOMC (I,K)}\} = \{\text{ROMC (I,K)}\}$$

The elements of this array FOMC are used primarily to compute gyroscopic cross-coupling loads.

The user may choose the frame of computation to be either the inertially fixed-reference frame or the body 1 fixed-reference frame. In Reference 1, pages 34 and 35, the angular rate of the body K fixed-reference relative to the computing frame is defined. The components of these vectors, relative to the computing frame, are stored in the array COMC; that is,

- Body K is a rigid or flexible body; K = 1, 2, . . . , NBOD

$$\{\text{COMC (I,K)}\} = \{{}_c\vec{\omega}_K\}_c$$

- Body K is a point mass; K = 2, 3, . . . , NBOD

$$\{\text{COMC (I,K)}\} = \{{}_c\vec{\omega}_{J(K)}\}_c$$

- K = NB1

$$\{\text{COMC (I,K)}\} = \{\text{ROMC (I,K)}\}$$

The elements of this array COMC are used primarily in the evaluation of the direction cosine equations to be integrated.

In Reference 1, equation 136, differentiation of the relative angular-velocity vector yields

$$\dot{\vec{\omega}}_K = \sum_{M@K-1} [\ddot{\theta}_M \vec{q}_M + \dot{\theta}_M \dot{\vec{q}}_M]$$

The second term reflects the fact that free-coordinate vectors are not inertially fixed. This term carries through the development and appears in the final partitioned matrix form of the equations of motion: Reference 1, equation 144; or Reference 2, equation 136; as

$$[\dot{q}] \{\dot{\theta}\}$$

The elements of the column matrix are computed and stored in the array DOMC according to the following storage convention:

- Body K is a rigid or flexible body; K = 1, 2, . . . , NBOD

$$\{\text{DOMC (I,K)}\} = \sum_{M@K-1} \dot{\theta}_M \{\dot{\vec{q}}_M\}_c$$

- Body K is a point mass K = 2, 3, . . . , NBOD or K = NB1;

$$\{\text{DOMC (I,K)}\} = \{0\}$$

The components of the vectors $\dot{\vec{q}}_M$ are needed only for the above computation and are not stored in COMMON; furthermore, only those having nonzero components are evaluated.

For any particular problem, all computation carried out in RATE can be outputted on the line printer by defining

LRATE = .TRUE.

on card 1 of the data deck.

In the coding of subroutine RATE, it should be noted that by use of the EQUIVALENCE statement the single subscript arrays EFOMC, EROMC, ECOMC, and EDOMC, are used interchangeably with the double subscript arrays FOMC, ROMC, COMC, and DOMC, respectively. This is done to optimize computation speed.

PROGRAMMER'S GUIDE TO SUBROUTINE XDY

The primary purpose of subroutine XDY is to compute all required hinge point to center-of-mass position vectors and all inertia and pseudo-inertia tensors. The user need never interface with this subroutine.

On the left-hand side of the system equations of motion given in Reference 1 by equation 144 and in Reference 2 by equation 136, the computation of numerous pseudo-inertia tensors is required. To compute these tensors, it is also required to compute the components of the position vectors which locate center-of-mass position relative to hinge points.

In TRANVD, the center-of-mass vector, hinge-point vector, and inertia tensor for each body of the system is computed. Once these vectors and tensors are determined, no distinction need be made between rigid and flexible bodies for the computation of inertia and pseudo-inertia tensors of nests of bodies. Consequently, equations 11 and 63 through 70 of Reference 1 may be applied to define the contents of the matrix of inertia tensors defined by the partition

[X]

in the equations of motion.

The following variables are defined in table 1 and computed in XDY:

{GAM (I,KL)}

[XDIC (I,J,KI)]

In Reference 1, equation 11, hinge point to center-of-mass position vectors are defined by the equation

$$\vec{\gamma}_{K-1,L} = \sum_{\substack{i \in S_{K-1,L-1} \\ i \neq K}} \vec{\beta}_i + \vec{\alpha}_L$$

where

$$K = 1, 2, \ldots, NBOD \qquad \text{and}$$

$$K \leqslant L$$

To conserve computer storage and eliminate a superfluous dimension, the integer function KT0, defined along with the other utility subroutines, is used to store the triangular matrix of vectors in a two-dimensional array. That is

$$\{GAM(I,KL)\} = \{\vec{\gamma}_{K-1,L}\}_c$$

where

$$KL = KT0(NB1, K-1, L)$$

In Reference 1, equations 63 through 70, the tensor in row K, column I, $I \geqslant K$, is defined by the equation that satisfies one of the following criteria:

1. $K \epsilon S_R$, $I \epsilon S_{K-1}$, $I \epsilon S_R$

$$X_{K,I} = \sum_{\substack{\lambda \epsilon S_{I-1} \\ \lambda \notin S_L}} \Phi_\lambda + \sum_{\lambda \epsilon S_{I-1}} G^{\lambda}_{K-1,I-1}$$

2. $K \epsilon S_R$, $I \epsilon S_{K-1}$, $I \epsilon S_L$

$$X_{K,I} = - m_I \Gamma_{K-1,I}$$

3. $K \epsilon S_R$, $I \notin S_{K-1}$

$$X_{K,I} = 0$$

4. $K \epsilon S_L$, $I = K$

$$X_{K,I} = m_I \, 1$$

5. $K \in S_L$, $I \notin S_{K-1}$

$$X_{K,I} = 0$$

6. $K \in S_R$, $I = NB1$

$$X_{K,I} = - \sum_{\lambda \in S_{K-1}} m_\lambda \Gamma_{K-1,\lambda}$$

7. $K \in S_L$, $I = NB1$

$$X_{K,I} = m_K \mathbf{1}$$

8. $K = NB1$, $I = NB1$

$$X_{K,I} = \sum_{\lambda \in S_0} m_\lambda \mathbf{1}$$

where

$$G^{\lambda}_{K-1,I-1} = m_\lambda \left[\left(\vec{\gamma}_{I-1,\lambda} \cdot \vec{\gamma}_{K-1,\lambda} \right) \mathbf{1} - \vec{\gamma}_{I-1,\lambda} \vec{\gamma}_{K-1,\lambda} \right]$$

$$\Gamma_{K-1,\lambda} = \mathscr{S}(\vec{\gamma}_{K-1,\lambda})$$

and

$\mathbf{1}$ = unit dyad

To conserve computer storage and eliminate a superfluous dimension, the integer function KT1, defined along with the other utility subroutines, is used to store the symmetric matrix of tensors in a three-dimensional array. That is

$$[XDIC(I, J, KI)]_{\ldots} = \left[X_{K,I} \right]_c$$

where

$$KI = KT1(NB1, K, I)$$

From symmetry

$$\left[X_{I,K}\right]_c = [\mathrm{XDIC}(I, J, KI)]^T$$

where post superscript T implies "transpose."

On the first pass through XDY, all elements of the arrays XDIC and GAM are computed. Thereafter, only those elements of the arrays that have time-varying components are recomputed. To do this, the integer arrays, defined by the code words SPI, SVD, and SXT, are used.

For any particular problem, all computation carried out in XDY can be outputted on the line printer by defining

LXDY = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE ETA

The primary purpose of subroutine ETA is to compute the forces and torques associated with the gyroscopic motion of each body and each nest of coupled bodies. The user need never interface with this subroutine.

On the right-hand side of equation 112 of Reference 2, the matrix

$$\{\eta_1\}$$

contains all gyroscopic forces and torques that enter into the coupled body equations of motion. The vector elements of this matrix are defined by the terms appearing in equations 107 and 108 of Reference 2. For the case of coupled rigid bodies, these terms reduce to those defined in Reference 1 by the partitions

$$\{\eta^c\} \text{ and } \{\eta^I\}$$

of equation 88.

Subroutine ETA is programmed to avoid redundant and trivial computations. The integer code words SCN, SCR, SIX, SMC, SFLX, SFXM, SSCN, and SSIX are used to pick out exactly which bodies contribute significantly in each step of the computation. Only nonzero time-varying contributions are computed.

The following variables are defined in table 1 and computed in ETA:

{ETIC(I,K)}	{FLCRC(I,K)}
{CNF(I,K)}	{ETMC(I,M)}
{FLIRC(I,K)}	{ETC(I,K)}

The gyroscopic torque acting on the nest K-1, which consists of at least one rigid body, is from Reference 2, equation 107

$$- \sum_{\lambda \epsilon S_{K-1}} \left[\mathring{\Phi}_\lambda \cdot \vec{\omega}_\lambda + \vec{\omega}_\lambda \times \vec{L}_{\lambda,\lambda} + \vec{\gamma}_{K-1,\lambda} \times \vec{C}_\lambda \right]$$

where

$$\vec{L}_{\lambda,\lambda} = \Phi_\lambda \cdot \vec{\omega}_\lambda + \Delta\vec{L}_{\lambda,\lambda} + \sum_{\substack{M: \\ MO(M)=\lambda}} \vec{\tilde{H}}_M$$

$$\Delta\vec{L}_{\lambda,\lambda} = \sum_{N,\lambda} \vec{Q}_{N,\lambda} \dot{a}_{N,\lambda}(t)$$

$$\mathring{\Phi}_\lambda = \sum_{N,\lambda} E_{N,\lambda} \dot{a}_{N,\lambda}(t)$$

$$\vec{C}_\lambda = m_\lambda \left[\sum_{\substack{i \epsilon S_{0,\lambda-1} \\ i \neq 1}} \left(\vec{\omega}_{J(i)} \times (\vec{\omega}_{J(i)} \times \vec{\beta}_i) + 2\vec{\omega}_{J(i)} \times \mathring{\vec{\beta}}_i \right) + \vec{\omega}_\lambda \times (\vec{\omega}_\lambda \times \vec{\alpha}_\lambda) + 2\vec{\omega}_\lambda \times \mathring{\vec{\alpha}}_\lambda \right]$$

(In N-BOD2 all $\mathring{\vec{\beta}}_i$ assumed zero)

The gyroscopic force acting on the nest L-1, which consists of the point mass defined as body L, is from Reference 2, equation 108

$$- \vec{C}_L$$

and from the same equation, the gyroscopic force acting on the composite system is

$$- \sum_{\lambda \epsilon S_0} \vec{C}_\lambda$$

These resultant quantities are computed and stored in the array ETC, so that

$\{ETC(I,K)\}$ = components of the gyroscopic load vector, in computing frame coordinates, acting on the nest K-1; K = 1, 2, . . ., NBOD.

$\{ETC(I,NBI)\}$ = components of the gyroscopic force vector, in computing frame coordinates, acting on the composite system.

The following procedure is used to compute the elements of the array ETC:

1. For all bodies L contained in the set defined by the code word SSIX, compute

$$\{ETIC(I,L)\} = \{\vec{\omega}_L \times (\Phi_L \cdot \vec{\omega}_L)\}_c$$

2. For all bodies L contained in the set defined by the code word SSCN, compute

$$\{CNF(I,L)\} = \left\{ m_L \left[\sum_{\substack{i \in S_{0,L-1} \\ i \neq 1}} \vec{\omega}_{J(i)} \times (\vec{\omega}_{J(i)} \times \vec{\beta}_i) + \vec{\omega}_L \times (\vec{\omega}_L \times \vec{\alpha}_L) \right] \right\}_c$$

3. For all flexible bodies L, that is, for those bodies contained in the set defined by the integer code word SFLX, compute

$$\{FLIRC(I,L)\} = \{\dot{\Phi}_L \cdot \vec{\omega}_L + \vec{\omega}_L \times \Delta\vec{L}_{L,L}\}_c$$

$$= \left\{ \left[\sum_{N,L} \left(E_{N,L} + \mathscr{P}(\vec{Q}_{N,L}) \right) \dot{a}_{N,L}(t) \right] \cdot \vec{\omega}_L \right\}_c$$

$$\{FLCRC(I,L)\} = \{2m_L \vec{\omega}_L \times \mathring{\vec{\alpha}}_L\}_c$$

$$= \left\{ 2m_L \vec{\omega}_L \times \sum_{N,L} \dot{a}_{N,L}(t) \vec{A}_{N,L} \right\}_c$$

4. Add gyroscopic loads associated with rigid body motion to that associated with relative elastic deformation to obtain

$$\{ETIC(I,L)\} = \{ETIC(I,L)\} + \{FLIRC(I,L)\}$$

$$\{CNF(I,L)\} = \{CNF(I,L)\} + \{FLCRC(I,L)\}$$

for each flexible body L.

5: For each symmetric wheel of the system, compute

$$\{ETMC(I,M)\} = \left\{\vec{\omega}_{MO(M)} \times \vec{\vec{H}}_M\right\}_c$$

6. Sum the gyroscopic loads associated with each body L contained in the nest K-1 to obtain the components of the gyroscopic load external to the nest K-1; K = 1, 2, . . . , NBOD+1, as follows:

   a. Initialize the array ETC:

$$\{ETC(I,K)\} = \{0\}; \qquad K = 1, 2, \ldots, NB1$$

   b. Add inertia cross-coupling terms for all bodies L contained in the set defined by the code word SIX(K-1):

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{L \epsilon SIX(K-1)} \{ETIC(I,L)\}$$

   where

$$K = 1, 2, \ldots, NBOD$$

   c. Add centripetal cross-coupling terms for all bodies L contained in the set defined by the code word SCN(K-1):

   - If body L is a rigid or flexible body

$$\{CNF(I,L)\} = \left\{\vec{C}_L\right\}_c$$

   - If body L is a point mass

$$\{CNF(I,L)\} = \left\{\vec{C}_L - 2m_L \vec{\omega}_L \times \mathring{\vec{\alpha}}_L\right\}_c$$

   Let

$$\{CNF(I,L)\} = \left\{\vec{C}_L^{\,*}\right\}_c$$

   then, for all nests K-1 containing one or more rigid bodies

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{L \epsilon SCN(K-1)} \left\{\vec{\gamma}_{K-1,L} \times \vec{C}_L^{\,*}\right\}_c$$

and for all nests K-1 containing the single-point mass labeled body K

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \left\{\vec{\gamma}_{K-1,K} \times \vec{C}_K^*\right\}_c$$

and for K = NB1

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{L=1}^{NBOD} \left\{\vec{C}_L^*\right\}_c$$

d. Add Coriolis cross-coupling terms for all point masses (bodies L) contained in the set defined by the code word SCR(K-1):

For all nests K-1 containing one or more rigid bodies

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{L \epsilon SCR(K-1)} \left\{\vec{\gamma}_{K-1,L} \times 2m_L \vec{\omega}_L \times \vec{\alpha}_L\right\}_c$$

For all nests K-1 containing the single-point mass labeled body K

$$\{ETC(I,K) = \{ETC(I,K)\} - \left\{2m_K \vec{\omega}_K \times \vec{\alpha}_K\right\}_c$$

and for K = NB1

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{L \epsilon SL} \left\{2m_L \vec{\omega}_L \times \vec{\alpha}_L\right\}_c$$

where SL is the code word defining all point-mass body labels.

e. Add symmetric wheel cross-coupling terms for all wheels M contained in the set defined by the code word SMC(K-1):

$$\{ETC(I,K)\} = \{ETC(I,K)\} - \sum_{M \epsilon SMC(K-1)} \{ETMC(I,M)\}$$

where

$$K = 1, 2, \ldots, NBOD$$

For any particular problem, all computation carried out in ETA can be outputted on the line printer by defining

LETA = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE TORQUE

The primary purpose of subroutine TORQUE is to provide a place for the user to define the nongyroscopic forces and torques acting on the particular problem under investigation. In Reference 1, pages 41 through 53, several nongyroscopic effects commonly encountered in the analysis of spacecraft attitude dynamics are described along with the methods that must be used to enter their resultant effects into the equations of motion. This section will define the coding (by example) that is required to interface with N-BOD2.

The following variables are defined in table 1 and computed by user defined algorithms in TORQUE:

$\{PHI(I,K)\}$

$\{CLM(M)\}$

NTQ

Y(N) N = NEQ+1, . . . , NEQ+NTQ

YD(N) N = NEQ+1, . . . , NEQ+NTQ

In subroutine TORQUE, the user has access to all computed system state variables. These are stored in the various labeled common blocks. Any user-defined function of state variables may be computed and stored in array DUMMY in labeled common block /SATELL/.

The nested body approach has been used to derive the equations of motion presented in References 1 and 2. In Reference 1, equation 88, the forces and torques acting external to the nests are contained in the partitions

$$\left\{\phi^{H_1}\right\} + \left\{\phi^{E_1}\right\}$$

while torques acting external to symmetric wheels are contained in the partition

$$\left\{\phi^{H_2}\right\}$$

In Reference 2, equation 111, the forces and torques acting external to the nests are contained in the partition

$$\left\{\phi^{(e)}\right\}$$

while torques acting external to symmetric wheels are contained in the partition

$$\{CL\}$$

The user must define the algorithm that is to be used to compute

$\{PHI(I,K)\}$ — resultant torque (force*) that is acting on and external to the nest K-1; K = 1, 2, . . ., NBOD

$\{PHI(I,NBI)\}$ — resultant sum of all external forces acting on the composite system of coupled bodies

CLM(M) — scalar magnitude of the torque acting external to the symmetric wheel M

Frequently the algorithm involves the integration of differential equations such as in the description of an onboard control system. The user must define

NTQ — total number of differential equations that are defined by the user in TORQUE and which must be integrated.

The state variables and associated time derivatives for all first order differential equations defined in TORQUE are stored in the arrays Y and YD.

Let

$$N = NEQ + M$$

where

$$M = 1, 2, \ldots, NTQ$$

then

Y(N) = storage location to the state variables associated with the M*th* differential equation defined by the user in TORQUE.

YD(N) = storage location of the time derivative of the state variable stored in Y(N).

To code subroutine TORQUE, the user should first zero out the external torque matrix upon entry. The resultant torque matrix is then obtained by vector addition of the various components. Thus, the first executable statements of TORQUE should be

```
C     ZERO ALL ELEMENTS OF EXTERNAL TORQUE MATRIX
      DO 1   K = 1, NB1
      DO 1    I = 1, 3
    1 PHI (I,K)  = 0.D0
      DO 2   M = 1, NMO
    2 CLM(M)   = 0.D0
```

*Force if body K in nest K-1 is a point mass.

Motion about or along any free coordinate vector may be constrained by springs, dashpots, or motors. The computation of the associated reaction loads and their inclusion into the equations of motion are discussed in Reference 1, pages 42 through 45. The following coding is to be used to include these effects in N-BOD2:

```
C                  REACTION TORQUE ACTING ACROSS OR ALONG GIMBAL AXIS M
C                    AT HINGE POINT K-1 DUE TO :
C                    LINEAR SPRINGS
C                    LINEAR VISCOUS DAMPERS
C                    MOTORS
C     LET:
C       SPR(M) = SPRING CONSTANT ABOUT OR ALONG GIMBAL AXIS M
C                 (M*L**2/T**2 OR M/T**2)
C       DPC(M) = DAMPING CONSTANT ABOUT OR ALONG GIMBAL AXIS M
C                 (M*L**2/T OR M/T)
C       CLT(M) = CONTROL TORQUE APPLIED BY MOTOR ABOUT OR ALONG GIMBAL
C                 AXIS M  M*L**2/T**2 OR M*L/T**2
C
C                       DIMENSION TEM(3)
C                  C    SPRING TORQUE
C                  C    SPR(M) = USER INPUT
C                       A = SPR(M)*THA(M)
C                       CALL SCLV(A,QFC(1,M),TEM)
C                       CALL VECSUB(PHI(1,K),TEM,PHI(1,K))
C                  C    DAMPER TORQUE
C                  C    DPC(M) = USER INPUT
C                       A = DPC(M)*THAD(M)
C                       CALL SCLV(A,QFC(1,M),TEM)
C                       CALL VECSUB(PHI(1,K),TEM,PHI(1,K))
C                  C    MOTOR TORQUE
C                  C    CLT(M) = FUNCTION OF STATE VARIABLES, USER DEF.
C                       CALL SCLV(CLT(M),QFC(1,M),TEM)
C                       CALL VECADD(PHI(1,K),TEM,PHI(1,K))
```

Symmetric wheels are usually used to simulate controlled momentum wheels. Since wheels have only one degree of freedom, it is not necessary to compute wheel torque as a vector quantity. The following coding is to be used to include this effect in N-BOD2:

```
C
C
C
C                  REACTION TORQUES ON SYSTEM DUE TO A CONTROL TORQUE
C                   APPLIED TO MOMENTUM WHEEL MW
C     LET:
C      CLM(MW) = CONTROL TORQUE APPLIED TO WHEEL MW ABOUT ITS SPIN AXIS
C                USER DEFINED FUNCTION OF STATE VARIABLES (M*L**2/T**2)
C
C                  C     CLM(MW) = USER DEFINED FUNCTION OF STATE VARB.
C
```

Locally applied forces may be applied to any body of the system; they may not be applied to symmetric wheels directly. The computation of the reaction loads and their inclusion into the equations of motion are discussed in Reference 1, page 45. The following coding is used to include this effect in N-BOD2:

```
C
C                  REACTION TORQUES ON SYSTEM DUE TO A LOCALLY
C                   APPLIED EXTERNAL FORCE (I.E. A GAS JET)
C     LET:
C           J = INTEGER LABEL ASSIGNED TO GAS JET
C           L = BODY TO WHICH EXTERNAL FORCE DIRECTLY APPLIED
C       RJ(I) = RADIUS VECTOR FROM CENTER OF MASS OF BODY L TO
C               GAS JET J, (COMPONENTS RELATIVE TO BODY L COORDINATES)
C       FJ(I) = COMPONENTS OF APPLIED FORCE DUE TO GAS JET J, (RELATVE
C               TO BODY L COORDINATES) USER DEFINED FUNCTION OF STATE
C               VARIABLES (M*L/T**2)
C
```

```
C              DIMENSION RJC(3),FJC(3),TEM(3),TEM1(3),RJ(3),FJ(3)
C              INTEGER S1(10),NS1
C              LOGICAL CTAIN
C        C     RJ(I) = USER INPUT
C              CALL VECTRN(RJ,XMC(1,1,L),RJC)
C        C     FJ(I) = USER DEFINED FUNCTION OF STATE VARIABLES
C              CALL VECTRN(FJ,XMC(1,1,L),FJC)
C              CALL VECADD(PHI(1,NB1),FJC,PHI(1,NB1))
C              DO 3  K=1,NBOD
C              CALL UNPAC(S1,NS1,SK(K-1))
C              IF(.NOT.CTAIN(L,S1,NS1)) GO TO 3
C              IF(RBLO(K)) GO TO 4
C              CALL VECADD(PHI(1,K),FJC,PHI(1,K))
C              GO TO 3
C            4 KL = KTO(NB1,K-1,L)
C              CALL VECADD(GAM(1,KL),RJC,TEM)
C              CALL VECROS (TEM,FJC,TEM1)
C              CALL VECADD(PHI(1,K),TEM1,PHI(1,K))
C            3 CONTINUE
C
C
```

The system to be modeled may be in a gravitational field; i.e., an Earth-based system. The computation of the reaction loads and their inclusion into the equations of motion are discussed in Reference 1, pages 45 and 46. The following coding is used to include this effect in N-BOD2:

```
C
C
C                    REACTION TORQUES ON SYSTEM DUE TO GRAVITATIONAL
C                     EFFECTS ON AN EARTH BASED SYSTEM
C     LET:
C         GRAV = ACCELERATION OF GRAVITY (L/T**2)
C        BH(I) = COMPONENTS OF UNIT VECTOR FROM INERTIAL ORIGIN TO COMP.
C                SYSTEM CENTER OF MASS, (RELATIVE TO INERTIAL FRAME)
C                THAT IS, PARALLEL TO DIRECTION OF GRAVITY FORCE
C
C              INTEGER S1(10),NS1
C              DIMENSION TEM(3),BHC(3),BH(3)
C        C     BH(I) = USER INPUT
C              CALL VECTRN(BH,XMC(1,1,0),BHC)
C              DO 4  K=1,NBOD
C        C     GRAV = USER INPUT
C              A = XMAS(K)*GRAV
C              CALL SCLV(A,BHC,TEM)
C              CALL VECSUB(PHI(1,NB1),TEM,PHI(1,NB1))
C              IF(RBLO(K)) GO TO 5
C              CALL VECSUB(PHI(1,K),TEM,PHI(1,K))
C              GO TO 4
C            5 CALL UNPAC(S1,NS1,SK(K-1))
C              DO 4  LL=1,NS1
C              L = S1(LL)
C              KL = KTO(NB1,K-1,L)
C              A = XMAS(L)*GRAV
C              CALL SCLV(A,BHC,TEM)
C              CALL VECROS (GAM(1,KL),TEM,TEM1)
C              CALL VECSUB(PHI(1,K),TEM1,PHI(1,K))
C            4 CONTINUE
C
C
```

The system to be modeled may be in orbit and subject to gravity gradient effects. The computation of the orbit and the gravity gradient torques are discussed in Reference 1, pages 46 through 52. The following coding is used to include these effects in N-BOD2:

```
C
C                    KEPLERIAN ORBIT
C     LET:
C      CB(I,0) = COMPONENTS OF VECTOR FROM EARTH'S CENTER TO COMPOSITE
C                SYSTEM CENTER OF MASS, RELATIVE TO INERTIAL REFERENCE
C                FRAME, ORBIT ASSUMED TO BE IN INERTIAL 2-3 PLANE
C           ASM = SEMI-MAJOR AXIS OF ELLIPTIC ORBIT INERTIAL 2 DIRECTION
C           ECC = ORBIT ECCENTRICITY
C           TPP = TIME OF PERIHELION PASSAGE, (T)
C           GEV = EARTH'S GRAVITATIONAL CONSTANT, (L**3/T**2)
C           ETE = MEAN MOTION
C           AME = MEAN ANOMALY
```

```
C          ECE = ECCENTRIC ANOMALY
C          TVE = TRUE ANOMALY
C          BTO = MAGNITUDE OF CB(I,0), (L)
C
C
C        C     ASM = USER INPUT
C        C     GEV = USER INPUT
C              ETE = 1./SQRT(ASM**3/GEV)
C        C     TPP = USER INPUT
C              AME = ETE*(T-TPP)
C              SM1 = SIN(AME)     ; SM4 = SIN(4*AME)
C              SM2 = SIN(2*AME)
C              SM3 = SIN(3*AME)
C        C     ECC = USER INPUT
C              ECE = AME + ECC*SM1 + ECC**2*SM2/2
C             *      +ECC**3*(9*SM3 - 3*SM1)/(24)
C             *      +ECC**4*(64*SM4 - 32*SM2)/192
C              CE = COS(ECE)
C              SE = SIN(ECE)
C              BTO = ASM*(1 - ECC*CE)
C              CB(1,0) = 0
C              CB(2,0) = BTO*((CE- ECC)/(1 - ECC*CE))
C              CB(3,0) = BTO*(SQRT(1-ECC**2)*SE/(1-ECC*CE))
C              CALL VECTRN(CB(1,0),XMC(1,1,0),CBC(1,0))
C
C
C
C
C                        REACTION TORQUES ON ORBITING SYSTEM DUE TO
C                             GRAVITY GRADIENT EFFECTS
C     LET:
C     CBC(I,1) = COMPONENTS OF VECTOR FROM COMPOSITE SYSTEM CENTER OF
C                MASS TO CENTER OF MASS OF BODY 1, RELATIVE TO
C                COMPUTING FRAME
C    * NOTE FOR GRAVITY GRADIENT OPTION
C    * CB(I,1) AND ITS INERTIAL DERIVATIVE
C    * ARE REDEFINED TO CIRCUMVENT DIFFERENCE
C    * OF LARGE NUMBER PROBLEMS, THAT IS
C    * THEY ARE MEASURED FROM COMPOSITE CM TO
C    * CM OF BODY 1 RATHER THAN FROM INERTIAL
C    * ORIGIN TO CM OF BODY 1
C        BH(I) = UNIT VECTOR FROM EARTH'S CENTER TO SYSTEM COMPOSITE
C                CENTER OF MASS, COMPONENTS RELATIVE TO INERTIAL FRAME
C     DEL(I,K) = COMPONENTS OF VECTOR FROM COMPOSITE SYSTEM CENTER OF
C                MASS TO CENTER OF MASS OF BODY K
C     DFG(I,K) = COMPONENTS OF GRAVITY GRADIENT FORCE ACTING ON BODY K
C       SGG(I) = COMPACTED INTEGER WORD, THOSE BODIES IN THE NEST I
C                WHICH SIGNIFICANTLY CONTRIBUTE TO GRAVITY GRADIENT EFF.
C           BTO = DISTANCE FROM EARTH'S CENTER TO COMPOSITE SYSTEM CM
C
C              DIMENSION DEL(3,10),DFG(3,10),BHC(3),BH(3),TEM(3),TEM1(3)
C              INTEGER SGG(0,9), S1(10)
C        C     KEPLERIAN ORBIT MUST BE USED WITH GRAVITY GRADIENT OPTION
C              DO 10  I=1,3
C           10 BHC(I) = CBC(I,0)/BTO
C              DO 7  K=1,NBOD
C              KL = KTO(NB1,0,K)
C              CALL VECADD(CBC(1,1),GAM(1,KL),DEL(1,K))
C              CALL VECDOT(BHC,DEL(1,K),A)
C              A = 3*A
C              CALL SCLV(A,BHC,TEM)
C              CALL VECSUB(DEL(1,K),TEM,TEM)
C              A = -GEV*XMAS(K)/BTO**3
C              CALL SCLV(A,TEM,DFG(1,K))
C            7 CONTINUE
C              DO 8  K=1,NBOD
C              IF(RBLO(K)) GO TO 9
C              CALL VECADD(PHI(1,K),DFG(1,K),PHI(1,K))
C              GO TO 8
C        C     SGG(I) = SK(1) IF ALL BODIES CONTRIBUTE TO GRAVITY GRAD.
C        C                    EFFECTS. IF NOT USE COMPAC TO CONSTRUCT
C        C                    SGG(I) FORM USER INPUT OR DEFINE DIRECTLY
C            9 CALL UNPAC(S1,NS1,SGG(K-1))
C              DO 8  LL=1,NS1
C              L = S1(LL)
C              KL = KTO(NB1,K-1,L)
C              CALL VECROS (GAM(1,KL),DFG(1,L),TEM)
C              CALL VXDYOV(BHC,XIC(1,1,L),TEM1)
C              A = 3*GEV/BTO**3
C              CALL SCLV(A,TEM1,TEM1)
C              CALL VECADD(TEM,TEM1,TEM)
C              CALL VECADD(PHI(1,K),TEM,PHI(1,K))
C            8 CONTINUE
C
```

As previously mentioned, it is often necessary to integrate several differential equations to define control torques in TORQUE. The following coding is used to include additional differential equations for integration by N-BOD2 along with the system equations:

```
C
C                  PARAMETERS DEFINED BY FIRST ORDER
C                    DIFFERENTIAL EQUATIONS
C     LET:
C          NTQ = TOTAL NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS TO
C                BE SOLVED FOR USE IN SUBROUTINE TORQUE
C        TQ(N) = MAGNITUDE OF PARAMETER NUMBER N DEFINED WITHIN SUB.
C                TORQUE AT TIME T
C       TQD(N) = TIME DERIVATIVE OF PARAMETER TQ(N). A USER DEFINED
C                FUNCTION OF THE SYSTEM'S STATE VARIABLES
C
C                      DIMENSION TQ(20),TQD(20)
C                C     FOR THE PARAMETER N
C                      IF(CT4.NE.1) GO TO 11
C                C     Y(NEQ+N) = TQ(N) = INITIAL VALUE FOR TQ(N)
C                   11 TQ(N) = Y(NEQ+N)
C                C     TQD(N) = USER DEFINED FUNCTION OF STATE VARB.
C                C
C                C     AFTER DEFINITION OF LAST DIFFERENTIAL EQUATION
C                C     NTQ = TOTAL NUMBER OF FIRST ORDER DIFFERENTIAL
C                C           EQUATIONS TO BE SOLVED FOR USE IN TORQUE
C                      DO 12  N=1,NTQ
C                   12 YD(NEQ+N) = TQD(N)
C
```

At times, appendages on spacecraft are subject to thermally induced motion. This is discussed in Reference 1, pages 52 and 53. The following coding is used to include this effect in N-BOD2:

```
C
C                  THERMALLY INDUCED MOTION ABOUT GIMBAL AXIS M
C                    AT HINGE POINT K-1
C     ASSUME:
C          ALL THERMALLY INDUCED DEFLECTION IS SMALL ANGLE
C             RELATIVE TO THE SYSTEM'S NOMINAL ZERO STRESS STATE
C          THERMALLY INDUCED DEFLECTION IS MODELLED AS A MOVEMENT
C             OF THE ZERO STRESS STATE
C          ACROSS ALL HINGE POINTS SUBJECT TO THERMAL DEFORMATION
C             SPRINGS AND DAMPERS ACT
C          A RESONABLE MODEL OF THE THERMAL INPUT CAN BE DEFINED
C             IN TERMS OF THE SYSTEM'S STATE VARIABLES
C          THERMAL EQUILIBRIUM POSITION ABOUT ANY GIMBAL AXIS IS DEF.
C             BY SOLUTION OF THE HEAT CONDUCTION EQUATION
C     LET:
C        SPR(M) = SPRING CONSTANT ACROSS GIMBAL AXIS M
C        DPC(M) = DAMPING CONSTANT ACROSS GIMBAL AXIS M
C        TAU(M) = THERMAL TIME CONSTANT FOR DEFORMATION ABOUT GIMBAL
C                 AXIS M, (T)
C         TQ(N) = THERMAL EQUILIBRIUM POSITION FOR THERMAL DEFORMATION
C                 ABOUT GIMBAL AXIS M, (RAD)
C       TQD(N) = RATE OF CHANGE OF THERMAL EQUILIBRIUM POSITION ABOUT
C                 GIMBAL AXIS M, FIRST ORDER DIFF. EQ., (RAD/T)
C        TINP = THERMAL INPUT USER DEFINED FUNCTION OF STATE VARIABLES,
C               (RAD/T)
C
C              DIMENSION TEM(3)
C        C     N = USER DEFINED LABEL, DEPENDS UPON EQUATION NUMBERING
C        C       SEQUENCE DEFINED WITHIN SUBROUTINE TORQUE
C              IF(CT4.NE.1) GO TO 13
C        C     Y(NEQ+N) = TQ(N) = INITIAL VALUE FOR THERMAL DEFORMATION
C        C                        ABOUT GIMBAL AXIS M, USER INPUT
C           13 TQ(N) = Y(NEQ+N)
C        C     TINP = USER DEFINED THERMAL INPUT FOR THERMAL DEFORMATION
C        C            ABOUT GIMBAL AXIS M
C        C     TAU(M) = USER INPUT
C              TQD(N) = -TQ(N)/TAU(M) + TINP
C              A = SPR(M)*(THA(M) - TQ(N))
C              CALL SCLV(A,OFC(1,M),TEM)
C              CALL VECSUB(PHI(1,K),TEM,PHI(1,K))
C              A = DPC(M)*THAD(M)
C              CALL SCLV(A,OFC(1,M),TEM)
C              CALL VECSUB(PHI(1,K),TEM,PHI(1,K))
C
```

For any particular problem, all computation carried out in TORQUE can be outputted on the line printer by defining

LTORQU = .TRUE.

on card 1 of the data deck, if the user codes the print statements.

PROGRAMMER'S GUIDE TO SUBROUTINE QFDOT

The primary purpose of subroutine QFDOT is to eliminate the forces and torques associated with kinematic constraints. In so doing, the coupled vector-dyadic equations of motion will reduce to a set of simultaneous scalar differential equations. The user need never interface with this subroutine.

The equations of motion to be integrated are given in Reference 2 by equation 136. This equation, in the absence of body flexibility, reduces to equation 144 of Reference 1. Making use of the array names used in the coding of N-BOD2, the equation takes on the general form

$$[\text{XMN}]\ \{\text{THADD}\} = \{\text{ETM}\}$$

where, to be consistent with the coding of N-BOD2 and the notation of Reference 2,

$$\{\text{THADD}\} = \begin{Bmatrix} \ddot{\theta} \\ \ddot{a} \\ \ddot{\theta}_W \end{Bmatrix}$$

This matrix equation defines exactly

NFER + NMODS + NMV

scalar second-order ordinary differential equations.

The following variables are defined in table 1 and computed in QFDOT:

$$[\text{XMN (M,N)}]$$

$$\{\text{ETM (M)}\}$$

The elements of these arrays are obtained by carrying out the vector-dyadic operations defined by the partitioned matrix equation 136 of Reference 2.

The equations coded in N-BOD2 assume that all hinge points and symmetric wheel attachment points are either on rigid bodies or at node points of flexible bodies. The restriction in modeling capability implies that in equation 136 of Reference 2, the following is true:

$$\varphi_H{}^R = 0,$$

$$\varphi_W{}^R = 0, \text{ and}$$

$$\varphi_H{}^T = 0$$

Furthermore, it is assumed in N-BOD2 that the user will not need to compute forces of constraint; that is, the elements of the partition F^c.

Introduction of these two modeling limitations significantly simplifies the required coding. The coding is structured so that, if required at a later date, both limitations can be removed by simple insertions into QFDOT and other relevant subroutines. Making use of the above assumptions, the equations coded in N-BOD2 are from equation 136 of Reference 2.

$$\begin{bmatrix} q^T & 0 & 0 \\ 0 & h^T & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X & I^c & F \\ I^{c\,T} & I^s & 0 \\ F^T & 0 & M^1 \end{bmatrix} \cdot \begin{bmatrix} q & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{\theta} \\ \ddot{\theta}_W \\ \ddot{a} \end{Bmatrix} =$$

$$\begin{bmatrix} q^T & 0 & 0 \\ 0 & h^T & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left\{ - \begin{bmatrix} X & I^c & F \\ I^{c\,T} & I^s & 0 \\ F^T & 0 & M^1 \end{bmatrix} \cdot \begin{bmatrix} \dot{q} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{\theta} \\ \dot{\theta}_W \\ \dot{a} \end{Bmatrix} + \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{Bmatrix} + \begin{Bmatrix} \phi^{(e)} \\ CL \\ 0 \end{Bmatrix} \right\}$$

Coding requirements force a rearrangement of rows and columns from that used in the theoretical development. The following storage allocation scheme is used:

- $[XMN(M, N)] = [q^T] \cdot [X] \cdot [q]$

  where $M = 1, \ldots, NFER$

  $N = 1, \ldots, NFER$

- $[XMN(M, N)] = [q^T] \cdot [F]$

  where $M = 1, \ldots, NFER$

  $N = NFER+1, \ldots, NFER+NMODS$

- $[XMN(M, N)] = [q^T] \cdot [I^c] \cdot [h]$

  where $M = 1, \ldots, NFER$

  $N = NFER+NMODS+1, \ldots, NFER+NMODS+NMV$

- $[XMN(M, N)] = [M^1]$

  where $M = NFER+1, \ldots, NFER+NMODS$

  $N = NFER+1, \ldots, NFER+NMODS$

- $[XMN(M, N)] = [0]$

  where $M = NFER+1, \ldots, NFER+NMODS$

  $N = NFER+NMODS+1, \ldots, NFER+NMODS+NMV$

- $[XMN(M, N)] = [h^T] [I^s] [h]$

  where $M = NFER+NMODS+1, \ldots, NFER+NMODS+NMV$

  $N = NFER+NMODS+1, \ldots, NFER+NMODS+NMV$

- $\{ETM(M)\} = [q^T] \cdot \left\{ \{\eta_1\} + \{\phi^{(e)}\} - [X] \cdot [\dot{q}] \{\dot{\theta}\} \right\}$

  where $M = 1, 2, \ldots, NFER$

- $\{ETM(M)\} = \{\eta_3\} - [F^T][\dot{q}]\{\dot{\theta}\}$

  where $M = NFER+1, \ldots, NFER+NMODS$

- $\{ETM(M)\} = [h^T] \cdot \left\{\{\eta_2\} + \{CL\} - [I^{cT}][\dot{q}]\{\dot{\theta}\}\right\}$

  where $M = NFER+NMODS+1, \ldots, NFER+NMODS+NMV$

In the computation of the ETM array, the contents of the ETC array defined in subroutine ETA are changed. In ETA, the components of the vector in row K of $\{\eta_1\}$ are stored in $\{ETC(I,K)\}$. In QFDOT, the components of the vector in row K of

$$\{\eta_1\} + \{\phi^{(e)}\} - [X] \cdot [\dot{q}]\{\dot{\theta}\}$$

are stored in $\{ETC(I,K)\}$.

In the computation of the full XMN array only the nontrivial entires of XMN are computed, and the lower triangular portion of the array is obtained by making use of the fact that the array must be symmetric. To ensure that all redundant and trivial operations are circumvented, the labels stored in the integer code words

SI, SK, SR, SCXC, SMV, SOK, SQF,
JCON, SFCC, SFLX, and SFXM

are used extensively.

For any particular problem, all computation carried out in QFDOT can be outputted on the line printer by defining

LQFDOT = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE DCT

The primary purpose of subroutine DCT is to set up the differential equations required to compute the elements of the transformation matrices that are to be obtained by direction cosine techniques. The user need never interface with this subroutine.

In Reference 1, equation 119, the direction cosine equation is given in a form compatible with this coupled body analysis; namely

$$[{}_c\dot{\mathscr{T}}_\lambda] = -\mathscr{S}\left(\{{}_c\vec{\omega}_\lambda\}_c\right)[{}_c\mathscr{T}_\lambda]$$

N-BOD2 is coded to integrate only six of the nine equations defined for each transformation matrix that is to be derived by direction cosine techniques. The other three parameters of the respective transformation matrices are obtained in TRAN from orthogonality conditions.

The following array is defined in table 1 and the elements of it are computed in DCT:

YMCD (I,J,K).

In DCT, a unique numbering sequence is used to facilitate the handling of the direction cosine data. The body labels stored in the integer code word SD are obtained first. Let

$\lambda_1, \lambda_2, \ldots, \lambda_N$ = body labels associated with those body fixed-reference frames for which transformation matrices are to be obtained by direction cosine techniques

where

$$\lambda_1 > \lambda_2 > \ldots > \lambda_N$$

If $\lambda_N = 1$ and the body 1 fixed frame is the frame of computation, then to obtain the transformation matrix between inertial and body 1 coordinates,

$$\{\text{YMCD (I, 1, 1)}\} = \left[{}_c\dot{\mathscr{T}}_0\right]\begin{Bmatrix}1\\0\\0\end{Bmatrix}$$

$$\{\text{YMCD (I, 2, 1)}\} = \left[{}_c\dot{\mathscr{T}}_0\right]\begin{Bmatrix}0\\1\\0\end{Bmatrix}$$

and for the rest, M = 1, 2, . . . , N-1

$$\{YMCD(I,1,M+1)\} = \left[{}_c\dot{\mathscr{T}}_{\lambda_M}\right] \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$\{YMCD(I,2,M+1)\} = \left[{}_c\dot{\mathscr{T}}_{\lambda_M}\right] \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

Otherwise, for M = 1, 2, . . . , N,

$$\{YMCD(I,1,M)\} = \left[{}_c\dot{\mathscr{T}}_{\lambda_M}\right] \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$\{YMCD(I,2,M)\} = \left[{}_c\dot{\mathscr{T}}_{\lambda_M}\right] \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

For any particular problem, all computation carried in DCT can be outputted on the line printer by defining

LDCT = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE ANGLE

The primary purpose of subroutine ANGLE is to set up the differential equations required to compute the displacement coordinates that are measured about or along the free coordinate vectors. The user need never interface with this subroutine.

The following array is defined in table 1 and the elements of it are computed in ANGLE:

ANGD(N) – time derivative of the displacement coordinate associated with the motion about or along the free coordinate vector SFR(N); N = 1, 2, . . . , NFRC

In Reference 1, pages 37 and 38, the differential equations which define relative displacement about or along free coordinate vectors are defined. These equations are coded in

ANGLE. The user has the capability by subroutine INOPT to select which displacement coordinate should be evaluated. The integer array SFR defines the selected coordinates.

Several special cases are recognized in ANGLE:

- If relative translational motion is measured along free coordinate vector

$$M = SFR(N)$$

then

$$ANGD(N) = THAD(M) = \dot{\theta}_M$$

(Point mass motion or body 1 center-of-mass motion.)

- If free coordinate vector

$$M = SFR(N)$$

is aligned with the gimbal axis of a one-axis gimbal, then

$$ANGD(N) = THAD(M) = \dot{\theta}_M$$

- If the free coordinate vectors

$$M = SFR(N) \quad \text{and}$$

$$M+1 = SFR(N+1)$$

are aligned with the gimbal axes of a two-axis gimbal at hinge point K-1, then

$$C = \vec{q}_M \cdot \vec{q}_{M+1}$$

$$A_1 = \vec{q}_M \cdot \vec{\tilde{\omega}}_K$$

$$A_3 = \vec{q}_{M+1} \cdot \vec{\tilde{\omega}}_K$$

and

$$ANGD(N) = \frac{A_1 - C*A_3}{1 - C^2}$$

$$ANGD(N+1) = \frac{A_3 - C*A_1}{1 - C^2}$$

- If the free coordinate vectors

$$M + i = SFR(N + i) \qquad i = 0, 1, 2$$

are aligned with the gimbal axes of a three-axis gimbal at hinge point K-1, then

$$C = \vec{q}_M \cdot \vec{q}_{M+2}$$

$$A_1 = \vec{q}_M \cdot \vec{\tilde{\omega}}_K$$

$$A_2 = \vec{q}_{M+1} \cdot \vec{\tilde{\omega}}_K$$

$$A_3 = \vec{q}_{M+2} \cdot \vec{\tilde{\omega}}_K$$

and

$$ANGD(N) = \frac{A_1 - C * A_3}{1 - C^2}$$

$$ANGD(N + 1) = A_2$$

$$ANGD(N + 2) = \frac{A_3 - C * A_1}{1 - C^2}$$

For any particular problem, all computations carried out in ANGLE can be outputted on the line printer by defining

LANGLE = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINE SETUP

The primary purpose of SETUP is to provide the necessary interface between the symbolic array names used in the derivation of the equations of motion in N-BOD2 and the symbolic array names used in the integration subroutine RUNGE. The user need never interface with this subroutine.

The following sequential ordering scheme, used to define the locations of state variables and their time derivatives in the Y and YD arrays, is identical to that used in EQIV:

- Accelerations and rates relative to free coordinate vectors

  YD (N) = THADD (N) ; THAD (N) = Y (N)

  N = 1, 2, . . . , NFER

  Sum equations: NEQ = NFER

- Accelerations and rates of generalized elastic coordinates

  YD (N) = THADD (N) ; THAD (N) = Y (N)

  N = NEQ+1, . . . , NEQ+NMODS

  Sum equations: NEQ = NEQ+NMODS

- Accelerations and rates of symmetric wheels with variable rate. Let

  $M_1, M_2, \ldots, M_L$ = wheel labels in the set defined by integer code word SMV

  YD (N) = THADD (N) ; THADW(M_1) = Y(N)

  N = NEQ+1, . . . , NEQ+L

  I = 1, 2, . . . , L

  Sum equations: NEQ = NEQ+L

- Rate and displacement relative to free coordinate vectors

  YD (N) = ANGD (MM) ; THA (M) = Y (N)

  N = NEQ+1, . . . , NEQ+NFRC

  MM = 1, 2, . . . , NFRC

  M = SFR (MM)

  Sum equations: NEQ = NEQ+NFRC

- Rate and displacement of generalized elastic coordinates

  YD (N) = THAD (M) ; THA (M) = Y (N)

  N = NEQ+1, . . . , NEQ+NMODS

  M = 1, 2, . . . , NMODS

  Sum equations: NEQ = NEQ+NMODS

- Rate and angle of symmetric wheels

  YD (N) = THADW (M) ; THAW (M) = Y (N)

  N = NEQ+1, . . . , NEQ+NMOA

  MM = 1, 2, . . . , NMOA

  M = SMA (MM)

  Sum equations: NEQ = NEQ+NMODS

- Rate and magnitude of direction cosine

  $K_1, K_2, \ldots, K_{ID}$ = body labels associated with the coordinate frames for which transformation matrix is to be obtained via direction cosine

  YD (N) = YMCD (I, J, M)

  M = 1, 2, . . . , ID

  J = 1, 2

  I = 1, 2, 3

  N = NEQ + 6 * (M - 1) + 3 * (J - 1) + I

  XMC (I, J, K_M) = Y (N)

  I = 1, 2, 3

  J = 1, 2

  M = 1, 2, . . . , ID

The exact algorithm is obtainable in the discussion of subroutine DCT.

For any particular problem, all computations carried out in SETUP can be outputted on the line printer by defining

LSETUP = .TRUE.

on card 1 of the data deck.

PROGRAMMER'S GUIDE TO SUBROUTINES OUTPUT AND OUTPSP

The primary purpose of subroutines OUTPUT and OUTPSP is to provide a convenient place to put all output data statements. OUTPUT is entered only at time zero while OUTPSP is entered at the end of every integration step unless otherwise directed by the user. The user is expected to refrain from modifying subroutine OUTPUT. It provides a useful echo of the initial state of the system for checkout purposes..

Nominally, subroutine OUTPSP is identical to OUTPUT. The user is expected to remove unwanted print statements from OUTPSP, add additional computation if desired, and insert any other set of desirable output statements.

The following parameters are outputted on the line printer by subroutine OUTPUT:

TIME – simulation time t

CENTER OF MASS – components of the vector from hinge point 0 to the center-of-mass of the composite system relative to the frame of computation

TOTAL SYSTEM MASS – total mass of the system

SYSTEM INERTIA TENSOR – components of the total system inertia tensor relative to the composite system center-of-mass and relative to the frame of computation

ANGULAR MOMENTUM – magnitude of the total system inertial angular momentum vector

HBODY – components of the total system inertial angular momentum vector relative to body 1 fixed coordinates

HINERT – components of the total system inertial angular momentum vector relative to inertial coordinates

LINEAR MOMENTUM – magnitude of the total system inertial linear momentum vector

LBODY – components of the total system inertial linear momentum vector in computing frame coordinates

KINETIC ENERGY – kinetic energy of total system

After the above parameters are printed, a block of data for each body of the system is printed:

BODY K – body label for the body for which the adjacent block of data applies; K = 1, 2, . . . , NBOD

$$\text{ROMC} = \{\text{ROMC}(I,K)\} = \left\{\vec{\tilde{\omega}}_K\right\}_c$$

$$\text{FOMC} = \{\text{FOMC}(I,K)\} = \left\{\vec{\omega}_K\right\}_c$$

ACC = not in common = $\left\{\dot{\vec{\omega}}_K\right\}_c$

ETC = $\{ETC(I,K)\}$ = see QFDOT, also page 83

PHI = $\{PHI(I,K)\}$ = see TORQUE

CAC = $\{CAC(I,K)\}$ = $\left\{\vec{\alpha}_K\right\}_c$

CBC = $\{CBC(I,K)\}$ = $\left\{\vec{\beta}_K\right\}_c$

POS = not in common = $\left\{\vec{\gamma}_{I,K}\right\}_c$

VEL = not in common = $\left\{\dot{\vec{\gamma}}_{I,K}\right\}_c$

XMC = $[XMC(I,J,K)] = [{}_c\mathscr{T}_K]$

XIC = $[XIC(I,J,K)] = [\Phi_K]_c$

HB = not in common = $\left\{\vec{L}_{I,K}\right\}_c$

LM = not in common = $\left\{\vec{G}_{I,K}\right\}_c$

TK = not in common

$$= \frac{1}{2}\left[\vec{\omega}_K \cdot (\Phi_\lambda \cdot \vec{\omega}_K + m_K \vec{\gamma}_{I,K} \times \dot{\vec{\gamma}}_{I,K}) + m_K \dot{\vec{\gamma}}_{I,K} \cdot \dot{\vec{\gamma}}_{I,K} + \sum_{\substack{M: \\ MO(M)=K}} \vec{\vec{\omega}}_{W_M} \cdot \vec{\vec{H}}_M\right]$$

The block of data pertinent to the translational motion of the center-of-mass of body 1 is given adjacent to the title, "ORIGIN."

$$\text{FOMC} = \{\text{FOMC (I, NB1)}\} = \{\dot{\vec{\beta}}_1\}_c$$

$$\text{ACC} = \text{not in common} = \{\ddot{\vec{\beta}}_1\}_c$$

$$\text{CBC} = \{\text{CBC (I, 0)}\} = \{\vec{\beta}_0\}_c$$

$$\text{ETC} = \{\text{ETC (I, NB1)}\} = \text{See QFDOT}$$

$$\text{PHI} = \{\text{PHI (I, NB1)}\} = \text{See TORQUE}$$

$$\text{XMC} = \{\text{XMC (I, J, 0)}\} = [{}_c\mathscr{T}_0]$$

For each symmetric wheel of the system, the following data are outputted:

$$\text{HMOM (M)} = |\vec{\tilde{H}}_M|$$

$$\text{CLM (M)} = |\vec{CL}_M|$$

For each flexible body of the system, a block of data associated with its flexible body characteristics is outputted:

FLEXIBLE BODY K – body label for the flexible body for which the adjacent block of data applies

$$\text{EP} = \left\{\vec{\Delta\alpha}_K\right\}_K = \sum_{N,K} \left\{\vec{A}_{N,K}\right\}_K a_{N,K}(t)$$

$$\text{EPD} = \left\{\mathring{\vec{\alpha}}_K\right\}_c = \sum_{N,K} \left\{\vec{A}_{N,K}\right\}_c \dot{a}_{N,K}(t)$$

$$\text{QD} = \sum_{N,K} \left\{\vec{Q}_{N,K}\right\}_K \dot{a}_{N,K}(t)$$

$$\text{EI} = \sum_{N,K} [E_{N,K}]_K a_{N,K}(t) = [\Delta\Phi]_K$$

$$\text{EID} = \sum_{N,K} [E_{N,K}]_K \dot{a}_{N,K}(t) = [\mathring{\Phi}_K]_K$$

$$\text{FIR} = \{\text{FLIRC (I, K)}\} = \text{See ETA}$$

$$FCR = \{FLCRC(I,K)\} = \text{See ETA}$$

$$DHM = \sum_{N,K} \{Q_{N,K}\}_c \dot{a}_{N,K}(t) = \{\vec{\Delta L}_{\lambda,\lambda}\}_c$$

Following the last block of flexible body data, the system state variables associated with relative body motion and flexible body motion are printed:

For M = 1, 2, . . . , NFER

$$THA(M) = \theta_M(t)$$

$$THAD(M) = \dot{\theta}_M(t)$$

$$THADD(M) = \ddot{\theta}_M(t)$$

$$QFC(M) = \{QFC(I,M)\}_c = \{\vec{q}_M\}_c$$

For MN = 1, 2, . . . , NMODS and M = NFER+MN

$$THA(M) = a_{N,K}(t)$$

$$THAD(M) = \dot{a}_{N,K}(t)$$

$$THADD(M) = \ddot{a}_{N,K}(t)$$

where

$$MN = N + \sum_{I=1}^{K-1} SFXM(I)$$

K = 2, 3, . . . , NBOD

N = 1, 2, . . . , SFXM(K) [See INOPT, subsection 15]

All of the above data will be printed at the end of each integration step. The user is expected to delete the undesired print statements from OUTPSP and to insert those more applicable to the problem at hand.

PROGRAMMER'S GUIDE TO THE UTILITY ROUTINES

The utility routines required by N-BOD2 are:

SIMQ – used to obtain the solution to a set of simultaneous linear equations of the form

$$[X]\{\ddot{\theta}\} = \{\eta\}$$

A standard reduction technique is used to solve for $\{\ddot{\theta}\}$ without ever obtaining the inverse of the coefficient matrix.

RUNGE – used to numerically integrate the set of simultaneous nonlinear differential equations,

$$\{\dot{y}(t)\} = \{F(t)\}$$

Fourth order, fixed-step Runge-Kutta integration is used.

COMPAC – used to create integer code words.

Integer code words are used primarily to conserve computer storage. This was a critical factor on the small XDS-9300 computer on which N-BOD2 was originally debugged.

Let

$$(I_1, I_2, \ldots, I_N)$$

be a set of unique integer labels so that

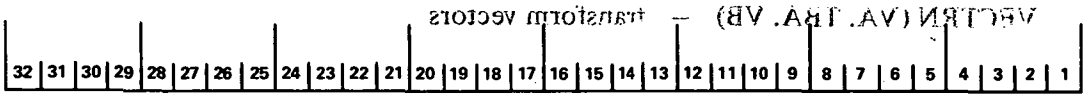
$$1 \leqslant I_i \leqslant 24 \quad ; \quad i = 1,2,\ldots,N$$

and

$$N \leqslant 24$$

If computer storage is limited, it is a waste of computer storage to reserve 24 words of memory to store this set.

All integer code words are 32 bits long.* These bits may be numbered as shown below:



*IBM 360 Computer, subroutine must be modified for computers having different word lengths.

COMPAC stores the number of labels (N; $N \leq 24$) in bits 25 through 32. The set of integers $I_1, I_2, \ldots, I_N$ are stored in bits 1 through 24. For $J = 1, 2, \ldots, 24$, if bit $J = 1$, J is a member of the set; if bit $J = 0$, J is not a member of the set.

UNPAC — used to decode the integer code words created by COMPAC

KT0 and KT1 — integer function subprograms used to assign storage area locations in one-dimensional arrays for elements of triangular or symmetric matrices; another tool used to conserve storage requirements:

$$KT0(N, J, K) = \begin{cases} K(N-1) + J + 1 - K(K-1)/2 & K < J \\ J(N-1) + K + 1 - J(J-1)/2 & K \geq J \end{cases}$$

$$J = 0, 1, \ldots, N-1$$

$$K = 0, 1, \ldots, N-1$$

$$KT1(N, J, K) = \begin{cases} (K-1)(N-1) + J - (K-1)(K-2)/2 & K < J \\ (J-1)(N-1) + K - (J-1)(J-2)/2 & K \geq J \end{cases}$$

$$J = 1, 2, \ldots, N$$

$$K = 1, 2, \ldots, N$$

CTAIN — logical function subprogram used to determine if a particular integer is contained in a particular set of integers

CTAIN (I, S, N) = .TRUE. if the integer I is contained in the set of integers S (1), S (2), . . . , S (N)

= .FALSE. if it is not

VECTRN (VA, TBA, VB) — transform vectors

$$\{\vec{v}\}_B = [{}_B\mathscr{T}_A]\,\{\vec{v}\}_A$$

TENTRN (XA, TBA, XB) — transform dyads

$$[X]_B = [{}_B\mathscr{T}_A]\,[X]_A\,[{}_B\mathscr{T}_A]^T$$

VECNRM (V) – normalize vector to unit magnitude

$$\vec{\omega} = \frac{\vec{v}}{|\vec{v} \cdot \vec{v}|^{1/2}}$$

$$\vec{v} = \vec{\omega}$$

Error message if $\vec{v} \cdot \vec{v} = 0$; inputted vector destroyed

MATMUL (A, B, C, N) – multiply two N X N matrices of scalars

$$[C] = [A]\,[B]$$

TRNSPS (TBA) – transpose a 3 X 3 matrix of scalars

$$[TEM] = [TBA]^T$$

$$[TBA] = [TEM]$$

Inputted matrix destroyed

ROT (A, J, T) – forms transformation matrix for rotation about a coordinate axis

A – sine of the rotation angle

J – axis about which positive rotation is measured; J = +1, -1, +2, -2, +3 or -3

Define:

$$JJ = |J|$$

$$V_1 = 0$$

$$V_2 = 0$$

$$V_3 = 0$$

Redefine:

$$V_{JJ} = A*J/JJ$$

Then:

$$[T] = \begin{bmatrix} \sqrt{1 - V_2^2 - V_3^2} & -V_3 & V_2 \\ V_3 & \sqrt{1 - V_1^2 - V_3^2} & -V_1 \\ -V_2 & V_1 & \sqrt{1 - V_1^2 - V_2^2} \end{bmatrix}$$

VECADD (V1, V2, S) – add vectors

$$\vec{S} = \vec{v}_1 + \vec{v}_2$$

VECSUB (V1, V2, D) – subtract vectors

$$\vec{D} = \vec{v}_1 - \vec{v}_2$$

SCLV (SC, V, P) – multiply vector by a scalar

$$\vec{P} = SC * \vec{v}$$

VECDOT (V1, V2, D) – vector dot product

$$D = \vec{v}_1 \cdot \vec{v}_2$$

VECROS (V1, V2, C) – vector cross product

$$\vec{C} = \vec{v}_1 \times \vec{v}_2$$

TRIPVP (V1, V2, V) – special triple vector product

$$\vec{v} = \vec{v}_1 \times (\vec{v}_1 \times \vec{v}_2)$$

DYADD (D1, D2, D) – add dyads

$$\mathbf{D} = \mathbf{D}_1 + \mathbf{D}_2$$

SCLD (A, D, T) – multiply dyad by a scalar

$$\mathbf{T} = A * \mathbf{D}$$

DYDOTV (A, V, D) — dyad dot vector

$$\vec{D} = \mathbf{A} \cdot \vec{v}$$

VXDYOV (V1, D, V) — special vector cross dyad dot vector

$$\vec{v} = \vec{v}_1 \times (\mathbf{D} \cdot \vec{v}_1)$$

DYTOV (D, V1, V) — dyad transpose dot vector

$$\vec{v} = \mathbf{D}^T \cdot \vec{v}_1$$

VODYOV (V1, D, V2, SC) — vector dot dyad dot vector equals scalar

$$SC = \vec{v}_1 \cdot (\mathbf{D} \cdot \vec{v}_2)$$

DYOP (V, D) — skew operator

$$\mathbf{D} = \mathscr{P}(\vec{v})$$

$$= \begin{bmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{bmatrix}$$

SUEOP (V1, V2, XM, D) — construct pseudo-inertia tensors

$$\mathbf{D} = m\,[(\vec{v}_1 \cdot \vec{v}_2)\,\mathbf{1} - \vec{v}_2\,\vec{v}_1]$$

QUTMUL (Q1, Q2, P) — multiply quaternions

$$\bar{p} = \bar{Q}_1 * \bar{Q}_2$$

QUATOP (QF, THA, ZT) — construct rotation quaternion

$$\theta = \text{THA}$$

$$\{\vec{QF}\} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix}$$

$$\{\bar{ZT}\} = \begin{Bmatrix} \cos \theta/2 \\ q_1 \sin \theta/2 \\ q_2 \sin \theta/2 \\ q_3 \sin \theta/2 \end{Bmatrix}$$

TRANSO (ZT, TAU) — construct transformation matrix from quaternion component

$$\{\bar{ZT}\} = \begin{Bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{Bmatrix}$$

$$[TAU] = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_0 e_3 + e_1 e_2) & 2(e_1 e_3 - e_0 e_2) \\ 2(e_1 e_2 - e_0 e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2 e_3 + e_0 e_1) \\ 2(e_1 e_3 + e_0 e_2) & 2(e_2 e_3 - e_0 e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix}$$

UNCAGE (SCG, SC, T, TUG) – check to see if it is time to uncage any of the caged degrees of freedom. If it is time, redefine SCG and SC.

COMPRS (XMN, THADD, N, SC, SCG, LG) – remove rows from XMN and THADD arrays and columns from XMN array associated with caged degrees of freedom, renumber and return compressed XMN and THADD arrays.

UNPRS (THADD, M, SCG, LG) – expand THADD array putting zeros in locations associated with caged degrees of freedom.

NOTATIONAL CROSS-REFERENCE

The equations of motion are derived in References 1 and 2 using standard analytic notation. They are programmed in N-BOD2 using FORTRAN IV notation and are outputted on the line printer using an abbreviated FORTRAN notation. This section is intended to provide the user with a notation cross-reference between the three techniques used to define and evaluate the equations of motion.

1. Matrix Notation

   $\{\ \}$ – column matrix

   $[\ ]$ – square matrix

   $\llcorner\ \lrcorner$ – row or rectangular matrix

2. Vector-Tensor-Matrix Notation

   Vectors and tensors must be stored relative to particular reference frames. A post subscript outside of the brace or bracket is used to specify the reference frame in which the vector or tensor is defined. For example,

   $\{\vec{V}_K\}_K$ – 3 X 1 column matrix of the components of the vector $\vec{V}_K$ relative to body K fixed coordinates.

   $[T_K]_C$ – 3 X 3 square matrix of the components of the tensor T_K relative to the frame of computation

If the elements of the vector $\vec{V}_K$ are stored relative to the body K fixed-reference frame in the array VE and relative to the frame of computation in the array VEC,

where

```
DIMENSION VE (3, 10), VEC (3, 10),
```

the following notation is used:

Analytic Notation References 1 and 2	N-BOD2 FORTRAN IV Coding	N-BOD2 Users Manual	N-BOD2 Outputted Equations
$\{\vec{V}_K\}_K$	VE (1, K) VE (2, K) VE (3, K)	$\{VE(I,K)\}_K$	VE (K)
$\{\vec{V}_K\}_C$	VEC (1, K) VEC (2, K) VEC (3, K)	$\{VEC(I,K)\}_C$	VEC (K)

3. Mathematical Operational Notation

In the output of the equations of motion, the analytic symbology used to define several mathematical operations cannot be duplicated with a standard line printer. The following notation has been adopted:

Let

```
REAL    S1, S2, S3,
        V1 (3), V2 (3), V3 (3),
        T1 (3, 3), T2 (3, 3), T3 (3, 3),
        Q1 (4), Q2 (4), Q3 (4),
        M1 (3, 3), M2 (3, 3), M3 (3, 3)
```

define the dimension of the following quantities:

S1, S2, S3 – storage locations for the scalars S_1, S_2, S_3

V1, V2, V3 – storage locations for the components of the vectors $\vec{V}_1, \vec{V}_2, \vec{V}_3$

T1, T2, T3 – storage locations for the components of the tensors of rank two, dyads $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$

Q1, Q2, Q3 – storage locations for the components of the quaternions $\bar{q}_1$, $\bar{q}_2$, $\bar{q}_3$

M1, M2, M3 – storage locations for the components of the 3 X 3 matrices of scalars $[M_1]$, $[M_2]$, $[M_3]$

The following cross-reference for mathematical operations is used in the output of equations of motion:

- Addition

$$S_3 = S_1 + S_2 \quad \approx \text{S3} = \text{S1} + \text{S2}$$

$$\vec{v}_3 = \vec{v}_1 + \vec{v}_2 \quad \approx \text{V3} = \text{V1} + \text{V2}$$

$$T_3 = T_1 + T_2 \quad \approx \text{T3} = \text{T1} + \text{T2}$$

$$[M_3] = [M_1] + [M_2] \quad \approx \text{M3} = \text{M1} + \text{M2}$$

- Subtraction

  (Same as addition)

- Division

$$S_3 = \frac{S_1}{S_2} \quad \approx \text{S3} = \text{S1/S2}$$

$$\vec{v}_3 = \frac{1}{S_1}\vec{v}_1 \quad \approx \text{V3} = \text{V1/S1}$$

$$T_3 = \frac{1}{S_1} T_1 \quad \approx \text{T3} = \text{T1/S1}$$

- Scalar Multiplication

$$S_3 = S_1 S_2 \quad \approx \text{S3} = \text{S1*S2}$$

$$\vec{v}_3 = S_1 \vec{v}_1 \quad \approx \text{V3} = \text{S1*V1}$$

$$T_3 = S_1 T_1 \quad \approx \text{T3} = \text{S1*T1}$$

$$[M_3] = S_1 [M_1] \quad \approx \text{M3} = \text{S1*M1}$$

- Quaternion Multiplication

$$\bar{Q}_3 = \bar{Q}_1 \bar{Q}_2 \quad \approx \text{Q3} = \text{Q1*Q2}$$

- Scalar Product

$$S_1 = \vec{v}_1 \cdot \vec{v}_2 \approx S1 = V1 \cdot V2$$

$$\vec{v}_1 = T_1 \cdot \vec{v}_2 \approx V1 = T1 \cdot V2$$

$$\vec{v}_3 = \vec{v}_2 \cdot T_1 \approx V3 = V2 \cdot T1$$

- Vector Product

$$\vec{v}_3 = \vec{v}_1 \times \vec{v}_2 \approx V3 = V1 \times V2$$

$$T_1 = \vec{v}_1 \times T_2 \approx T1 = V1 \times T2$$

$$T_3 = T_2 \times \vec{v}_1 \approx T3 = T2 \times V1$$

- Tensor Product

$$T_1 = \vec{v}_1 \vec{v}_2 \approx T1 = V1\ V2$$

- Transpose Matrix

$$[M_2] = [M_1]^T \approx M_2 = M1 **T$$

- Transform Coordinates of Vectors and Tensors

If

$[M_1]$ – transformation matrix, reference frame A to reference frame B

V1, T1 – storage area for components of $\vec{v}_1$ and T_1 relative to reference frame A respectively

V2, T2 – storage area for components of $\vec{v}_1$ and T_1 relative to reference frame B respectively,

then

$$\{\vec{v}_1\}_B = [M_1]\{\vec{v}_1\}_A \approx V2 = M1 * V1$$

$$[T_1]_B = [M_1]\,[T_1]_A\,[M_1]^T \approx T2 = M1 * T1 * M1 ** T$$

- Rotation Quarternion

  Using the notation of Reference 1 (see equations 99, 100, 101, and A-21),

$$\bar{Q}_1 = \text{Cos}\frac{S_1}{2} + \mathcal{Q}\left(\{\vec{v}_1\}\right)\text{Sin}\frac{S_1}{2}$$

$$\approx \text{Q1} = \text{QUATOP (V1, S1)}$$

  Subroutine QUATOP makes use of the scalar S1 and the three elements stored in the array V1 to compute the four components of the quarternion Q1 (see Utility Routine/QUATOP).

- Transformation Matrix from Rotation Quarternion

  Using the notation of Reference 1 (see equations 96, 97, 98, and A-18),

$$[M_1] = \mathcal{F}(\bar{Q}_1)^T \approx \text{M1} = \text{TRANSO (Q1)}$$

  Subroutine TRANSO makes use of the four elements stored in the array Q1 to compute the components of the associated transformation matrix (see Utility Routine/TRANSO).

- Transform Vectors to Tensors

  Using the notation of Reference 1 (see equation 51),

$$T_1 = \mathcal{S}(\vec{v}_1) \approx \text{T1} = \text{SKEW (V1)}$$

- Pseudo-Inertia Tensors

  Using the notation of Reference 1 (see equation 53 or Reference 2, equation 98),

$$T_1 = S_1\left[\vec{v}_1 \cdot \vec{v}_2\, 1 - \vec{v}_1\,\vec{v}_2\right]$$

$$\approx \text{T1} = \text{S1} * (\text{V1} \cdot \text{V2} * 1 - \text{V1 V2})$$

$$= \text{SUEOP (V2, V1, S1)}$$

  Subroutine SUEOP makes use of the scalar S1 and the elements stored in two arrays V1 and V2 to compute the components of the tensor T1 (see Utility Routines/SUEOP).

- Normalize Vectors

$$\vec{V}_2 = \frac{\vec{V}_3 \times \vec{V}_1}{|\vec{V}_3 \times \vec{V}_1|} \approx V2 = NORM\,(V3 \times V1)$$

Goddard Space Flight Center
National Aeronautics and Space Administration
Greenbelt, Maryland December 1977

REFERENCES

1. Frisch, Harold P., "A Vector - Dyadic Development of the Equations of Motion for N-coupled Rigid Bodies and Point Masses," NASA Document TN D-7767, October 1974.

2. Frisch, Harold P., "A Vector - Dyadic Development of the Equations of Motion for N-coupled Flexible Bodies and Point Masses," NASA Document TN D-8047, August 1975.

APPENDIX A

PROGRAM LISTING FOR N-BOD2

```
C                                                                       C0100000
C                                                                       C0100100
C                                                                       C0100200
C*********************************** MAIN ***************************************C0100300
C                                                                       C0100400
C                                N-BOD2                                 C0100500
C                                                                       C0100600
C     A PROGRAM TO COMPUTE THE RELATIVE ATTITUDE DYNAMICS OF            C0100700
C       N-COUPLED FLEXIBLE BODIES, RIGID BODIES, POINT                  C0100800
C          MASSES AND N SYMMETRIC WHEELS                                C0100900
C                                                                       C0101000
C     A CONSISTANT SET OF UNITS MUST BE USED TO DEFINE INPUT DATA       C0101100
C     THESE UNITS ARE ASSUMED CONSISTANT AND WILL BE USED IN A          C0101200
C      CONSISTANT MANNER THROUGHOUT THE COMPUTATION                     C0101300
C      NO INTERNALLY CODED CONVERSION OF UNITS IS NEEDED OR PROVIDED    C0101400
C                                                                       C0101500
C     REFERENCES:                                                       C0101600
C       NASA TN D-7767  'A VECTOR-DYADIC DEVELOPEMENT OF THE EQUATIONS  C0101700
C                        OF MOTION FOR N-COUPLED RIGID BODIES AND       C0101800
C                        POINT MASSES'                                  C0101900
C                  BY HAROLD P. FRISCH   OCT,1974                       C0102000
C                                                                       C0102100
C       NASA TN D-8047  'A VECTOR-DYADIC DEVELOPEMENT OF THE EQUATIONS  C0102200
C                        OF MOTION FOR N-COUPLED FLEXIBLE BODIES AND    C0102300
C                        POINT MASSES'                                  C0102400
C                  BY HAROLD P. FRISCH   AUG, 1975                      C0102500
C                                                                       C0102600
C      IN PREPERATION   'THE N-BOD2 USER'S AND PROGRAMMERS MANUEL'      C0102700
C                  BY HAROLD P. FRISCH   SUBMITTED TO EDITORIAL 12/3/77 C0102800
C                                                                       C0102900
C                                                                       C0103000
C                                                                       C0103100
C     N-BOD2 IS DIMENSIONED TO ACCEPT A MAXIMUM OF                      C0103200
C       N - BODIES (FLEXIBLE BODIES + RIGID BODIES + POINT MASSES)      C0103300
C       N - SYMMETRIC WHEELS                                            C0103400
C      2N - FLEXIBLE MODES OF VIBRATION (TOTAL FOR ALL FLEXIBLE BODIES) C0103500
C      4N - MODAL CROSS-COUPLING COEFFICIENTS (TOTAL)                   C0103600
C      33 - INDEPENDENT DEGREES OF FREEDOM                              C0103700
C     160 - FIRST ORDER NON-LINEAR DIFFERENTIAL EQUATIONS               C0103800
C                                                                       C0103900
C         THIS VERSION OF N-BOD2 USES                                   C0104000
C              N = 10                                                   C0104100
C                                                                       C0104200
C     MAKING USE OF N.LT.10 SAVES CONSIDERABLE COMPUTER STORAGE         C0104300
C                   N.GT.10 RUN TIME FOR PRATICAL APPLICATION EXCESSIVE C0104400
C                                                                       C0104500
C                                                                       C0104600
C                                                                       C0104700
C           SYMBOL LIST ABBREVIATIONS                                   C0104800
C     IDEM2 = N**2 + N + 1 - (N*(N-1))/2                                C0104900
C     IDEM3 = (N-1)**2 + N - ((N-1)*(N-2))/2                            C0105000
C     IDEM4 = SIZE OF /LOGIC/     16    LOGICAL WORDS                   C0105100
C     IDEM5 = SIZE OF /INTG/      724   INTEGER WORDS                   C0105200
C     IDEM6 = SIZE OF /INTGZ/     70    INTEGER WORDS                   C0105300
C     IDEM7 = SIZE OF /REAL/      4354  REAL WORDS                      C0105400
C     IDEM8 = SIZE OF /REALZ/     168   REAL WORDS                      C0105500
C     IDEM9 = SIZE OF /SATELL/    1000  REAL WORDS                      C0105600
C     ALL COMPUTED VECTORS AND TENSORS IN COMPUTING FRAME COORDINATES   C0105700
C     NA = OPTION NOT AVAILABLE IN N-BOD2                               C0105800
C     # = NUMBER OF                                                     C0105900
```


Preceding Page Blank

```
C     BFC = BODY FIXED COORDINATES                                          C0106000
C     CFC = COMPUTING FRAME FIXED COORDINATES (BODY 1 OR INERTIAL)          C0106100
C     IFC = INERTIALLY FIXED COORDINATES                                    C0106200
C     EQIV(XMN) = EQUIVALENCED TO XMN ARRAY                                 C0106300
C     C:N = BY USE OF A DUMMY VARIABLE SUBSCRIPT 0 ALLOWABLE                C0106400
C                                                                           C0106500
C                        SYMBOL LIST AND STORAGE LOCATION                   C0106600
C                                                                           C0106700
C  NAME      TYPE DIMENSION STORAGE    DEFINITION AND SUBROUTINE USED IN    C0106800
C  ----      ---- --------- -------    ---------------------------------    C0106900
C  ANGD        R  3(N+1)    EQIV(XMN)  EULER ANGLE DOT (ANGLE,SETUP)        C0107000
C  AWORK       I  200       /INTG/     LOCAL WORK AREA TO SAVE STORAGE      C0107100
C  CA          R  3,N       /REAL/     CM VECTOR BFC (INBS)                 C0107200
C  CAC         R  3,N       /REAL/     CM VECTOR CFC (VDIV,TRANVD)          C0107300
C  CAC         R  3,N       /REAL/     ZERO DEF CM VECT BFC (VDIV)          C0107400
C  CBDUM,CB    R  3,0:N     /REALZ/    HINGE VECTOR BFC (INBS)              C0107500
C  CBCDUM,CBC  R  3,0:N     /REALZ/    HINGE VECTOR CFC (VDIV,TRANVD)       C0107600
C  CBN         R  3         /REALZ/    HINGE VECTOR PART (INBS)             C0107700
C  CLM         R  N         /REAL/     SCALAR TORQUE ON WHEEL (TORQUE)      C0107800
C  CNF         R  3,N       EQIV(XMN)  FORCE CENTRIPETAL + CORIOLIS(ETA)    C0107900
C  CCMC        R  3,N+1     /REAL/     ANG RATE TO COMP FRAME (RATE,DCT)    C0108000
C  CT1         I            /INTG/     COUNTER (INCPT) UNUSED AFTER         C0108100
C  CT2         I            /INTG/     COUNTER (INCPT) UNUSED AFTER         C0108200
C  CT3         I            /INTG/     COUNTER (INOPT) UNUSED AFTER         C0108300
C  CT4         I            /INTG/     COUNTER (INOPT)+PASSES THRU (DYN)    C0108400
C  CT5         I            /INTG/     COUNTER (INOPT) UNUSED AFTER         C0108500
C  DCMC        R  3,N+1     /REAL/     PART OF ANG. ACC. VEC. (RATE)        C0108600
C  DUMMY       R  1000      /SATELL/   STORAGE AREA FOR USER                C0108700
C  ETC         R  3,N+1     /REAL/     GYRO+EXT.TORQ.ON NEST (ETA,QFDOT)    C0108800
C  ETIC        R  3,N       EQIV(XMN)  INERT X-COUP TURQ. (ETA)             C0108900
C  ETM         R  33        /REAL/     SCALAR,GENEALIZED TORQUES (QFDOT)    C0109000
C  ETMC        R  3,N       EQIV(XMN)  WHEEL X-COUP TURQ. (ETA)             C0109100
C  FCF         R  3,3,4N    /REAL/     MODAL CENTRIP X-COUP(INOPT,QFDOT)    C0109200
C  FCK         R  3,4N      /REAL/     MODAL CORIOLIS X-COUP(INOPT,QFDOT)   C0109300
C  FCON        I  3(N+1)    /INTG/     CODE,FREE VECTORS (INBS)             C0109400
C  FG1         L            /LOGIC/    END OF RUN FLAG (MAIN,DYN,OUTPSP)    C0109500
C  FG2         L            /LOGIC/    ERROR INPUT DATA(MAIN,INEROR)        C0109600
C  FG3         L            /LOGIC/    ERROR INPUT DATA(MAIN,INOPT)         C0109700
C  FG4         L            /LOGIC/    UNUSED                               C0109800
C  FG5         L            /LOGIC/    OUTPUT DATA ? FLAG (MAIN,TORQLE)     C0109900
C  FLA         R  3,2N      /REAL/     MODAL CM VECTOR BFC (INOPT)          C0110000
C  FLAC        R  3,2N      /REAL/     MODAL CM VECTOR CFC (VDIV,TRANVD)    C0110100
C  FLB         R  3,2N      /REAL/     MODAL MOMENT VECTOR BFC (INOPT)      C0110200
C  FLC         R  3,2N      /REAL/     MODAL ROTATION MOMENT BFC (INOPT)    C0110300
C  FLCRC       R  3,N       /REAL/     GYRO FLEXIBLITY FORCE (ETA)          C0110400
C  FLD         R  3,3,2N    /REAL/     MODAL INERTIA DYAD BFC (INOPT)       C0110500
C  FLE         R  3,3,2N    EQIV(FLD)  FLD + FLD**T (VDIV)                  C0110600
C  FLF         R  3,3,2N    EQIV(FLJ)  FLD + FLH (VDIV)                     C0110700
C  FLIRC       R  3,N       /REAL/     GYRO FLEXIBLITY TORQUE (ETA)         C0110800
C  FLJ         R  3,3,2N    /REAL/     MODAL ROTATION DYAD BFC (INOPT)      C0110900
C  FLQ         R  3,2N      EQIV(FLB)  MODAL MOMENTUM VECTOR BFC (VDIV)     C0111000
C  FLQC        R  3,2N      /REAL/     FLQ IN CFC (VDIV,TRANVD)             C0111100
C  FLCM        R  2N        /REAL/     MODAL FREQUENCY (INOPT)              C0111200
C  FCMC        R  3,N+1     /REAL/     INERTIAL RATE VECTOR (RATE)          C0111300
C  GAM         R  3,IDEM2   /REAL/     HINGE TO CM VECTOR (VDIV,XDY)        C0111400
C  H           R            /REAL/     INTEGRATION STEP SIZE (RUNGE,INES)   C0111500
C  HM          R  3,N       /REAL/     WHEEL SPIN AXIS BFC (INBS)           C0111600
C  HMC         R  3,N       /REAL/     WHEEL SPIN AXIS CFC (VDIV,TRANVD)    C0111700
C  HMCM        R  N         /REAL/     WHEEL ANGULAR MOMENTUM(INBS,SETUP)   C0111800
C  IDEM4       I            LOCAL      SIZE OF /LOGIC/ (RSTART)             C0111900
```

```
C   IDEM5      I                LOCAL       SIZE OF /INTG/  (RSTART)              C0112000
C   IDEM6      I                LOCAL       SIZE OF /INTGZ/ (RSTART)              C0112100
C   IDEM7      I                LOCAL       SIZE OF /REAL/  (RSTART)              C0112200
C   IDEM8      I                LOCAL       SIZE OF /REALZ/ (RSTART)              C0112300
C   IDEM9      I                LOCAL       SIZE OF /SATELL/(RSTART)              C0112400
C   IINIT      I   IDEM5        EQIV(AWORK) ZERO OUT /INTG/ (RSTART)              C0112500
C   INERF      L                /LOGIC/     FLAG,BFC OR IFC FOR CFC (INOPT)       C0112600
C   IZINIT     I   IDEM6        EQIV(SCNDUM)ZERO OUT /INTGZ/ (RSTART)             C0112700
C   JCCN       I   N            /INTG/      BODY CONNECTION MATRIX (INBS)         C0112800
C   LCON       I   2(N+1)       /INTG/      CODE,LOCKED VECTORS (INBS)            C0112900
C   LANGLE     L                /LDEBUG/    PRINT EQUATIONS IN ANGLE? (MAIN)      C0113000
C   LDCT       L                /LDEBUG/    PRINT EQUATIONS IN DCT? (MAIN)        C0113100
C   LEQL       L                EQIV(  )    EQIV IN EACH SUB TO PRINT FLAG        C0113200
C   LEQUIV     L                /LDEBUG/    PRINT EQUATIONS IN EQIV? (MAIN)       C0113300
C   LETA       L                /LDEBUG/    PRINT EQUATIONS IN ETA? (MAIN)        C0113400
C   LINIT      L   IDEM4        EQIV(FG1)   ZERO OUT /LOGIC/  (RSTART)            C0113500
C   LQFDOT     L                /LDEBUG/    PRINT EQUATIONS IN QFDOT? (MAIN)      C0113600
C   LRATE      L                /LDEBUG/    PRINT EQUATIONS IN RATE? (MAIN)       C0113700
C   LRTAPE     L                /CHEKS/     CREATE RESTART TAPE? (MAIN)           C0113800
C   LRUNGE     L                /LDEBUG/    PRINT EQUATIONS IN RUNGE? (MAIN)      C0113900
C   LSETUP     L                /LDEBUG/    PRINT EQUATIONS IN SETUP? (MAIN)      C0114000
C   LSIMQ      L                /LDEBUG/    PRINT EQUATIONS IN SIMQ? (MAIN)       C0114100
C   LTORQL     L                /LDEBUG/    PRINT EQUATIONS IN TORQUE? (MAIN)     C0114200
C   LTRAN      L                /LDEBUG/    PRINT EQUATIONS IN TRAN? (MAIN)       C0114300
C   LTRANV     L                /LDEBUG/    PRINT EQUATIONS IN TRANVD? (MAIN)     C0114400
C   LTRNSI     L                /LDEBUG/    PRINT EQUATIONS IN TRNSIV? (MAIN)     C0114500
C   LVDIV      L                /LDEBUG/    PRINT EQUATIONS IN VDIV? (MAIN)       C0114600
C   LXDY       L                /LDEBUG/    PRINT EQUATIONS IN XDY? (MAIN)        C0114700
C   MC         I   N            /INTG/      BODY IN WHICH WHEEL IS IN (INBS)      C0114800
C   NBOD       I                /INTG/      NUMBER OF BODIES (INBS)               C0114900
C   NB1        I                /INTG/      NUMBER OF BODIES + 1 (INBS)           C0115000
C   NCTC       I                /INTG/      (INOPT,NA) CONSTRAINT TORQUES         C0115100
C   NEQ        I                LOCAL       # EQUATIONS SETUP BY N-BOD2 (EQIV)    C0115200
C   NFER       I                /INTG/      # FREE COORD VECTORS (INBS)           C0115300
C   NFKC       I                /INTG/      (INOPT,NA) CONSTRAINT FORCES          C0115400
C   NFLXB      I                /INTG/      # FLEXIBLE BODIES (INOPT)             C0115500
C   NFRC       I                /INTG/      # RELATIVE ANGLES COMPUTED (INOPT)    C0115600
C   NLOR       I                /INTG/      # LOCKED COORD VECTORS(INBS)          C0115700
C   NMC        I                /INTG/      TOTAL NUMBER OF WHEELS (INBS)         C0115800
C   NMOA       I                /INTG/      # WHEELS TO COMP REL ANGLE (INOPT)    C0115900
C   NMODS      I                /INTG/      TOTAL # MODES FOR SYSTEM (INOPT)      C0116000
C   NNV        I                /INTG/      # VARIABLE SPEED WHEELS (INOPT)       C0116100
C   NSTART     L                /CHEKS/     NEW OR RESTART RUN? (MAIN)            C0116200
C   NSVP       I                /INTG/      # LOCKED VECTORS TRANSFORM(VDIV)      C0116300
C   NSVG       I                /INTG/      # FREE VECTORS TRANSFORM (VDIV)       C0116400
C   NTQ        I                /INTG/      # DIFF EQS.IN SUB TORQUE (TORQUE)     C0116500
C   PCON       I   N+1          /INTG/      # CONSTRAINED AXES AT HINGES(INBS)    C0116600
C   PHI        R   3,N+1        /REAL/      EXTERNAL TORQUE ON NEST (TORQUE)      C0116700
C   PLM        R   N            /REAL/      WHEEL SPIN INERTIA (INBS)             C0116800
C   QF         R   3,3(N+1)     /REAL/      FREE VECTOR BFC (INBS)                C0116900
C   QFC        R   3,3(N+1)     /REAL/      FREE VECTOR CFC (VDIV,TRANVD)         C0117000
C   QL         R   3,2(N+1)     /REAL/      LOCKED VECTOR BFC (INBS)              C0117100
C   QLC        R   3,2(N+1)     /REAL/      LOCKED VECTOR CFC (VDIV,TRANVD)       C0117200
C   RBLC       L   N            /LOGIC/     RIGID BODY OR POINT MASS? (INBS)      C0117300
C   RINIT      R   IDEM7        EQIV(CA)    ZERO OUT /REAL/ (RSTART)              C0117400
C   RQMC       R   3,N+1        /REAL/      RELATIVE RATE VECTOR (RATE)           C0117500
C   RZINIT     R   IDEM8        EQIV(CBDUM) ZERO OUT /REALZ/ (RSTART)             C0117600
C   SC         I   3(N+1)       /INTG/      FREE VECTORS CAGED (INOPT,UNCAGE)     C0117700
C   SCC        I   N            /INTG/      UNUSED                                C0117800
C   SCG        I                /INTG/      # CAGED DEGREES (INOPT,UNCAGE)        C0117900
```

```
C   SCNCUM,SCN  I   0:N-1       /INTGZ/     CODE,CENTRIPETAL EFFECTS (INOPT)   C0118000
C   SCRCUM,SCR  I   0:N-1       /INTGZ/     CODE,CORIOLIS EFFECTS (INOPT)      C0118100
C   SCXC        I   2N          EQIV(TCRQ)  CODE,X-COUP MODES (INOPT,QFDOT)    C0118200
C   SD          I               /INTG/      CODE,DIRECTION COSINES (INOPT)     C0118300
C   SEU         I               /INTG/      CODE,EULER ANGLES (INOPT)          C0118400
C   SFCC        I               /INTG/      CODE,BODIES FLEX X-COUPLING (INOPT C0118500
C   SFKCUM,SFK  I   0:N-1       /INTGZ/     CODE,CONSTRAINT FORCE(INOPT,NA)    C0118600
C   SFLX        I               /INTG/      CODE,ALL FLEXIBLE BODIES (INOPT)   C0118700
C   SFR         I   3(N+1)      /INTG/      CODE,COMPUTE FREE VEC ANGLE(INOPT) C0118800
C   SFXM        I   N           /INTG/      # MODES EACH BODY (INOPT)          C0118900
C   SG          I               /INTG/      CODE,ALL GYROSTATS(SETS)           C0119000
C   SI          I   IDEM3       /INTG/      CODE,BODIES HINGE TO CM (SETS)     C0119100
C   SIG         I               /INTG/      UNUSED                             C0119200
C   SIXCUM,SIX  I   0:N-1       /INTGZ/     CODE,INERTIA EFFECTS (INOPT)       C0119300
C   SKCUM,SK    I   0:N-1       /INTGZ/     CODE,BODIES IN EACH NEST (SETS)    C0119400
C   SL          I               /INTG/      CODE,ALL POINT MASSES (SETS)       C0119500
C   SLK         I   3(N+1)      /INTG/      CODE,CONSTRAINT TORQUE(INOPT,NA)   C0119600
C   SMA         I   N           /INTG/      CODE,WHEEL ANGLE COMPUTE(INOPT)    C0119700
C   SMAL        I               /INTG/      CODE,SMALL ANGLES (INOPT)          C0119800
C   SMCCUM,SMC  I   0:N-1       /INTGZ/     CODE,ALL WHEELS IN NEST (INOPT)    C0119900
C   SMV         I               /INTG/      CODE,VARIABLE SPEED WHEELS (INOPT) C0120000
C   SCK         I   N+1         /INTG/      CODE,BODIES HINGE 0 - CM (VDIV)    C0120100
C   SPICUM,SPI  I   0:N-1       /INTGZ/     CODE,PSUEDO INERTIA TENSORS(INOPT) C0120200
C   SQF         I   N+1         /INTG/      CODE,FREE VECTOR AT HINGE (SETS)   C0120300
C   SQL         I   N+1         /INTG/      CODE,LOCKED VECTOR AT HINGE (SETS) C0120400
C   SR          I               /INTG/      CODE,ALL RIGID BODIES (SETS)       C0120500
C   SSCN        I               /INTG/      CODE,UNION OF ALL SCN (VDIV)       C0120600
C   SSIX        I               /INTG/      CODE,UNION OF ALL SIX (VDIV)       C0120700
C   SVA         I               /INTG/      CODE,CM VECTORS TRANSFORM (VDIV)   C0120800
C   SVB         I               /INTG/      CODE,HINGE VECTORS TRANSFORM(VDIV) C0120900
C   SVD         I               /INTG/      CODE,DON'T TRANSFORM (INOPT)       C0121000
C   SVI         I               /INTG/      CODE,INERTIA DYAD TRANSFORM (VDIV) C0121100
C   SVM         I               /INTG/      CODE,SPIN VECTORS TRANSFORM(VDIV)  C0121200
C   SVP         I   2(N+1)      /INTG/      CODE,LOCKED VECTORS TRANSFORM(VDIV C0121300
C   SVQ         I   3(N+1)      /INTG/      CODE,FREE VECTORS TRANFORM(VDIV)   C0121400
C   SXM         I   3,N         /INTG/      CODE,SMALL ANGLE KINEMATICS(INOPT) C0121500
C   SXT         I               /INTG/      CODE,TIME VARY CUL INER MAT(INOPT) C0121600
C   T           R               /REAL/      TIME (MAIN)                        C0121700
C   TEM         R   2,160       LOCAL       TEMP STORAGE AREA (RUNGE)          C0121800
C   THA         R   33          /REAL/      GENEALIZED COORDINATES(INBS,SETUP) C0121900
C   THAD        R   33          /REAL/      GENERALIZE COORD RATE (INBS,SETUP) C0122000
C   THADD       R   33          EQIV(ETM)   GENERALIZE COORD ACC (SETUP,SIMQ)  C0122100
C   THADW       R   N           /REAL/      WHEEL RATE (INBS,SETUP)            C0122200
C   THAW        R   N           /REAL/      WHEEL ANGLE (INBS,SETUP)           C0122300
C   TIMEND      R               /REAL/      TIME TO END RUN (INBS,DYN)         C0122400
C   TCRQ        I   97          /INTG/      UNUSED STORAGE AREA FOR USER       C0122500
C   TLG         R   3(N+1)      /REAL/      TIME TO UNCAGE (INOPT,UNCAGE)      C0122600
C   XCIC        R   3,3,IDEM2   /REAL/      MATRIX OF INERTIA TENSORS(VDIV,XCY C0122700
C   XI          R   3,3,N       /REAL/      INERTIA DYAD BFC (INBS)            C0122800
C   XIC         R   3,3,N       /REAL/      INERTIA DYAD CFC (VDIV,TRANVC)     C0122900
C   XIO         R   3,3,N       /REAL/      ZERO INERTIA DYAD BFC(VDIV)        C0123000
C   XMAS        R   N           /REAL/      BODY MASS (INBS)                   C0123100
C   XMCCUM,XMC  R   3,3,C:N     /REAL2/     TRANSFORM BFC TO CFC (TRNSIV,TRAN) C0123200
C   XMN         R   33,33       /REAL/      SCALAR INERTIA MATRIX (VDIV,QFDOT) C0123300
C   Y           R   160         LOCAL       SYSTEM STATE (EQIV,SETUP,TORQUE)   C0123400
C   YD          R   160         LOCAL       SYSTEM STATE DERIV (SETUP,TORQUE)  C0123500
C   YMCD        R   3,2,N+1     EQIV(XMN)   DIRECTION COSINE RATES (DCT)       C0123600
C   XMT         R   3,3,N       /REAL/      ZERO STATE TRANSFORMTION MAT(INES) C0123700
C   ZETA        R   2N          /REAL/      MODAL DAMPING RATIO (INOPT)        C0123800
C                                                                              C0123900
```

```
C                                                                       C0124000
C                                                                       C0124100
C     SUBROUTINE LOCATION                                               C0124200
C                                                                       C0124300
C        MAIN          00000100                                         C0124400
C        DYN           00200C00                                         C0124500
C        RSTART        00300C00                                         C0124600
C        INBS          00400C00                                         C0124700
C        INERCR        00500C00                                         C0124800
C        SETS          00600C00                                         C0124900
C        INOPT         00700C00                                         C0125000
C        INTCR         00800C00                                         C0125100
C        TRNSIV        00900C00                                         C0125200
C        VDIV          01000C00                                         C0125300
C        EQIV          01100C00                                         C0125400
C        TRAN          01200C00                                         C0125500
C        TRANVD        01300C00                                         C0125600
C        RATE          01400C00                                         C0125700
C        XDY           01500C00                                         C0125800
C        ETA           01600C00                                         C0125900
C        TORQUE        01700C00                                         C0126000
C        GFDGT         01800C00                                         C0126100
C        DCT           01900C00                                         C0126200
C        ANGLE         02000000                                         C0126300
C        SETUP         02100C00                                         C0126400
C        OUTPUT        02200C00                                         C0126500
C        OUTPSP        02300C00                                         C0126600
C        SIMC          02400C00                                         C0126700
C        RUNGE         02500C00                                         C0126800
C        LNCAGE        02600000                                         C0126900
C        COMPRS        0260260C                                         C0127000
C        LNPRS         02605100                                         C0127100
C        COMPAC        02606E00                                         C0127200
C        LNPAC         02609100                                         C0127300
C        KT0           02612200                                         C0127400
C        KT1           02613000                                         C0127500
C        CTAIN         02613E00                                         C0127600
C        VECTRN        02700C00                                         C0127700
C        TENTRN        02702C00                                         C0127800
C        VECNRM        02704E00                                         C0127900
C        MATMUL        02706100                                         C0128000
C        TRNSPS        02707400                                         C0128100
C        ROT           02708E00                                         C0128200
C        VECADD        02800000                                         C0128300
C        VECSUB        02801C00                                         C0128400
C        SCLV          02801500                                         C0128500
C        VECDOT        02802E00                                         C0128600
C        VECRCS        02803500                                         C0128700
C        TRIPVP        02804400                                         C0128800
C        DYADD         02806C00                                         C0128900
C        SCLD          02807C00                                         C0129000
C        DYDOTV        02808500                                         C0129100
C        VXDYCV        02809E00                                         C0129200
C        DYTCV         02811C00                                         C0129300
C        VGDYCV        02812400                                         C0129400
C        DYOP          02813500                                         C0129500
C        SUBCP         02815400                                         C0129600
C        OUTMUL        02900C00                                         C0129700
C        OUATCP        02902200                                         C0129800
C        TRANSU        02903300                                         C0129900
```

```
C
C
      IMPLICIT REAL*8(A-H,O-Z,$)
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)
      LOGICAL            NSTART, LRTAPE
      LOGICAL            LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,
     *                   LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,
     *                   LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ
C
C
      INTEGER
     * AWORK  , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , PCON   ,
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,
     * SMA    , SMCDUM, SMC    , SMV    , SOK    , SPIDUM, SPI    , SQF    ,
     * SCL    , SR     , SSCN   , SSIX   , SVA    , SVB    , SVD    , SVI    ,
     * SVM    , SVP    , SVQ    , SXM    , SXT    , TORQ   , SMAL   , SEU    ,
     * SC     , SCG    , NFLXB  , SFLX   , SFXM   , NMODS  , SFCC   , SCC    ,
     * IINIT(1)        , IZINIT(1)       , SD     , SCXC(20)
C
C
      REAL*8
     * ANGC   (33)     , CNF    (3,10)   , ETIC   (3,10)   , ETMC   (3,10)   ,
     * FLQ    (3,20)   , FLE    (3,3,20), FLH    (3,3,20),
     * THADD  (33)     , YMCD   (3,2,11), RINIT  (1)      , RZINIT(1)
C
C
      COMMON /CHEKS/              NSTART, LRTAPE
C
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,
     *                LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ
C
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)
C
C
      COMMON /INTG/     AWORK (200)     ,
     * CT1              , CT2              , CT3              , CT4              ,
     * CT5              , FCON   (33)      , JCON   (10)      , LCON   (22)      ,
     * MC     (10)      , NB1              , NGOD             , NCTC             ,
     * NFER             , NFKC             , NFRC             , NLOR             ,
     * NMV              , NMO              , NMOA             , NSVP             ,
     * NSVC             , FCON   (11)      , SD               , SFR    (33)      ,
     * SG               , SI     (55)      , SIG              , SL               ,
     * SLK    (33)      , SMA    (10)      , SOK    (11)      , SQF    (11)      ,
     * SCL    (11)      , SMV              , SR               , SSCN             ,
     * SSIX             , SVA              , SVB              , SVD              ,
     * SVI              , SVM              , SVP    (22)      , SVQ    (33)      ,
     * SXM    (3,10)    , SXT              , TORQ   (97)      , SMAL             ,
     * SEU              , NTQ              , SC     (33)      , SCG              ,
     * NFLXB            , SFLX             , SFXM   (10)      , NMODS            ,
     * SFCC             , SCC    (10)
C
C
      COMMON /INTGZ/
     * SCNDUM           , SCN    (9)       , SCRDUM           , SCR    (9)       ,
     * SFKDUM           , SFK    (9)       , SIXDUM           , SIX    (9)       ,
     * SKDUM            , SK     (9)       , SPIDUM           , SPI    (9)       ,
     * SMCDUM           , SMC    (9)
C
```

```
C                                                                       C0136000
      COMMON /REAL/                                                     C0136100
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , CCMC   (3,11)  ,  C0136200
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FCMC   (3,11)  ,  C0136300
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)  ,  C0136400
     * HMCM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)  ,  C0136500
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RUMC   (3,11)  ,  C0136600
     * T                ,                  THA    (33)     , THAD   (33)    ,  C0136700
     * THADW  (10)     , THAW   (10)     , XDIC   (3,3,66), XI     (3,3,10),  C0136800
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C0136900
     * TLG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)  ,  C0137000
     * FLD    (3,3,20), FLJ    (3,3,20), CAD    (3,10)   , XIO    (3,3,10),  C0137100
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)  ,  C0137200
     * FLCM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)  ,  C0137300
     * TIMEND                                                             C0137400
C                                                                       C0137500
C                                                                       C0137600
      COMMON /REALZ/                                                    C0137700
     * CBDUM (1,3)    , CE     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)  ,  C0137800
     * XMCDUM(1,1,9) , XMC    (3,3,10), CBN(3)                          C0137900
C                                                                       C0138000
C                                                                       C0138100
C     /SATELL/ AREA RESERVED FOR USER REQUIRED DATA                     C0138200
C                                                                       C0138300
      COMMON /SATELL/ DUMMY(1000)                                       C0138400
C                                                                       C0138500
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))     ,  C0138600
     *            (XMN(1,3),YMCC(1,1,1))     ,(XMN(1,6),CNF(1,1))    ,  C0138700
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))  ,  C0138800
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),  C0138900
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                          C0139000
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))     ,  C0139100
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))    ,  C0139200
     *            (SCNDUM,IZINIT(1))         ,(TORG(78),SCXC(1))        C0139300
C                                                                       C0139400
C                                                                       C0139500
C                                                                       C0139600
      DIMENSION Y(160),YD(160),TEM(2,160)                               C0139700
C                                                                       C0139800
C                                                                       C0139900
C     RETURN HERE FOR START OF NEXT RUN                                 C0140000
    1 CONTINUE                                                          C0140100
C                                                                       C0140200
C     INPUT CONTROL CARD                                                C0140300
      READ(5,102,END=7)                                                 C0140400
     *          NSTART,LRUNGE,LTRNSI,LVDIV,LEQUIV,LTRAN,LTRANV,LRATE,   C0140500
     *                 LXDY,LETA,LTORQU,LQFOUT,LDCT,LANGLE,LSETUP,LSIMQ,C0140600
     *                 LRTAPE                                           C0140700
C                                                                       C0140800
C     LOGIC CONTROL PARAMETERS FOR N-BOD2                               C0140900
C      NSTART = .TRUE.  A RESTART RUN AT T.NE.0                         C0141000
C      NSTART = .FALSE. STANDARD RUN START AT T.EQ.0                    C0141100
C      LEQU = .TRUE.    PRINT EQUATIONS                                 C0141200
C      LEQU = .FALSE.   BYPASS PRINTING                                 C0141300
C      LRTAPE = .TRUE.  DON'T CREATE A RESTART TAPE                     C0141400
C      LRTAPE = .FALSE. CREATE A RESTART TAPE                           C0141500
C                                                                       C0141600
C     ZERO ALL COMMON BLOCKS OR FILL THEM IN FROM THE RESTART TAPE      C0141700
      CALL RSTART (1,Y,YD,NEQ,TEM,62)                                   C0141800
C                                                                       C0141900
```

```
C     INPUT DESCRIPTION OF BASIC SYSTEM                                 C0142000
C           TOPOLOGY                                                    C0142100
C           INERTIA                                                     C0142200
C           MOMENTUM WHEELS                                             C0142300
C           GEOMETRY                                                    C0142400
C           NOMINAL STATE                                               C0142500
C           KINEMATICAL CONSTRAINTS                                     C0142600
C           INPUT  INITIAL CONDITIONS                                   C0142700
C                  RATES                                                C0142800
C                  DISPLACEMENT                                         C0142900
C                  FREE COORDINATES                                     C0143000
C                  MOMENTUM WHEELS                                      C0143100
C                  INTEGRATION TIME STEP                                C0143200
      CALL INBS                                                         C0143300
C     CHECK FOR PHYSICALLY REALIZABLE SYSTEM                            C0143400
      CALL INERCR                                                       C0143500
      IF(.NOT.FG2) GO TO 4                                              C0143600
C     FG2 RESET FALSE IN INERCR IF PHYSICALLY UNREALIZABLE              C0143700
C                                                                       C0143800
C     COMPUTE BODY LABEL SETS NEEDED FOR SUMMATION CHAINS               C0143900
      CALL SETS                                                         C0144000
C                                                                       C0144100
C     INPUT COMPUTATION OPTIONS                                         C0144200
C           FRAME OF COMPUTATION                                        C0144300
C           AUGMENTED SETS FOR SUMMATION TRUNCATION                     C0144400
C           DIRECTION COSINE DELETION                                   C0144500
C           COLUMNS OF INERTIA TENSOR DELETION                          C0144600
C           TRANSFORMATION SUPPRESSION                                  C0144700
C           EULER ANGLE TECHNIQUES                                      C0144800
C           SMALL ANGLE ASSUMPTIONS                                     C0144900
C           ANGULAR DISPLACEMENT                                        C0145000
C           MOMENTUM WHEEL RATE                                         C0145100
C           MOMENTUM WHEEL ANGLE                                        C0145200
C           FLEXIBLE BODIES                                             C0145300
C           MODAL COUPLING                                              C0145400
C           CAGED DEGREES OF FREEDOM                                    C0145500
      CALL INOPT                                                        C0145600
      IF(.NOT.FG3) GO TO 4                                              C0145700
C     FG3 RESET FALSE IN INOPT IF OPTION CARD NOT RECOGNIZED            C0145800
C                                                                       C0145900
C     INPUT PARAMETERS NEEDED TO DEFINE EXTERNAL DISTURBANCES           C0146000
C           GRAVITY                                                     C0146100
C           GRAVITY GRADIENT                                            C0146200
C           ORBIT                                                       C0146300
C           LOCALLY APPLIED FORCES                                      C0146400
C           SPRINGS                                                     C0146500
C           DAMPERS                                                     C0146600
C           MOTORS                                                      C0146700
C           MOMENTUM WHEEL CONTROL                                      C0146800
C           CONTROL SYSTEMS                                             C0146900
C           THERMAL DEFORMATION                                         C0147000
C           OTHER                                                       C0147100
      CALL INTCR                                                        C0147200
C                                                                       C0147300
C     COMPUTE INITIAL VALUES FOR ALL SYSTEM PARAMETERS                  C0147400
C           TRANSFORMATION MATRICES                                     C0147500
C           CENTER OF MASS VECTORS                                      C0147600
C           HINGE POINT VECTORS                                         C0147700
C           INERTIA TENSOR                                              C0147800
C           FREE VECTORS                                                C0147900
```

```
C              LOCKED VECTORS                                          C0148000
C              RATE VECTORS                                            C0148100
C              COMPOSITE VECTORS AND DYADS                             C0148200
C              CROSS COUPLING                                          C0148300
C              EXTERNAL DISTURBANCES                                   C0148400
C              SYSTEM DYNAMICS                                         C0148500
C              MOMENTUM WHEEL DYNAMICS                                 C0148600
C              ACCELERATION ABOUT-ALONG FREE VECTORS                   C0148700
C              DIRECTION COSINE RATES                                  C0148800
      T = 0                                                            C0148900
      CALL DYN(Y,YD,NEQ)                                               C0149000
C     OUTPUT TOTAL SYSTEM STATE AT T=0                                 C0149100
      CALL OUTPUT                                                      C0149200
      PRINT 101                                                        C0149300
      CALL OUTPSP                                                      C0149400
C                                                                      C0149500
C                                                                      C0149600
C     START BASIC INEGRATION OF SYSTEM EQUATIONS OF MOTION             C0149700
C       USE FIXED STEP FOURTH ORDER RUNGE KUTTA                        C0149800
    2 CONTINUE                                                         C0149900
C     DEGREES OF FREEDOM MAY BE UNCAGED ONLY AT THE START OF           C0150000
C      AN INTEGRATION STEP, CHECK IN UNCAGE IF IT IS TIME TO           C0150100
C      UNCAGE, SCG.EQ.0 IMPLIES NO MORE UNCAGING TO BE DONE            C0150200
C      NOTE: INITIAL UNCAGING VELOCITY = 0, IMPULSE EFFECTS            C0150300
C              HAVE NOT BEEN CODED IN PROGRAM                          C0150400
      IF(SCG.EQ.0) GO TO 3                                             C0150500
      CALL UNCAGE(SCG,SC,T,TUG)                                        C0150600
    3 CONTINUE                                                         C0150700
      CALL RUNGE(T,H,Y,YD,NEQ,ANG,TEM)                                 C0150800
C                                                                      C0150900
C     SUBROUTINE RUNGE CALLS DYN IN WHICH ALL SYSTEM DIFFERENTIAL      C0151000
C        EQUATIONS ARE SETUP AND PUT IN THE YD ARRAY                   C0151100
C                                                                      C0151200
C     CHECK END OF RUN FLAG                                            C0151300
      IF(.NOT.FG1) GO TO 5                                             C0151400
C     OUTPUT COMPUTED PARAMETERS                                       C0151500
C     IF FG5 TRUE PRINT                                                C0151600
C     IF FG5 FALSE SKIP PRINT AND GO TO RUNGE                          C0151700
C     DEFAULT FOR FG5 IS TRUE BUT MAY BE OVERRIDDEN IN TORQUE          C0151800
      IF(.NOT.FG5) GO TO 2                                             C0151900
      CALL OUTPSP                                                      C0152000
C     HAS END OF RUN FLAG BEEN SET IN OUTPSP BY USER?                  C0152100
      IF(FG1) GO TO 2                                                  C0152200
    5 CONTINUE                                                         C0152300
C                                                                      C0152400
C     SHOULD A RESTART TAPE BE MADE?                                   C0152500
      IF(LRTAPE) GO TO 6                                               C0152600
      CALL RSTART (2,Y,YD,NEQ,TEM,62)                                  C0152700
    6 CALL OUTPSP                                                      C0152800
    4 CONTINUE                                                         C0152900
C     USER SHOULD WRITE OUTPUT DATA ON FILE 1 OF TAPE 11, RESTART DATA C0153000
C        IS PUT INTO FILE 2 OF TAPE 11                                 C0153100
      REWIND 11                                                        C0153200
C                                                                      C0153300
C     GO TO 1 TO SEE IF ANOTHER N-BOD2 RUN FOLLOWS                     C0153400
      GO TO 1                                                          C0153500
    7 STOP                                                             C0153600
  100 FORMAT (A4)                                                      C0153700
  101 FORMAT ('1')                                                     C0153800
  102 FORMAT (4X,17L1)                                                 C0153900
```

```
      END                                                               C0154000
C                                                                       C0200000
      SUBROUTINE DYN(Y,YC,NEQ)                                          C0200100
C                                                                       C0200200
C           DEFINES THE LOGICAL PATHS THROUGH THE SUBROUTINES USED TO   C0200300
C             SET UP THE SIMULTANEOUS DIFFERENTIAL EQUATIONS            C0200400
C               OF MOTION FOR THE COUPLED N-BODY SYSTEM                 C0200500
C                                                                       C0200600
C                                                                       C0200700
C                                                                       C0200800
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C0200900
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C0201000
      LOGICAL          NSTART, LRTAPE                                   C0201100
C                                                                       C0201200
C                                                                       C0201300
      INTEGER                                                           C0201400
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  ,  C0201500
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C0201600
     * SI     , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C0201700
     * SMA    , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,  C0201800
     * SSCL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,  C0201900
     * SVM    , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  C0202000
     * SC     , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  00202100
     * IINIT(1)       , IZINIT(1)     , SD                              C0202200
C                                                                       C0202300
C                                                                       C0202400
C                                                                       C0202500
      REAL*8                                                            C0202600
     * ANGC  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C0202700
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                 C0202800
     * THACD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C0202900
C                                                                       C0203000
C                                                                       C0203100
      COMMON /CHEKS/           NSTART, LRTAPE                           C0203200
C                                                                       C0203300
C                                                                       C0203400
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C0203500
C                                                                       C0203600
C                                                                       C0203700
      COMMON /INTG/  AWORK (200)    ,                                   C0203800
     * CT1           , CT2           , CT3           , CT4           ,  C0203900
     * CT5           , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C0204000
     * MC    (10)    , NBI           , NBOD          , NCTC          ,  00204100
     * NFER          , NFKC          , NFRC          , NLOR          ,  C0204200
     * NMV           , NMO           , NMOA          , NSVP          ,  C0204300
     * NSVQ          , PCON  (11)    , SD            , SFR   (33)    ,  C0204400
     * SG            , SI    (33)    , SIG           , SL            ,  C0204500
     * SLK   (33)    , SMA   (10)    , SOK   (11)    , SQF   (10)    ,  C0204600
     * SSCL  (11)    , SMV           , SR            , SSCN          ,  C0204700
     * SSIX          , SVA           , SVB           , SVD           ,  C0204800
     * SVI           , SVM           , SVP   (22)    , SVQ   (33)    ,  C0204900
     * SXM   (3,10)  , SXT           , TORQ  (97)    , SMAL          ,  C0205000
     * SEU           , NTQ           , SC    (33)    , SCG           ,  C0205100
     * NFLXB         , SFLX          , SFXM  (10)    , NMODS         ,  C0205200
     * SFCC          , SCC   (10)                                      C0205300
```

```
C                                                                       C0205400
C                                                                       C0205500
      COMMON /INTGZ/                                                    C0205600
     * SCNDUM          , SCN    (9)      , SCRDUM          , SCR    (9)      ,  C0205700
     * SFKDUM          , SFK    (9)      , SIXDUM          , SIX    (9)      ,  C0205800
     * SKDUM           , SK     (9)      , SPIDUM          , SPI    (9)      ,  C0205900
     * SMCDUM          , SMC    (9)                                           C0206000
C                                                                       C0206100
C                                                                       C0206200
      COMMON /REAL/                                                     C0206300
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , CDMC   (3,11)   ,  C0206400
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FDMC   (3,11)   ,  C0206500
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)   ,  C0206600
     * HMDM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)   ,  C0206700
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RDMC   (3,11)   ,  C0206800
     * T                ,                  THA    (33)     , THAD   (33)     ,  C0206900
     * THADW  (10)     , THAW   (10)     , XDIC   (3,3,66), XI     (3,3,10),  C0207000
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C0207100
     * TUG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)   ,  C0207200
     * FLD    (3,3,20), FLJ    (3,3,20), CAD    (3,10)   , XID    (3,3,10),  C0207300
C                                                                       C0207400
     * FLIRC  (3,10)   , FLORC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)   ,  C0207500
     * FLDM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)   ,  C0207600
     * TIMEND                                                           C0207700
C                                                                       C0207800
      COMMON /REALZ/                                                    C0207900
     * CBDUM  (1,3)    , CB     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)   ,  C0208000
     * XMCDUM(1,1,9)   , XMC    (3,3,10), CBN(3)                        C0208100
C                                                                       C0208200
C                                                                       C0208300
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      ,     C0208400
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNF(1,1))     ,     C0208500
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))   ,     C0208600
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1)),      C0208700
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                              C0208800
     *            (FGI,LINIT(1))            ,(CA(1,1),RINIT(1))      ,     C0208900
     *            (CBDUM(1,1),RZINIT(1))    ,(AWORK(1),IINIT(1))     ,     C0209000
     *            (SCNDUM,IZINIT(1))                                       C0209100
C                                                                       C0209200
C                                                                       C0209300
      DIMENSION Y(NEQ),YD(NEQ)                                          C0209400
      LOGICAL LG(33)                                                    C0209500
C                                                                       C0209600
C                                                                       C0209700
C     COUNT TIMES THRU DYN                                              C0209800
      CT4 = CT4 + 1                                                     C0209900
C     CHECK FOR FIRST PASS                                              C0210000
      IF(CT4.GT.1) GO TO 1                                              C0210100
C                                                                       C0210200
C     MAKE USE OF                                                       C0210300
C        XMT = NOMINAL STATE TRANSFORMATION MATRICES                    C0210400
C        QF  = FREE COORDINATE VECTORS                                  C0210500
C        THA = INITIAL ROTATION ABOUT QF VECTOR                         C0210600
C     TO COMPUTE THE INITIAL VALUES FOR THE COMPONENTS OF THE           C0210700
C     TRANSFORMATION MATRICES.                                          C0210800
C        XMC = TRANSFORMATION MATRIX WHICH TAKES VECTORS FROM           C0210900
C              BODY FIXED COORDINATES TO COMPUTING FRAME COORDINATES    C0211000
      CALL TRNSIV(XMT,QF,THA,JCON,PCON,NBOD,RBLO,INERF,XMCDUM,XMC)      C0211100
      GO TO 2                                                           C0211200
C                                                                       C0211300
```

```
    1 CONTINUE                                                          C0211400
C                                                                       C0211500
C     RETURN HERE AFTER INITIALIZATION PASS THROUGH DYN                 C0211600
C      SORT OUT QUANTITIES IN ARRAY Y                                   C0211700
      CALL SETUP(Y,YD,NEQ,.TRUE.)                                       C0211800
C                                                                       C0211900
C     COMPLETE TRANSFORMATION MATRICES ARE NOT OBTAINED VIA INTEGRATION.C0212000
C       SIX OF NINE ELEMENTS IN EACH MATRIX HAVING LABEL IN SD OBTAINEDC0212100
C          BY INTEGRATION, REMAINING ONES BY ORTHOGONALITY               C0212200
C       MATRICES WITH LABELS NOT IN SD ASSUMED OBTAINABLE ALGEBRACIALLYC0212300
C          BY SMALL ANGLE OR EULER ANGLE TECHNIQUES                      C0212400
      CALL TRAN                                                         C0212500
C                                                                       C0212600
    2 IF(CT4.NE.1) GO TO 3                                              C0212700
C     FIRST PASS THROUGH                                                C0212800
C      TAKE INTO ACCOUNT SET SVD, ONLY THOSE VECTORS AND DYADS WITH     C0212900
C      BODY LABELS IN SET SVD ARE TO BE TRANSFORMED                     C0213000
C          COMPUTE QF AND QL VECTORS NOT DEFINED BY INPUT USING         C0213100
C           VECTOR CROSS PRODUCT DEFINITION                             C0213200
C          GET ALL VECTORS AND DYADS INTO COMPUTING REFERENCE FRAME      C0213300
C          SET UP SETS WHICH SPECIFY EXACTLY WHICH VECTORS TO BE        C0213400
C            TRANSFORMED FOR DO LOOPS IN TRANVD                         C0213500
      CALL VDIV                                                         C0213600
C     SET UP INITIAL VALUES FOR EQUATIONS AS THEY GO INTO RUNGE         C0213700
      CALL EQIV(Y,NEQ)                                                  C0213800
      GO TO 4                                                           C0213900
C                                                                       C0214000
C     NOT FIRST PASS THROUGH                                            C0214100
C       TRANSFORM ONLY SELECTED VECTORS AND DYADS                       C0214200
C     MAKE USE OF XMC TO TRANSFORM ALL BODY FIXED VECTORS AND DYAD INTO 00214300
C      COMPUTING FRAME COORDINATES                                      C0214400
C           CAC = XMC*CA                - CENTER OF MASS IN SET SVA     C0214500
C           CBC = XMC*CB                - HINGE POINT IN SET SVB        C0214600
C           QFC = XMC*QF                - FREE VECTOR IN SET SVQ        C0214700
C           QLC = XMC*QL                - LOCKED VECTORS IN SET SVP     C0214800
C           XIC = XMC*XI*(XMC)**(-1)    - INERTIA DYAD IN SET SVI       C0214900
C           HMC = XMC*HM                - WHEEL AXIS IN SET SVM         C0215000
    3 CALL TRANVD                                                       C0215100
    4 CONTINUE                                                          C0215200
C                                                                       C0215300
C     MAKE USE OF FREE VECTORS IN COMPUTING FRAME COORDINATES AND       C0215400
C      RATE ABOUT OR ALONG THEM TO DEFINE ALL RATE DEPENDENT TERMS      C0215500
      CALL RATE                                                         C0215600
C                                                                       C0215700
C     MAKE USE OF TRANSFORMED BODY FIXED VECTORS AND DYADS TO CONSTRUCT C0215800
C      THE MATRIX OF INERTIA AND PSUEDO INERTIA TENSORS                 C0215900
C         PASS ONE -    COMPUTE ELEMENTS,                             C0216000
C                          SKIP ALL ZERO ELEMENTS                       C0216100
C                          TRUNCATE SUMMATIONS USE SET SPI              C0216200
C      THERE-AFTER -    COMPUTE ELEMENTS,                              C0216300
C                          SKIP ALL ZERO ELEMENTS                       C0216400
C                          SKIP TIME CONSTANT ELEMENTS USE SET SXT      C0216500
C                          TRUNCATE SUMMATIONS USE SET SPI              C0216600
      CALL XDY                                                          C0216700
C                                                                       C0216800
C     MAKE USE OF VELOCITY AND BODY FIXED VECTORS TO COMPUTE GYROSCOPIC C0216900
C      CROSS COUPLING TERMS                                             C0217000
C          INERTIA CROSS COUPLING TRUNCATE ACCORDING TO SIX(I)          C0217100
C          CENTRIPITAL CROSS COUPLING TRUNCATE ACCORDING TO SCN(I)      C0217200
C          CORIOLIS CROSS COUPLING TRUNCATE ACCORDING TO SCR(I)         C0217300
```

```
C           MOMENTUM WHEEL COUPLING NOT TRUNCATED                          C0217400
C           FLEXIBLE BODY EFFECTS NOT TRUNCATED SET SFLX                   C0217500
      CALL ETA                                                             C0217600
C                                                                          C0217700
C     MAKE USE OF POSITION AND RATE INFORMATION TO COMPUTE                 C0217800
C      ALL NON-GYROSCOPIC TORQUES                                          C0217900
C     NOTE - SUBROUTINE TORQUE IS USER DEFINED (EMPTY IF NOT)              C0218000
      CALL TORQUE(Y,YD,NEQ)                                                C0218100
C                                                                          C0218200
C     MAKE USE OF FREE COORDINATE VECTOR TO DOT VECTOR-DYADIC EQUATION     C0218300
C      OF MOTION TO GET ACCELERATIONS ABOUT FREE COORDINATE AXES           C0218400
C      ALSO SET UP AND EXPAND EQUATIONS TO ACCOUNT FOR VARIABLE SPEED      C0218500
C      MOMENTUM WHEELS AND FLEXIBLE BODY EFFECTS                           C0218600
      CALL QFDOT                                                           C0218700
C     EXIT FROM QFDOT WITH EQUATIONS OF MOTION IN SCALAR FORM              C0218800
C                                                                          C0218900
C           REDUCE THE QFDOT EQUATIONS TO OBTAIN THADD                     C0219000
C     ENTER SIMQ WITH ELEMENTS XMN AND ETM OBTAINED IN QFDOT.              C0219100
C       EXIT WITH ACCELERATIONS THADD. XMN DESTROYED IN SIMQ               C0219200
C     EQUIVALENCE PUTS THADD AND ETM IN SAME STORAGE LOCATION              C0219300
C                                                                          C0219400
C     CHECK TO SEE IF ANY DEGREES OF FREEDOM CAGED                         C0219500
      IF(SCG.NE.0) GO TO 5                                                 C0219600
C           NONE CAGED.                                                    C0219700
      N = NPER+NMV+NMODS                                                   C0219800
      CALL SIMQ(XMN,THADD,N,33)                                            C0219900
      GO TO 6                                                              C0220000
    5 CONTINUE                                                             C0220100
C           ONE OR MORE CAGED DEGREES OF FREEDOM                           C0220200
C           DELETE AND RENUMBER ROWS AND COLS OF XMN,ETM IN COMPRS         C0220300
C           SOLVE REDUCED SET OF EQUATIONS IN SIMQ                         C0220400
C           RESTRUCTURE THADD ARRAY PLUGGING IN ZEROS IN UMPRS             C0220500
      N = NPER+NMV+NMODS-SCG                                               C0220600
      CALL COMPRS(XMN,THADD,N,SC,SCG,LG)                                   C0220700
      CALL SIMQ(XMN,THADD,N,33)                                            C0220800
      CALL UNPRS(THADD,N,SCG,LG)                                           C0220900
    6 CONTINUE                                                             C0221000
C                                                                          C0221100
C     DEFINE DIRECTION COSINE EQUATIONS IN ACCORDANCE WITH SET SD          C0221200
      CALL DCT                                                             C0221300
C                                                                          C0221400
C     DEFINE ANGULAR POSITION EQUATIONS ACCORDING TO SFR(I) AND SMA(I)     C0221500
      CALL ANGLE                                                           C0221600
C                                                                          C0221700
C     PUT ALL FIRST ORDER EQUATIONS IN ONE DIMENSIONAL ARRAY ACCEPTABLE    C0221800
C      TO INTEGRATION ROUTINE RUNGE                                        C0221900
      CALL SETUP(Y,YD,NEQ,.FALSE.)                                         C0222000
      IF(T.GE.TIMEND) GO TO 500                                            C0222100
      RETURN                                                               C0222200
C                                                                          C0222300
  500 CONTINUE                                                             C0222400
      FG1 = .FALSE.                                                        C0222500
      RETURN                                                               C0222600
      END                                                                  C0222700
```

```
C                                                                          C0300000
```

```
      SUBROUTINE RSTART (J,Y,YC,NEC,TEM,*)                                  C0300100
C                                                                           C0300200
C                                                                           C0300300
C      ZERO OUT ALL ARRAYS OR FILL THEM IN FROM THE RESTART TAPE            C0300400
C       AT THE END OF RUN CREATE A RESTART TAPE, TO DO THIS                 C0300500
C       PUT THE RESTART DATA IN FILE 2 OF TAPE 11                           C0300600
C                                                                           C0300700
C     NOTE: APPROPRIATE JOB CONTROL LANGUAGE MUST ACCOMPANY                 C0300800
C                                                                           C0300900
C     SUGGESTION: FILE 1 ON TAPE 10 PREVIOUS OUTPUT DATA                    C0301000
C                 FILE 2 ON TAPE 10 RESTART DATA                            C0301100
C                 WITH A UTILITY ROUTINE COPY FILE 1 OF TAPE 10             C0301200
C                  ONTO FILE 1 OF TAPE 11, THEN PUT NEW OUTPUT DATA         C0301300
C                  RIGHT AFTER IT WITHOUT AN EOF MARK, TAPE 11 WILL THEN    C0301400
C                  HAVE IN FILE 1 A CONTINUOUS RECORD OF THE OUTPUT         C0301500
C                  DATA FROM T=0                                            C0301600
C                AT END OF RUN NEW RESTART DATA PUT AFTER AN EOF MARK       C0301700
C                 ON TAPE 11; IT WILL THEN BE IN FILE 2 OF TAPE 11          C0301800
C                OLD TAPE 10 WILL BE UNDISTURBED                            C0301900
C                                                                           C0302000
      IMPLICIT REAL*8(A-H,O-Z,$)                                            C0302100
C                                                                           C0302200
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)         C0302300
      LOGICAL          NSTART, LRTAPE                                       C0302400
C                                                                           C0302500
      INTEGER                                                               C0302600
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , GCON   ,  C0302700
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,  C0302800
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,  C0302900
     * SMA    , SMCDUM, SMC    , SMV    , SOK    , SPIDUM, SPI    , SQF    ,  C0303000
     * SQL    , SR     , SSCN   , SSIX   , SVA    , SVB    , SVD    , SVI    ,  C0303100
     * SVM    , SVP    , SVQ    , SXM    , SXT    , TORQ   , SMAL   , SEU    ,  C0303200
     * SC     , SCG    , NFLXB , SFLX   , SFXM   , NMODS , SFCC   , SCC    ,  C0303300
     * IINIT(1)        , IZINIT(1)      , SD                                 C0303400
C                                                                           C0303500
C                                                                           C0303600
      REAL*8                                                                C0303700
     * ANGC  (33)     , CNF   (3,10)   , ETIC  (3,10)   , ETMC  (3,10)   ,  C0303800
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                    C0303900
     * THACD (33)     , YMCD  (3,2,11), RINIT (1)      , RZINIT(1)         C0304000
C                                                                           C0304100
C                                                                           C0304200
      COMMON /CHEKS/           NSTART, LRTAPE                               C0304300
C                                                                           C0304400
C                                                                           C0304500
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)               C0304600
C                                                                           C0304700
C                                                                           C0304800
      COMMON /INTG/    AWORK (200)    ,                                     C0304900
     * CT1             , CT2           , CT3           , CT4           ,    C0305000
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,    C0305100
     * MC    (10)      , NB1           , NBOD          , NCTC          ,    C0305200
     * NFER            , NFKC          , NFRC          , NLOR          ,    C0305300
     * NMV             , NMO           , NMOA          , NSVP          ,    C0305400
     * NSVQ            , PCON  (11)    , SD            , SFR   (33)    ,    C0305500
     * SG              , SI    (55)    , SIG           , SL            ,    C0305600
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,    C0305700
     * SQL   (11)      , SMV           , SR            , SSCN          ,    C0305800
     * SSIX            , SVA           , SVB           , SVD           ,    C0305900
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,    C0306000
```

```
     * SXM    (3,10)   , SXT            , TORQ   (97)     , SMAL           ,  C0306100
     * SEU             , NTQ            , SC     (33)     , SCG            ,  C0306200
     * NFLXB           , SFLX           , SFXM   (10)     , NMODS          ,  C0306300
     * SFCC            , SCC    (10)                                          C0306400
C                                                                             C0306500
C                                                                             00306600
      COMMON /INTGZ/                                                          C0306700
     * SCNDUM          , SCN    (9)      , SCRDUM         , SCR    (9)      ,  C0306800
     * SFKDUM          , SFK    (9)      , SIXDUM         , SIX    (9)      ,  C0306900
     * SKDUM           , SK     (9)      , SPIDUM         , SPI    (9)      ,  C0307000
     * SMCDUM          , SMC    (9)                                           C0307100
C                                                                             C0307200
C                                                                             C0307300
      COMMON /REAL/                                                           C0307400
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , COMC   (3,11)  ,  C0307500
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FDMC   (3,11)  ,  C0307600
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)  ,  C0307700
     * HMOM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)  ,  C0307800
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RCMC   (3,11)  ,  C0307900
     * T                ,                  THA    (33)     , THAD   (33)    ,  C0308000
     * THADW  (10)     , THAW   (10)     , XDIC   (3,3,66), XI     (3,3,10),  C0308100
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C0308200
     * TUG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)  ,  C0308300
     * FLD    (3,3,20), FLJ    (3,3,20), CAD    (3,10)   , XIO    (3,3,10),  C0308400
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)  ,  C0308500
     * FLOM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)  ,  C0308600
     * TIMEND                                                                  C0308700
C                                                                             C0308800
C                                                                             C0308900
      COMMON /REALZ/                                                          C0309000
     * CEDUM (1,3)     , CE     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)  ,  C0309100
     * XMCDUM(1,1,9)   , XMC    (3,3,10), CBN(3)                              C0309200
C                                                                             C0309300
      COMMON /SATELL/ DUMMY(1000)                                             C0309400
C                                                                             C0309500
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))     ,        C0309600
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNF(1,1))    ,        C0309700
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))  ,        C0309800
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1)),        C0309900
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                                C0310000
     *            (FG1,LINIT(1))            ,(CA(1,1),RINIT(1))     ,        C0310100
     *            (CBDUM(1,1),RZINIT(1))    ,(AWORK(1),IINIT(1))    ,        C0310200
     *            (SCNDUM,IZINIT(1))                                         C0310300
C                                                                             C0310400
C                                                                             C0310500
C                                                                             C0310600
      DIMENSION Y(160),YD(160),TEM(2,160)                                     C0310700
C                                                                             C0310800
C     COMMON BLOCK SIZES                                                      C0310900
      IDEM4 = 16                                                              C0311000
      IDEM5 = 724                                                             C0311100
      IDEM6 = 70                                                              C0311200
      IDEM7 = 4354                                                            C0311300
      IDEM8 = 168                                                             C0311400
      IDEM9 = 1000                                                            C0311500
C                                                                             C0311600
      GO TO (1,2),J                                                           C0311700
C     INITIALIZE ALL STORAGE LOCATIONS TO ZERO                                C0311800
    1 DO 3 I=1,IDEM7                                                          C0311900
    3 RINIT(I) = 0.D0                                                         C0312000
```

```
      DO 4 I=1,IDEM5                                                    C0312100
    4 DUMMY(I) = 0.D0                                                   C0312200
      DO 10 I=1,IDEM8                                                   C0312300
   10 RZINIT(I) = 0.D0                                                  C0312400
      DO 11 I=1,IDEM5                                                   C0312500
   11 IINIT(I) = 0                                                      C0312600
      DO 12 I=1,IDEM6                                                   C0312700
   12 IZINIT(I) = 0                                                     C0312800
      DO 13 I=1,IDEM4                                                   C0312900
   13 LINIT(I) = .TRUE.                                                 C0313000
      DO 14 I=1,160                                                     C0313100
      Y(I) = 0.D0                                                       C0313200
      YD(I) = 0.D0                                                      C0313300
      TEM(1,I) = 0.D0                                                   C0313400
   14 TEM(2,I) = 0.D0                                                   C0313500
      IF(.NOT.NSTART) RETURN                                            C0313600
C                                                                       C0313700
C     RESTART RUN  LOAD ALL COMMON BLOCKS AND LOCAL ARRAYS FROM         C0313800
C        THE RESTART TAPE, TAPE 10 FILE 2                               C0313900
      READ(10,102)  Y                                                   C0314000
      READ(10,102)  YD                                                  C0314100
      READ(10,101) NEQ                                                  C0314200
      READ (10,102) (DUMMY(I),I=1,IDEM9)                                C0314300
      READ(10,102)  (RINIT(I),I=1,IDEM7)                                C0314400
      READ(10,102)  (RZINIT(I),I=1,IDEM8)                               C0314500
      READ(10,101)(IINIT(I),I=1,IDEM5)                                  C0314600
      READ(10,101)(IZINIT(I),I=1,IDEM6)                                 C0314700
      READ(10,101)(LINIT(I),I=1,IDEM4)                                  C0314800
      NSTART = .FALSE.                                                  C0314900
      FG1 = .TRUE.                                                      C0315000
C     UPDATE TERMINATION TIME                                           C0315100
      READ 103, TIMEND                                                  C0315200
C     ALL DATA REQUIRED TO RESUME COMPUTATION HAS BEEN INPUTTED         C0315300
C        GO TO THE START OF THE INTEGRATION LOOP IN MAIN                C0315400
      RETURN 1                                                          C0315500
C                                                                       C0315600
C                                                                       C0315700
C     PUT EOF MARK ON TAPE 11 TO SEPERATE RESTART DATA FROM OUTPUT DATA C0315800
C        IN FILE 2 OF TAPE 11 PUT RESTART DATA                          C0315900
    2 END FILE 11                                                       C0316000
      WRITE(11,102)  Y                                                  C0316100
      WRITE(11,102)  YD                                                 C0316200
      WRITE(11,101) NEQ                                                 C0316300
      WRITE(11,102)  (DUMMY(I),I=1,IDEM9)                               C0316400
      WRITE(11,102)  (RINIT(I),I=1,IDEM7)                               C0316500
      WRITE(11,102)  (RZINIT(I),I=1,IDEM8)                              C0316600
      WRITE (11,101) ( IINIT(I),I=1,IDEM5)                              C0316700
      WRITE (11,101) (IZINIT(I),I=1,IDEM6)                              C0316800
      WRITE (11,101) ( LINIT(I),I=1,IDEM4)                              C0316900
C     REST IN PEACE ALL DATA NEEDED TO RESTART JOB IS ON FILE 2, TAPE 11C0317000
C     FURTHERMORE OLD RESTART TAPE, TAPE 10 HAS NOT BEEN DESTROYED      C0317100
C                                                                       C0317200
  101 FORMAT (16Z8)                                                     C0317300
  102 FORMAT (8Z16)                                                     C0317400
  103 FORMAT(D15.5)                                                     C0317500
      RETURN                                                            C0317600
      END                                                               C0317700
```

```
C                                                                       CC400000
      SUBROUTINE INBS                                                   CC400100
C                                                                       CC400200
C                                                                       CC400300
C       ACCEPTS ALL INFORMATION NEEDED TO DEFINE BASIC N-BODY SYSTEM    CC400400
C         1) TOPOLOGY                                                   CC400500
C         2) INERTIA CHARACTERISTICS                                    CC400600
C         3) GEOMETRIC CHARACTERISTICS                                  CC400700
C         4) KINEMATIC TRANSFORMATION                                   CC400800
C         5) MOMENTUM WHEELS, GYROSTATS                                 CC400900
C         6) INITIAL CONDITIONS                                         CC401000
C         7) FREE AND LOCKED COORDINATE AXES                            CC401100
C         8) INTEGRATION STEP SIZE                                      CC401200
C                                                                       CC401300
C        ***********  INBS INPUT DATA SETUP  ***********                CC401400
C                                                                FORMATSCC401500
C                                                                +CODES CC401600
C                                                                ********CC401700
C                                                                       CC401800
C          NBOD                                                  A 1C0 CC401900
C                                                                       CC402000
C  ************* NBOD SETS OF THE FOLLOWING CARDS, ONE PER BODY ********CC402100
C  *          N          MESS(J)                                   B 1C5*CC402200
C  *    RBLC(N)          JCON(N)          PCON(N)        XMAS(N)   C 1C1*CC402300
C  *  XI(1,1,N)        XI(1,2,N)        XI(1,3,N)                   D 1C2*CC402400
C  *  XI(2,1,N)        XI(2,2,N)        XI(2,3,N)                   D 1C2*CC402500
C  *  XI(3,1,N)        XI(3,2,N)        XI(3,3,N)                   D 1C2*CC402600
C  * XMT(1,1,N)       XMT(1,2,M)       XMT(1,3,N)                   E 1C2*CC402700
C  * XMT(2,1,N)       XMT(2,2,N)       XMT(2,3,N)                   E 1C2*CC402800
C  * XMT(3,1,N)       XMT(3,2,N)       XMT(3,3,N)                   E 1C2*CC402900
C  *    CA(1,N)          CA(2,N)          CA(3,N)                   F 1C2*CC403000
C  *    CB(1,N)          CB(2,N)          CB(3,N)                   G 1C2*CC403100
C  *    QF(1,M)          QF(2,M)          QF(3,M)                   H 1C2*CC403200
C  *  QF(1,M+2)        QF(2,M+2)        QF(3,M+2)                   H 1C2*CC403300
C  *     THA(M)           THA(M+1)         THA(M+2)                   H 1C2*CC403400
C  *    THAD(M)          THAD(M+1)        THAD(M+2)                   H 1C2*CC403500
C  *    QF(1,M)          QF(2,M)          QF(3,M)                   I 1C2*CC403600
C  *  QF(1,M+1)        QF(2,M+1)        QF(3,M+1)                   I 1C2*CC403700
C  *     THA(M)           THA(M+1)                                   I 1C2*CC403800
C  *    THAD(M)          THAD(M+1)                                   I 1C2*CC403900
C  *    QF(1,)           QF(2,M)          QF(3,M)                   J 1C2*CC404000
C  *    QL(1,L)          QL(2,L)          QL(3,L)                   J 1C2*CC404100
C  *     THA(M)                                                     J 1C2*CC404200
C  *    THAD(M)                                                     J 1C2*CC404300
C  *    QL(1,L)          QL(2,L)          QL(3,L)                   K 1C2*CC404400
C  *  QL(1,L+1)        QL(2,L+1)        QL(3,L+1)                   K 1C2*CC404500
C  ************* END OF SET OF BODY N DESCRIPTIVE CARDS ***************CC404600
C                                                                       CC404700
C  PCON(NBOD+1)                                                    L 1C0 CC404800
C       THAD(M)          THAD(M+1)        THAD(M+2)                  M 1C2 CC404900
C       THAD(M)          THAD(M+1)                                   N 1C2 CC405000
C       THAD(M)                                                      O 1C2 CC405100
C          NMO                                                       P 1C0 CC405200
C                                                                       CC405300
C  ************* NMO SETS OF THE FOLLOWING CARDS, ONE PER WHEEL ********CC405400
C  *       MO(I)          HM(1,I)          HM(2,I)        HM(3,I)   Q 1C4*CC405500
C  *     PLM(I)          MESS(J)                                     R 1C6*CC405600
C  *     THAW(I)         THADW(I)                                    S 1C2*CC405700
C  ************* END OF SET OF WHEEL I DESCRIPTIVE CARDS ***************CC405800
C                                                                       CC405900
```

```
C                       TIMEND                                  T 102 C0406000
C                                                                     C0406100
C  ***********  END OF DATA CALLED FOR BY INBS  ************************C0406200
C                                                                     C0406300
C                       CODE LIST                                     C0406400
C                                                                     C0406500
C     A - ALWAYS READ, FIRST CARD READ BY INBS                        C0406600
C           NBOD=TOTAL NUMBER OF BODIES (RIGID+FLEXIBLE+POINT MASSES) C0406700
C                                                                     C0406800
C     B - FIRST CARD FOR EACH SET OF BODY DESCRIPTION CARDS           C0406900
C           N=BODY NUMBER (SETS TO BE READ SEQUENTIALLY N=1,2,...     C0407000
C         MESS(J)=ANY ALPHANUMERIC MESSAGE WILL BE PRINTED AS A HEADINGC0407100
C                 FOR BODY N IN THE INPUT DATA ECHO PRINTED BY INBS    C0407200
C                                                                     C0407300
C     C - SECOND CARD FOR EACH SET OF BODY DESCRIPTION CARDS          C0407400
C         JCON(N)=BODY LABEL OF BODY CONTIGUOUS TO AND INBOARD OF     C0407500
C                 BODY N, HINGE POINT BETWEEN BODIES JCON(N) AND N    C0407600
C                 IS DEFINED AS HINGE POINT N-1, IF N=1 THEN JCON(1)=0 C0407700
C                 AND HINGE POINT 0 IS THE CENTER OF MASS OF BODY 1   C0407800
C         RBLC(N)=.TRUE. IF BODY N A RIGID OR FLEXIBLE BODY           C0407900
C                 .FALSE. IF BODY N A POINT MASS                      C0408000
C         PCON(N)=TOTAL NUMBER OF RIGIDLY CONSTRAINED ROTATIONAL      C0408100
C                 DEGREES OF FREEDOM AT HINGE POINT N-1 OF BODY N     C0408200
C         XMAS(N)=TOTAL MASS OF BODY N AND ALL IMBEDDED WHEELS(IF ANY) C0408300
C                                                                     C0408400
C     D - DELETE THESE 3 CARDS IF BODY N A POINT MASS                 C0408500
C        XI(I,J,N)=INERTIA TENSOR OF BODY N WITH ALL DESPUN WHEELS    C0408600
C                 INCLUDED(IF ANY) ABOUT ITS OWN CENTER OF MASS AND   C0408700
C                 RELATIVE TO THE BODY N FIXED REFERENCE FRAME, IF BODYC0408800
C                 N IS FLEXIBLE INPUT INERTIA TENSOR IN UNDEFORMED    C0408900
C                 ZERO INTERNAL STRESS STATE                          C0409000
C                                                                     C0409100
C     E - DELETE THESE 3 CARDS IF BODY N A POINT MASS                 C0409200
C       XMT(I,J,N)=TRANSFORMATION MATRIX WHICH TAKES VECTORS FROM THE C0409300
C                 BODY N TO THE BODY JCON(N) FIXED REFERENCE FRAMES   C0409400
C                 FOR ZERO RELATIVE ANGULAR ROTATION BETWEEN THE BODIESC0409500
C                 IF BODY N IS A POINT MASS THE BODY N FIXED REFERENCE C0409600
C                 FRAME AXES ARE RESPECTIVELY PARALLEL TO THOSE OF THE C0409700
C                 BODY JCON(N) FIXED REFERENCE FRAME                  C0409800
C                 IF BODY N IS A FLEXIBLE BODY IT IS ASSUMED TO BE    C0409900
C                 CLAMPED IN TRANSLATION AND ROTATION AT THE ORIGIN   C0410000
C                 OF THE BODY N FIXED REFERENCE FRAME.                C0410100
C                                                                     C0410200
C     F - ALWAYS READ                                                 C0410300
C         CA(I,N)=CENTER OF MASS VECTOR COMPONENTS OF THE VECTOR FROM C0410400
C                 THE HINGE POINT N-1(ORIGIN OF BODY N FIXED REFERENCE)C0410500
C                 TO THE CENTER OF MASS OF BODY N(UNDEFORMED POSITION C0410600
C                 IF BODY N FLEXIBLE) RELATIVE TO BODY N FIXED        C0410700
C                 REFERENCE FRAME, FOR N=1 CA(I,N)=0 I=1,2,3 SINCE BY C0410800
C                 DEFINITION THE HINGE POINT OF BODY 1 IS THE CENTER  C0410900
C                 OF MASS OF BODY 1                                   C0411000
C                                                                     C0411100
C     G - ALWAYS READ                                                 C0411200
C0CE1A03  CB(I,N)=HINGE VECTOR COMPONENTS OF THE VECTOR FROM HINGE    C0411300
C                 POINT JCON(N)-1 TO HINGE POINT N-1 RELATIVE TO THE  C0411400
C                 BODY JCON(N) FIXED REFERENCE. FOR N=1 IT IS THE     C0411500
C                 VECTOR FROM THE INERTIAL ORIGIN TO THE CENTER OF    C0411600
C                 MASS OF BODY 1,WHICH IS THE INGE POINT OF BODY 1,   C0411700
C                 RELATIVE TO THE INERTIAL REFERENCE FRAME            C0411800
C                                                                     C0411900
```

```
C     H - READ ONLY IF 3 DEGREES OF RELATIVE FREEDOM AT HINGE POINT N-1  C0412000
C                                                                        C0412100
C     I - READ ONLY IF 2 DEGREES OF RELATIVE FREEDOM AT HINGE POINT N-1  C0412200
C                                                                        C0412300
C     J - READ ONLY IF 1 DEGREES OF RELATIVE FREEDOM AT HINGE POINT N-1  C0412400
C                                                                        C0412500
C     K - READ ONLY IF 0 DEGREES OF RELATIVE FREEDOM AT HINGE POINT N-1  C0412600
C          QF(I,M)=COMPONENTS OF FREE COORDINATE VECTOR M                C0412700
C          QL(I,L)=COMPONENTS OF LOCKED COORDINATE VECTOR L              C0412800
C           THA(M)=RELATIVE DISPLACEMENT ABOUT OR ALONG QF(I,M)          C0412900
C          THAD(M)=RELATIVE RATE OF DISPLACEMENT ABOUT OR ALONG QF(I,M)  C0413000
C                  FREE AND LOCKED COORINATE VECTORS ARE INPUTTED        C0413100
C                  RELATIVE TO THE BODY FIXED FRAME IN WHICH THEY        C0413200
C                  ARE FIXED, WHEN FIXED IN BOTH BODY N AND BODY JCON(N) C0413300
C                  THEY ARE INPUTTED IN BODY JCON(N) COORDINATES         C0413400
C                                                                        C0413500
C     L - ALWAYS READ                                                    C0413600
C     PCON(NBOD+1)=NUMBER OF CONSTRAINED DEGREES OF TRANSLATIONAL        C0413700
C                  FREEDOM FOR TOTAL SYSTEM                              C0413800
C                                                                        C0413900
C     M - READ ONLY IF 3 DEGREES OF TRANSLATIONAL FREEDOM FOR SYSTEM     C0414000
C                                                                        C0414100
C     N - READ ONLY IF 2 DEGREES OF TRANSLATIONAL FREEDOM FOR SYSTEM     C0414200
C                                                                        C0414300
C     O - READ ONLY IF 1 DEGREES OF TRANSLATIONAL FREEDOM FOR SYSTEM     C0414400
C          THAD(M)=INITIAL TRANSLATIONAL RATE ALONG INERTIAL AXIS 1      C0414500
C        THAD(M+1)=INITIAL TRANSLATIONAL RATE ALONG INERTIAL AXIS 2      C0414600
C        THAD(M+2)=INITIAL TRANSLATIONAL RATE ALONG INERTIAL AXIS 3      C0414700
C                                                                        C0414800
C     P - ALWAYS READ                                                    C0414900
C           NMO=TOTAL NUMBER OF SYMMETRIC WHEELS                         C0415000
C                                                                        C0415100
C     Q - ALWAYS READ (NMO.NE.0)                                         C0415200
C           MC(I)=BODY LABEL OF BODY IN WHICH WHEEL I IS IMBEDDED        C0415300
C         HM(J,I)=COMPONENTS OF A UNIT VECTOR ALONG WHEEL SPIN AXIS      C0415400
C                 IN BODY MC(I) FIXED COORDINATES                        C0415500
C                                                                        C0415600
C     R - ALWAYS READ (NMO.NE.0)                                         C0415700
C          MESS(J)=MESSAGE TO BE PRINTED WITH WHEEL I DATA ECHO          C0415800
C           PLM(I)=SPIN INERTIA OF WHEEL I                               C0415900
C                                                                        C0416000
C     S - ALWAYS READ (NMO.NE.0)                                         C0416100
C           THAW(I)=INITIAL WHEEL ANGLE                                  C0416200
C         THAWD(I)=INITIAL WHEEL RATE                                    C0416300
C                                                                        C0416400
C     T - ALWAYS READ                                                    C0416500
C                 H=INTEGRATION STEP SIZE                                C0416600
C           TIMEND=TIME AT WHICH RUN IS TO BE ENDED                      C0416700
C                                                                        C0416800
C     100 FORMAT(I5)                                                     C0416900
C     101 FORMAT(L5,2I5,D15.5)                                           C0417000
C     102 FORMAT(3D15.5)                                                 C0417100
C     104 FORMAT(I5,3D15.5)                                              C0417200
C     105 FORMAT(I5,18A4)                                                C0417300
C     106 FORMAT(D15.5,16A4)                                             C0417400
C                                                                        C0417500
C                                                                        C0417600
      IMPLICIT REAL*8(A-H,O-Z,$)                                         C0417700
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)      C0417800
      LOGICAL          RSTART, LRTAPE                                    C0417900
```

```
C                                                                       C0418000
C                                                                       C0418100
      INTEGER                                                           C0418200
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON , PCON  ,  C0418300
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR  , SG    ,  C0418400
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL   , SLK   ,  C0418500
     * SMA    , SMCDUM, SMC    , SMV    , SOK   , SPIDUM, SPI  , SQF   ,  C0418600
     * SGL    , SR     , SSCN   , SSIX   , SVA   , SVB    , SVD  , SVI   ,  C0418700
     * SVM    , SVP    , SVQ    , SXM    , SXT   , TORQ   , SMAL , SEU   ,  C0418800
     * SC     , SCG    , NFLXB , SFLX   , SFXM  , NMODS  , SFCC , SCC   ,  C0418900
     * IINIT(1)        , IZINIT(1)      , SD                              C0419000
C                                                                       C0419100
C                                                                       C0419200
      REAL*8                                                            C0419300
     * ANGC  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C0419400
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                  C0419500
     * THADD (33)    , YMOD  (3,2,11), RINIT (1)     , RZINIT(1)       C0419600
C                                                                       C0419700
C                                                                       C0419800
      COMMON /CHEKS/           NSTART, LRTAPE                           C0419900
C                                                                       C0420000
C                                                                       C0420100
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C0420200
C                                                                       C0420300
C                                                                       C0420400
      COMMON /INTG/    AWORK (200)    ,                                 C0420500
     * CT1             , CT2           , CT3           , CT4           ,  C0420600
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C0420700
     * MC    (10)      , NBI           , NBOD          , NCTC          ,  C0420800
     * NFER            , NFKC          , NFRC          , NLOK          ,  C0420900
     * NMV             , NMO           , NMOA          , NSVP          ,  C0421000
     * NSVG            , PCON  (11)    , SD            , SFR   (33)    ,  C0421100
     * SG              , SI    (55)    , SIG           , SL            ,  C0421200
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,  C0421300
     * SGL   (11)      , SMV           , SR            , SSCN          ,  C0421400
     * SSIX            , SVA           , SVB           , SVD           ,  C0421500
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,  C0421600
     * SXM   (3,10)    , SXT           , TORQ  (97)    , SMAL          ,  C0421700
     * SEU             , NTQ           , SC    (33)    , SCG           ,  C0421800
     * NFLXB           , SFLX          , SFXM  (10)    , NMODS         ,  C0421900
     * SFCC            , SCC   (10)                                      C0422000
C                                                                       C0422100
C                                                                       C0422200
      COMMON /INTGZ/                                                    C0422300
     * SCNDUM          , SCN   (9)     , SCRDUM        , SCR   (9)     ,  C0422400
     * SFKDUM          , SFK   (9)     , SIXDUM        , SIX   (9)     ,  C0422500
     * SKDUM           , SK    (9)     , SPIDUM        , SPI   (9)     ,  C0422600
     * SMCDUM          , SMC   (9)                                       C0422700
C                                                                       C0422800
C                                                                       C0422900
      COMMON /REAL/                                                     C0423000
     * CA    (3,10)    , CAC   (3,10)  , CLM   (10)    , CCMC  (3,11)  ,  C0423100
     * DCMC  (3,11)    , ETC   (3,11)  , ETM   (33)    , FCMC  (3,11)  ,  C0423200
     * GAM   (3,66)    , H             , HM    (3,10)  , HMC   (3,10)  ,  C0423300
     * HMCM  (10)      , PHI   (3,11)  , PLM   (10)    , QF    (3,33)  ,  C0423400
     * QFC   (3,33)    , QL    (3,22)  , QLC   (3,22)  , RCMC  (3,11)  ,  C0423500
     * T               ,                 THA   (33)    , THAD  (33)    ,  C0423600
     * THADW (10)      , THAW  (10)    , XDIC  (3,3,66), XI    (3,3,10),  C0423700
     * XIC   (3,3,10), XMAS  (10)    , XMN   (33,33) , XMT   (3,3,10),  C0423800
     * TLG   (33)      , FLA   (3,20)  , FLB   (3,20)  , FLC   (3,20)  ,  C0423900
```

```
     *  FLD   (3,3,20), FLJ   (3,3,20), CAO   (3,10)  , XIO   (3,3,10),   CC424000
     *  FLIRC (3,10)  , FLCRC (3,10)  , FLAC  (3,20)  , FLQC  (3,20)  ,   C0424100
     *  FLCM  (20)    , ZETA  (20)    , FCF   (3,3,40), FCK   (3,40)  ,   C0424200
     *  TIMEND                                                            C0424300
C                                                                        C0424400
C                                                                        C0424500
      COMMON /REALZ/                                                     C0424600
     *  CBDUM (1,3)   , CB    (3,10)  , CBCDUM(1,3)   , CBC   (3,10)  ,   C0424700
     *  XMCDUM(1,1,9) , XMC   (3,3,10), CBN(3)                           C0424800
C                                                                        C0424900
C                                                                        C0425000
      EQUIVALENCE (ETM(1),THACD(1))          ,(XMN(1,1),ANGD(1))     ,   C0425100
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))    ,   C0425200
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))  ,   C0425300
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),   C0425400
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                           C0425500
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))     ,   C0425600
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))    ,   C0425700
     *            (SCNDUM,IZINIT(1))                                     C0425800
C                                                                        C0425900
C                                                                        C0426000
C                                                                        C0426100
C                                                                        C0426200
      DIMENSION TEM1(3,3),TEM2(3,3)                                      C0426300
      INTEGER MESS(18)                                                   C0426400
C                                                                        C0426500
C                                                                        C0426600
      N = 1                                                              C0426700
      L = 1                                                              C0426800
C     NBOD = NUMBER OF BODIES                                            C0426900
      READ 100, NBOD                                                     C0427000
      NB1 = NBOD+1                                                       C0427100
      PRINT 199                                                          C0427200
      PRINT 200, NBOD                                                    C0427300
      PRINT 254                                                          C0427400
      PRINT 255                                                          C0427500
      PRINT 256                                                          C0427600
      PRINT 257                                                          C0427700
      PRINT 258                                                          C0427800
      PRINT 259                                                          C0427900
      PRINT 226                                                          C0428000
C     READ INPUT FOR EACH BODY                                           C0428100
      DO 1 K=1,NBOD                                                      C0428200
      K1 = K-1                                                           C0428300
C     MESS(J) = ALPHANUMERIC DESCRIPTION OF BODY N TO BE PRINTED         C0428400
      READ 105, N, (MESS(J),J=1,18)                                      C0428500
      IF(N.EQ.K) GO TO 24                                                C0428600
      PRINT 247                                                          C0428700
C     TYPE 247                                                           C0428800
      RETURN                                                             C0428900
   24 CONTINUE                                                           C0429000
C     RBLO(K) = TRUE  IF BODY K RIGID BODY                               C0429100
C             = FALSE IF BODY K POINT MASS                               C0429200
C     JCON(K) = BODY LABEL TO WHICH BODY K IS ATTACHED AT HINGE K-1      C0429300
C     PCON(K) = NUMBER OF CONSTRAINED AXES AT HINGE POINT K-1            C0429400
C     XMAS(K) = TOTAL MASS OF BODY K PLUS MOMENTUM WHEELS                C0429500
      READ 101, RBLO(K),JCON(K),PCON(K),XMAS(K)                          C0429600
      IF(RELO(K)) GO TO 2                                                C0429700
      PRINT 201, K, (MESS(J),J=1,18)                                     C0429800
      PRINT 206,K,XMAS(K)                                                C0429900
```

```
      GO TO 3                                                           CC430000
    2 PRINT 202, K, (MESS(J),J=1,18)                                    C0430100
      PRINT 206, K,XMAS(K)                                              C0430200
C     XI(I,J,K),I,J=1,3 = COMPONENTS OF INERTIA TENSOR OF BODY K WITH   C0430300
C                         ALL DESPUN MOMENTUM WHEELS IN BODY K          C0430400
C                   MASS RELATIVE TO BODY K FIXED FRAME                 C0430500
C                         COORDINATES (ABOUT THE COMPOSITE CENTER OF    C0430600
      READ 102, ((XI(I,J,K),J=1,3),I=1,3)                               C0430700
      PRINT 216, (XI(1,J,K),J=1,3), K                                   C0430800
      PRINT 217, (XI(2,J,K),J=1,3)                                      C0430900
      PRINT 218, (XI(3,J,K),J=1,3)                                      C0431000
    3 CONTINUE                                                          C0431100
      IF(K.EQ.1) GO TO 4                                                C0431200
      PRINT 203, K,JCON(K),K1                                           C0431300
      GO TO 5                                                           C0431400
    4 PRINT 204                                                         C0431500
      PRINT 262, NBOD                                                   C0431600
    5 CONTINUE                                                          C0431700
      IF(.NOT.RBLO(K)) GO TO 8                                          C0431800
C     XMT(I,J,K),I,J=1,3 = COMPONENT OF TRANSFORMATION MATRIX BODY      C0431900
C                         K TO BODY JCON(K) COORDINATES                 C0432000
      READ 102, ((XMT(I,J,K),J=1,3),I=1,3)                              C0432100
      GO TO 9                                                           C0432200
    8 XMT(1,1,K) = 1                                                    C0432300
      XMT(1,2,K) = 0                                                    C0432400
      XMT(1,3,K) = 0                                                    C0432500
      XMT(2,1,K) = 0                                                    C0432600
      XMT(2,2,K) = 1                                                    C0432700
      XMT(2,3,K) = 0                                                    C0432800
      XMT(3,1,K) = 0                                                    C0432900
      XMT(3,2,K) = 0                                                    C0433000
      XMT(3,3,K) = 1                                                    C0433100
    9 PRINT 219, (XMT(1,J,K),J=1,3)                                     C0433200
      PRINT 220, K,(XMT(2,J,K),J=1,3)                                   C0433300
      IF(K.EQ.1) GO TO 10                                               C0433400
      PRINT 222, JCON(K),(XMT(3,J,K),J=1,3)                             C0433500
      GO TO 11                                                          C0433600
   10 PRINT 221, (XMT(3,J,K),J=1,3)                                     C0433700
   11 CONTINUE                                                          C0433800
C     CA(J,K),J=1,3 = COMPONENTS OF CENTER OF MASS FROM HINGE POINT K-1 C0433900
      READ 102, (CA(J,K),J=1,3)                                         C0434000
      PRINT 207, CA(1,K), K                                             C0434100
      PRINT 208, K1,CA(2,K)                                             C0434200
      PRINT 209, K,CA(3,K)                                              C0434300
C     CB(J,K),J=1,3 = COMPONENTS OF VECTOR FROM HINGE POINT (JCON(K)-1) C0434400
C                     HINGE POINT K-1 IN BODY JCON(K) COORDINATES       C0434500
      READ 102, (CB(J,K),J=1,3)                                         C0434600
      IF(K.NE.1) GO TO 6                                                C0434700
      PRINT 210, CB(1,K)                                                C0434800
      PRINT 211, CB(2,K)                                                C0434900
      PRINT 212, CB(3,K)                                                C0435000
      GO TO 7                                                           C0435100
    6 K2 = JCON(K)-1                                                    C0435200
      PRINT 213, CB(1,K),JCON(K)                                        C0435300
      PRINT 214, K2,CB(2,K)                                             C0435400
      PRINT 215, K1,CB(3,K)                                             C0435500
    7 CONTINUE                                                          C0435600
C                                                                       C0435700
C     READ IN FREE AND LOCKED COORDINATE VECTORS ALONG WITH INITIAL COND C0435800
      IF(K.EQ.1) GO TO 12                                               C0435900
```

```
      M = M + 3 - FCON(K-1)                                          C0436000
      L = L + FCON(K-1)                                              C0436100
   12 M1 = M+1                                                       C0436200
      M2 = M+2                                                       C0436300
      L1 = L+1                                                       C0436400
      L2 = L+2                                                       C0436500
      IF(RELO(K)) GO TO 20                                           C0436600
      IASIGN = 1                                                     C0436700
      GO TO 21                                                       C0436800
   20 IASIGN = 0                                                     C0436900
   21 CONTINUE                                                       C0437000
      IGOTO = FCON(K) + 1                                            C0437100
      GO TO (13,14,15,17),IGOTO                                      C0437200
C                                                                    C0437300
C     FCON(K) = 0, THREE DEGREES OF FREEDOM                          C0437400
   13 READ 102, ((QF(J,I),J=1,3),I=M,M2,2)                           C0437500
      READ 102, (THA(I),I=M,M2)                                      C0437600
      READ 102, (THAD(I),I=M,M2)                                     C0437700
      PRINT 246, K1                                                  C0437800
      IF(RELO(K)) GO TO 31                                           C0437900
      FCON(M)  = K                                                   C0438000
      FCON(M1) = K                                                   C0438100
      GO TO 32                                                       C0438200
   31 FCON(M)  = JCON(K)                                             C0438300
C     QF(I,M1) COMPUTED FROM QF(1,M) AND QF(1,M2)                    C0438400
C        FCON(M1) = -M HELPS LOGIC IN SUBROUTINE TRANVD              C0438500
      FCON(M1) = -M                                                  C0438600
   32 FCON(M2) =  K                                                  C0438700
      PRINT 226, QF(1,M),QF(1,M2)                                    C0438800
      PRINT 229, M,QF(2,M),M2,QF(2,M2)                               C0438900
      IF(K.EQ.1) GO TO 29                                            C0439000
      PRINT 230, FCON(M),QF(3,M),FCON(M2),QF(3,M2)                   C0439100
      GO TO 30                                                       C0439200
   29 PRINT 223,          QF(3,M),FCON(M2),QF(3,M2)                  C0439300
   30 CONTINUE                                                       C0439400
      PRINT 263, M1                                                  C0439500
      IF(IASIGN .EQ. 0 )  PRINT 224                                  C0439600
      IF(IASIGN .EQ. 1 )  PRINT 243                                  C0439700
      PRINT 225, (I,THA(I),I,THAD(I),I=M,M2)                         C0439800
      PRINT 226                                                      C0439900
      GO TO 1                                                        C0440000
C                                                                    C0440100
C     FCON(K) = 1, TWO DEGREES OF FREEDOM                            C0440200
   14 READ 102, ((QF(J,I),J=1,3),I=M,M1)                             C0440300
      READ 102,  THA(M),THA(M1)                                      C0440400
      READ 102,  THAD(M),THAD(M1)                                    C0440500
      PRINT 227, K1                                                  C0440600
      IF(RELO(K)) GO TO 33                                           C0440700
      FCON(M)  = K                                                   C0440800
      FCON(M1) = K                                                   C0440900
      LCON(L)  = K                                                   C0441000
      GO TO 34                                                       C0441100
   33 FCON(M) = JCON(K)                                              C0441200
      FCON(M1) =  K                                                  C0441300
C     QL(I,L) COMPUTED FROM QF(1,M) AND QF(1,M1)                     C0441400
C        LCON(L) = -M HELPS LOGIC IN SUBROUTINE TRANVD               C0441500
      LCON(L) = -M                                                   C0441600
   34 PRINT 226, QF(1,M), QF(1,M1)                                   C0441700
      PRINT 229, M,QF(2,M),M1,QF(2,M1)                               C0441800
      IF(K.EQ.1) GO TO 35                                            C0441900
```

```
      PRINT 230, FCON(M),QF(3,M),FCON(M1),QF(3,M1)                    C0442000
      GO TO 36                                                        C0442100
   35 PRINT 223,          QF(3,M),FCON(M1),QF(3,M1)                   C0442200
   36 CONTINUE                                                        C0442300
      PRINT 231, L                                                    C0442400
      IF(IASIGN .EQ. 0 )  PRINT 224                                   C0442500
      IF(IASIGN .EQ. 1 )  PRINT 243                                   C0442600
      PRINT 225, (I,THA(I),I,THAD(I),I=M,M1)                          C0442700
      PRINT 226                                                       C0442800
      GO TO 1                                                         C0442900
C                                                                     C0443000
C     FCON(K) = 2  ONE DEGREE OF FREEDOM                              C0443100
   15 READ 102, (QF(J,M),J=1,3)                                       C0443200
      READ 102, (QL(J,L),J=1,3)                                       C0443300
      READ 102,  THA(M)                                               C0443400
      READ 102, THAD(M)                                               C0443500
      PRINT 232, K1                                                   C0443600
      IF(RELO(K)) GO TO 37                                            C0443700
      FCON(M)  = K                                                    C0443800
      LCON(L)  = K                                                    C0443900
      LCON(L1) = K                                                    C0444000
      GO TO 38                                                        C0444100
   37 FCON(M)  = JCON(K)                                              C0444200
      LCON(L)  =  K                                                   C0444300
      LCON(L1) =  K                                                   C0444400
   38 PRINT 233, QF(1,M)                                              C0444500
      PRINT 234, M,QF(2,M)                                            C0444600
      IF(K.EQ.1) GO TO 39                                             C0444700
      PRINT 235, FCON(M), QF(3,M)                                     C0444800
      GO TO 40                                                        C0444900
   39 PRINT 266, QF(3,M)                                              C0445000
   40 PRINT 269, QL(1,L)                                              C0445100
      PRINT 270, L,QL(2,L)                                            C0445200
      PRINT 271, LCON(L),QL(3,L)                                      C0445300
      PRINT 231, L1                                                   C0445400
      IF(IASIGN .EQ. 0 )  PRINT 224                                   C0445500
      IF(IASIGN .EQ. 1 )  PRINT 243                                   C0445600
      PRINT 225, M,THA(M),M,THAD(M)                                   C0445700
      PRINT 226                                                       C0445800
      GO TO 1                                                         C0445900
C                                                                     C0446000
C     FCON(K) = 3, ZERO DEGREES OF FREEDOM                            C0446100
   17 READ 102, ((QL(J,I),J=1,3),I=L,L1)                              C0446200
      PRINT 272, K1                                                   C0446300
      LCON(L) = K                                                     C0446400
      LCON(L1) = K                                                    C0446500
      LCON(L2) = K                                                    C0446600
      PRINT 236, QL(1,L), QL(1,L1)                                    C0446700
      PRINT 237, L,QL(2,L),L1,QL(2,L1)                                C0446800
      PRINT 238, LCON(L),QL(3,L),LCON(L1),QL(3,L1)                    C0446900
      PRINT 231, L2                                                   C0447000
      PRINT 226                                                       C0447100
    1 CONTINUE                                                        C0447200
                                                                      C0447300
C     READ IN TRANSLATIONAL VELOCITY CONDITIONS FOR HINGE POINT ZERO  C0447400
C       RELATIVE TO INERTIAL ORIGIN                                   C0447500
      READ 100, PCON(NBOD+1)                                          C0447600
      NFR = M+3-PCON(NBOD)                                            C0447700
      NLR = L+FCON(NBOD) - 1                                          C0447800
      PRINT 244                                                       C0447900
```

```
      PRINT 266,PCON(NBOD+1)                                          C0448000
      I3 = 3-PCON(NBOD+1)                                             C0448100
      IF(I3.EQ.0) GO TO 25                                            C0448200
C     COMPUTE INITIAL DISPLACEMENT OF HINGE POINT ZERO FROM CB(1)     C0448300
C       INPUTING THIS WOULD BE REDUNDANT INFORMATION                  C0448400
C     DO 16, I=NFR,NFR+I3-1                                           C0448500
      ITEST=NFR+I3-1                                                  C0448600
      IF(ITEST.LT.NFR)GO TO 5000                                      C0448700
      DO 16 I=NFR,ITEST                                               C0448800
   16 THA(I) = CB(I+1-NFR,1)                                          C0448900
 5000 CONTINUE                                                        C0449000
      READ 102,(THAD(I),I=NFR,ITEST)                                  C0449100
      PRINT 261                                                       C0449200
      PRINT 243                                                       C0449300
      PRINT 225,(I,THA(I),I,THAD(I),I=NFR,ITEST)                      C0449400
      PRINT 253                                                       C0449500
      GO TO 26                                                        C0449600
   25 PRINT 248                                                       C0449700
   26 PRINT 264                                                       C0449800
      DO 22 I=1,3                                                     C0449900
      IF(I.GT.I3) GO TO 23                                            C0450000
      CBN(I) = 0.0D+00                                                C0450100
      GO TO 22                                                        C0450200
   23 CBN(I) = CB(I,1)                                                C0450300
   22 CONTINUE                                                        C0450400
      PRINT 265                                                       C0450500
      PRINT 266                                                       C0450600
      PRINT 267                                                       C0450700
      NFER = NFR+I3-1                                                 C0450800
      NLOR = NLR+PCON(NBOD+1)                                         C0450900
      PRINT 253                                                       C0451000
      PRINT 249, NFER                                                 C0451100
      PRINT 250,NLOR                                                  C0451200
      PRINT 226                                                       C0451300
C                                                                     C0451400
C     READ IN MOMENTUM WHEEL DESCRIPTION                              C0451500
C                                                                     C0451600
      READ 100, NMO                                                   C0451700
      IF(NMO.EQ.0) GO TO 50                                           C0451800
      PRINT 239, NMO                                                  C0451900
      DO 16 I=1,NMO                                                   C0452000
      READ 104, MO(I),(HM(J,I),J=1,3)                                 C0452100
      READ 106, PLM(I),(MESS(J),J=1,16)                               C0452200
      READ 102, THAW(I),THADW(I)                                      C0452300
      PRINT 285, I,MO(I),(MESS(J),J=1,16)                             C0452400
      PRINT 274, HM(1,I),MO(I)                                        C0452500
      PRINT 275, HM(2,I)                                              C0452600
      PRINT 276, I,HM(3,I)                                            C0452700
      PRINT 286, PLM(I)                                               C0452800
      PRINT 280, I                                                    C0452900
      PRINT 281, I,THAW(I)                                            C0453000
      PRINT 282, I,THADW(I)                                           C0453100
      HMOM(I) = PLM(I)*THADW(I)                                       C0453200
      PRINT 284, HMOM(I)                                              C0453300
      IF((I/2)*2.EQ.I) GO TO 15                                       C0453400
      PRINT 253                                                       C0453500
      PRINT 253                                                       C0453600
      GO TO 16                                                        C0453700
   15 PRINT 226                                                       C0453800
   16 CONTINUE                                                        C0453900
```

```
   50 CONTINUE                                                          C0454000
C                                                                       C0454100
C     READ IN INTEGRATION STEP SIZE                                     C0454200
      READ 102, H,TIMEND                                                C0454300
      PRINT 253                                                         C0454400
      PRINT 273, H                                                      C0454500
      PRINT 253                                                         C0454600
      PRINT 285, TIMEND                                                 C0454700
  100 FORMAT (I5)                                                       C0454800
  101 FORMAT (L5,2I5,D15.5)                                             C0454900
  102 FORMAT (3D15.5)                                                   C0455000
  103 FORMAT (3L5)                                                      C0455100
  104 FORMAT (I5,3D15.5)                                                C0455200
  105 FORMAT (I5,18A4)                                                  C0455300
  106 FORMAT (D15.5,16A4)                                               C0455400
  199 FORMAT('1',20(/))                                                 C0455500
  200 FORMAT (22X,'INPUT DATA FOR ',I3,' BODIES (A CONSISTANT SET OF UNIC0455600
     *TS MUST BE USED) ',/)                                             C0455700
  201 FORMAT (5X,'BODY NUMBER',I3,'  (POINT MASS)',18A4,/)              C0455800
  202 FORMAT (5X,'BODY NUMBER',I3,'  (RIGID BODY)',18A4,/)              C0455900
  203 FORMAT (10X,'BODY',I3,' CONNECTED TO BODY',I3,' AT HINGE POINT',I3C0456000
     1,/)                                                               C0456100
  204 FORMAT (10X,'CENTER OF MASS OF BODY 1 CAN TRANSLATE AND ROTATE IN C0456200
     *INERTIAL SPACE ')                                                 C0456300
  205 FORMAT (10X,'MOTION CONSTRAINED ABOUT',I3,' AXES AT HINGE POINT',IC0456400
     *3,' OF BODY',I3,/)                                                C0456500
  206 FORMAT (10X,'TOTAL MASS OF BODY ',I2,' =',D11.5,' (M) ',//)       C0456600
  207 FORMAT(10X,'COMPONENTS OF VECTOR FROM      ',D15.5,'     (BODY',I3,C0456700
     *' FIXED ')                                                        C0456800
  208 FORMAT (10X,' HINGE POINT',I3,' TO CENTER      ',D15.5,'       COORDC0456900
     *INATES)')                                                         C0457000
  209 FORMAT (10X,' OF MASS OF BODY',I3,11X,D15.5,10X,'(L)',//)         C0457100
  210 FORMAT (10X,'COMPONENTS OF VECTOR FROM      ',D15.5,'     (INERTIAL C0457200
     *')                                                                C0457300
  211 FORMAT (10X,' INERTIAL ORIGIN TO CENTER     ',D15.5,'          COORDC0457400
     *INATES)')                                                         C0457500
  212 FORMAT (10X,' OF MASS OF BODY 1             ',D15.5,10X,'(L)',//)  C0457600
  213 FORMAT (10X,'COMPONENTS OF VECTOR FROM      ',D15.5,'     (BODY',I3,C0457700
     *' FIXED ')                                                        C0457800
  214 FORMAT (10X,' HINGE POINT',I3,' TO HINGE       ',D15.5,'      COORDC0457900
     *INATES) ')                                                        C0458000
  215 FORMAT (10X,' POINT',I3,21X,D15.5,10X,'(L)',//)                   C0458100
  216 FORMAT (26X,3D12.5,'   BODY',I3,' COORDINATES ')                  C0458200
  217 FORMAT (10X,'INERTIA TENSOR =',3D12.5,'      UNITS ')             C0458300
  218 FORMAT (26X,3D12.5,5X,'(M*L**2) ',//)                             C0458400
  219 FORMAT (10X,'TRANSFORMATION MATRIX  ',3D12.5,5X,'DEFINES NOMINAL RC0458500
     1ELATIVE ')                                                        C0458600
  220 FORMAT (10X,'   BODY',I3,' TO          =',3D12.5,12X,'ORIENTATION')C0458700
  221 FORMAT (10X,'    INERTIAL            ',3D12.5,7X,'OF BODY FIXED AXEC0458800
     *S',//)                                                            C0458900
  222 FORMAT (10X,'   BODY',I3,13X,3D12.5,7X,'OF BODY FIXED AXES',//)   C0459000
  223 FORMAT (12X,'INERTIALLY',9X,D12.5,6X,' IN BODY',I3,10X,D12.5,//)  C0459100
  224 FORMAT (10X,'       INITIAL CONDITIONS ABOUT FREE AXES (RAD,RAD/SEC0459200
     *C)',/)                                                            C0459300
  225 FORMAT (15X,'THA(',I2,') =',D12.5,'     THAD(',I2,') =',D12.5)    C0459400
  226 FORMAT ('1')                                                      C0459500
  227 FORMAT (10X,' TWO DEGREES OF FREEDOM AT HINGE POINT ',I3,/)       C0459600
  228 FORMAT (10X,'COMPONENTS OF FREE   ',D12.5,6X,'COMPONENTS OF FREE  C0459700
     * ',D12.5)                                                         C0459800
  229 FORMAT (10X,' VECTOR',I3,' FIXED    =',D12.5,6X,' VECTOR',I3,' FIXC0459900
```

```
     *ED     =',D12.5)                                                      C0460000
  230 FORMAT (10X,' IN BODY',I3,10X,D12.5,6X,' IN BODY',I3,10X,D12.5,//)C0460100
  231 FORMAT (10X,'COMPONENTS OF LOCKED VECTOR',I3,' DEFINED INTERNALLY C0460200
     *BY VECTOR CROSS PRODUCT ',//)                                      C0460300
  232 FORMAT (10X,'ONE DEGREE OF FREEDOM AT HINGE POINT',I3,/)           C0460400
  233 FORMAT (10X,'COMPONENTS OF FREE    ',D12.5)                        C0460500
  234 FORMAT (10X,' VECTOR',I3,' FIXED     =',D12.5)                     C0460600
  235 FORMAT (10X,' IN BODY',I3,10X,D12.5,//)                            C0460700
  236 FORMAT (10X,'COMPONENTS OF LOCKED ',D12.5,6X,'COMPONENTS OF LOCKEDC0460800
     *  ',D12.5)                                                         C0460900
  237 FORMAT (10X,' VECTOR',I3,' FIXED     =',D12.5,6X,' VECTOR',I3,' FIXC0461000
     *ED     =',D12.5)                                                   C0461100
  238 FORMAT (10X,' IN BODY',I3,10X,D12.5,6X,' IN BODY',I3,10X,D12.5,//)C0461200
  239 FORMAT (20X,I3,'  MOMENTUM WHEELS IN SYSTEM',//)                   C0461300
  240 FORMAT (10X,'MOMENTUM WHEEL',I3,' EMBEDDED    ',D12.5,'      UNITS')C0461400
  241 FORMAT (10X,' IN BODY',I3,' , COMPONENTS     =',D12.5,'     L*F*T')C0461500
  242 FORMAT (10X,' FIXED IN BODY',I3,'            ',D12.5,//)           C0461600
  243 FORMAT (10X,'        INITIAL CONDITIONS ALONG FREE AXES (L,L/T)    C0461700
     *',/)                                                               C0461800
  244 FORMAT (10X,' TRANSLATIONAL CONDITIONS ON CENTER OF MASS OF BODY 1C0461900
     * RELATIVE TO INERTIAL ORIGIN ')                                   C0462000
  245 FORMAT (10X,5X,I3,' FREE DIRECTIONS OF TRANSLATION,PARALLEL TO I,JC0462100
     *,K INERTIAL AXES RESP ',//)                                       C0462200
  246 FORMAT (10X,'THREE DEGREES OF FREEDOM AT HINGE POINT',I3,/)        C0462300
  247 FORMAT ('  INPUT ERROR IN DATA CARDS, BODIES OUT OF SEQUENCE OR CAC0462400
     *RDS MISSING FROM PRECEEDING BODY CARDS ')                         C0462500
  248 FORMAT (10X,' TRANSLATIONAL MOTION OF BODY 1 C.M. CONSTRAINED ALONC0462600
     *G ALL THREE INERTIALLY FIXED AXES ',//)                           C0462700
  249 FORMAT (10X,' ENTIRE SYSTEM HAS',I3,' UNCONSTRAINED DEGREES OF FREC0462800
     *EDOM ',//)                                                        C0462900
  250 FORMAT (10X,' ENTIRE SYSTEM HAS',I3,' LOCKED OR CONSTRAINED DEGREEC0463000
     *S OF FREEDOM ',//)                                                C0463100
  251 FORMAT (10X,' DISPLACEMENT ABOUT OR ALONG FREE VECTOR ',I2,' COMPUC0463200
     *TED ')                                                            C0463300
  252 FORMAT (10X,' CONSTRAINT TORQUE ABOUT OR ALONG LOCKED VECTOR ',I2 C0463400
     *,' COMPUTED ')                                                    C0463500
  253 FORMAT (3(/))                                                      C0463600
  254 FORMAT (25X,'(QUANTITY)',6X,'(ENGLISH)',9X,'  (SI)',9X,'(SYMBOL)')00463700
  255 FORMAT (25X,' LENGTH    ',6X,'  FOOT    ',8X,' METER    ',10X,'L')  C0463800
  256 FORMAT (25X,' FORCE     ',6X,'  POUND   ',8X,' NEWTON   ',10X,'F')  00463900
  257 FORMAT (25X,' TIME      ',6X,'  SECOND  ',8X,' SECOND   ',10X,'T')  C0464000
  258 FORMAT (25X,' MASS      ',6X,'  SLUG    ',8X,' KILOGRAM ',10X,'M')  C0464100
  259 FORMAT (25X,' ANGLE     ',6X,'  RADIAN  ',8X,' RADIAN   ',10X,'R',//)C0464200
  260 FORMAT (20X,'(MOTION CONSTRAINED ALONG ',I3,' AXES)',//)           C0464300
  261 FORMAT (15X,'FREE VECTORS ALIGNED WITH INERTIALLY FIXED AXES RESPEC0464400
     1CTIVELY ',///)                                                    C0464500
  262 FORMAT ( 10X,'   (TRANSLATIONAL CONDITIONS GIVEN AFTER BODY',I3,' C0464600
     *DATA)',//)                                                        C0464700
  263 FORMAT (10X,' COMPONENTS OF FREE VECTOR',I3,' DEFINED INTERNALLY BC0464800
     *Y VECTOR CROSS PRODUCT ',/)                                       C0464900
  264 FORMAT (10X,'INITIAL CONDITIONS OF ALL DIRECTION COSINE TRANSFORMAC0465000
     *TION MATRICES COMPUTED')                                          C0465100
  265 FORMAT (12X,'INTERNALLY FROM THE TRANSFORMATION MATRICES WHICH DEFC0465200
     *INE NOMINAL RELATIVE')                                           C0465300
  266 FORMAT (12X,'ORIENTATION (ZERO INTERNAL STRESS STATE) AND INITIAL C0465400
     *CONDITIONS FOR')                                                  C0465500
  267 FORMAT (12X,'ROTATION ABOUT THE SPECIFIED FREE VECTORS')           C0465600
  268 FORMAT (11X,'INERTIALLY',10X,D12.5,//)                             C0465700
  269 FORMAT (10X,'COMPONENTS OF LOCKED ',D12.5)                         C0465800
  270 FORMAT (10X,' VECTOR',I3,' FIXED     =',D12.5)                     C0465900
```

```
  271 FORMAT (10X,' IN BODY',I3,1CX,D12.5,//)                          C0466000
  272 FORMAT (10X,'ZERO DEGREES OF FREEDOM AT HINGE POINT',I3,/)       C0466100
  273 FORMAT (10X,'INTEGRATION STEP SIZE FOR FIXED STEP RUNGE KUTTA INTEC0466200
     *GRATION (FOURTH ORDER) =',D12.5)                                 C0466300
  274 FORMAT (15X,'COMPONENTS OF UNIT VECTOR   ',D12.5,' (BODY',I3,' FIXC0466400
     *ED')                                                            C0466500
  275 FORMAT (15X,'  ALONG SPIN AXIS OF       =',D12.5,'   COORDINATES )C0466600
     *')                                                              C0466700
  276 FORMAT (15X,'   WHEEL',I3,16X,D12.5,//)                         C0466800
  280 FORMAT (25X,' INITIAL CONDITIONS FOR WHEEL',I3,/)                C0466900
  281 FORMAT (30X,'THAW(',I2,') = ',D12.5,' RAD')                      C0467000
  282 FORMAT (29X,'THADW(',I2,') = ',D12.5,' RAD/SEC ',/)              C0467100
  284 FORMAT (10X,'ANGULAR MOMENTUM ABOUT SPIN AXIS =',D12.5,' M*L**2/SEC0467200
     *C ',/)                                                          C0467300
  285 FORMAT (10X,'MOMENTUM WHEEL',I3,' EMBEDDED IN BODY',I3,5X,16A4,//)C0467400
  286 FORMAT (10X,'ROTATIONAL INERTIA ABOUT SPIN AXIS     ',D12.5,'  M*LC0467500
     ***2 ',//)                                                       C0467600
  288 FORMAT (10X,'RIGID BODY OR GYROSTAT NUMBER',I3)                  C0467700
  289 FORMAT(10X,'PROGRAM TIME TERMINATOR=',D12.5)                     C0467800
      RETURN                                                           C0467900
      END                                                              C0468000
```

```
C                                                                      C0500000
      SUBROUTINE INERGR                                                C0500100
C                                                                      C0500200
C                                                                      C0500300
C     INERGR  PERFORMS VARIOUS CHECKS ON INPUT DATA TO CHECK           C0500400
C      FOR BASIC TYPING ERRORS IN INPUT DATA AND VIOLATIONS            C0500500
C      OF BASIC FORMALISM RULES                                        C0500600
C                                                                      C0500700
C              THE DEFINED SYSTEM MUST BE PHYSICALLY REALIZABLE        C0500800
C                                                                      C0500900
C                                                                      C0501000
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C0501100
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, KBLG, LEQU, LINIT(1)    C0501200
      LOGICAL          NSTART, LRTAPE                                  C0501300
C                                                                      C0501400
C                                                                      C0501500
      INTEGER                                                          C0501600
     * AWORK , CTI   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  , C0501700
     * SCRDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    , C0501800
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   , C0501900
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SUF   , C0502000
     * SGL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SV1   , C0502100
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   , C0502200
     *,SC    , SCG   , NFLXB , SFLX  , SFXM  , NMCDS , SFCC  , SCC   , C0502300
     * IINIT(1)      , IZINIT(1)     , SD                              C0502400
C                                                                      C0502500
C                                                                      C0502600
      REAL*8                                                           C0502700
     * ANGC  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  , C0502800
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                 C0502900
     * THADD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C0503000
C                                                                      C0503100
C                                                                      C0503200
C                                                                      C0503300
```

```
      COMMON /CHEKS/          NSTART, LRTAPE                                  C0503400
C                                                                             C0503500
C                                                                             C0503600
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)                 C0503700
C                                                                             C0503800
C                                                                             C0503900
      COMMON /INTG/    AWORK (200)    ,                                       C0504000
     * CT1             , CT2           , CT3           , CT4           ,      C0504100
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,      C0504200
     * ME    (10)      , NBI           , NBOD          , NCTC          ,      C0504300
     * NFER            , NFKC          , NERC          , NLOR          ,      C0504400
     * NMV             , NMO           , NMOA          , NSVP          ,      C0504500
     * NSVC            , PCON  (11)    , SD            , SER   (33)    ,      C0504600
     * SG              , SI    (55)    , SIG           , SL            ,      C0504700
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,      C0504800
     * SQL   (11)      , SNV           , SR            , SSCN          ,      C0504900
     * SSIX            , SVA           , SVB           , SVC           ,      C0505000
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,      C0505100
     * SXM   (3,10)    , SXT           , TORQ  (97)    , SMAL          ,      C0505200
     * SEU             , NTQ           , SC    (33)    , SCG           ,      C0505300
     * NFLXB           , SFLX          , SFXM  (10)    , NMODS         ,      C0505400
     * SFCC            , SCC   (10)                                          C0505500
C                                                                             C0505600
C                                                                             C0505700
      COMMON /INTGZ/                                                          C0505800
     * SCNDUM          , SCN   (9)     , SCRDUM        , SCR   (9)     ,      C0505900
     * SFKDUM          , SFK   (9)     , SIXDUM        , SIX   (9)     ,      C0506000
     * SKDUM           , SK    (9)     , SPIDUM        , SPI   (9)     ,      C0506100
     * SMCDUM          , SMC   (9)                                           C0506200
C                                                                             C0506300
C                                                                             C0506400
      COMMON /REAL/                                                           C0506500
     * CA    (3,10)    , CAC   (3,10)  , CLM   (10)    , CCMC  (3,11)  ,      C0506600
     * DCMC  (3,11)    , ETC   (3,11)  , ETM   (33)    , FCMC  (3,11)  ,      C0506700
     * GAM   (3,66)    , H             , HM    (3,10)  , HMC   (3,10)  ,      C0506800
     * HMCM  (10)      , PHI   (3,11)  , PLM   (10)    , QF    (3,33)  ,      C0506900
     * QFC   (3,33)    , QL    (3,22)  , QLC   (3,22)  , RGMC  (3,11)  ,      C0507000
     * T               ,                 THA   (33)    , THAD  (33)    ,      C0507100
     * THADW (10)      , THAW  (10)    , XDIC  (3,3,66), XI    (3,3,10),      C0507200
     * XIC   (3,3,10)  , XMAS  (10)    , XMN   (33,33) , XMT   (3,3,10),      C0507300
     * TLG   (33)      , FLA   (3,20)  , FLB   (3,20)  , FLC   (3,20)  ,      C0507400
     * FLD   (3,3,20)  , FLJ   (3,3,20), CAC   (3,10)  , XIO   (3,3,10),      C0507500
     * FLIRC (3,10)    , FLCRC (3,10)  , FLAC  (3,20)  , FLQC  (3,20)  ,      C0507600
     * FLCM  (20)      , ZETA  (20)    , FCF   (3,3,40), FCK   (3,40)  ,      C0507700
     * TIMEND                                                                C0507800
C                                                                             C0507900
C                                                                             C0508000
      COMMON /REALZ/                                                          C0508100
     * CBDUM (1,3)     , CB    (3,10)  , CBCDUM(1,3)   , CBC   (3,10)  ,      C0508200
     * XMCDUM(1,1,9)   , XMC   (3,3,10), CBN(3)                              C0508300
C                                                                             C0508400
C                                                                             C0508500
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      ,      C0508600
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,      C0508700
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,      C0508800
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),       C0508900
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                               C0509000
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,      C0509100
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,      C0509200
     *            (SCNDUM,IZINIT(1))                                          C0509300
```

```
C                                                                       C0SC94C0
C                                                                       C0S09500
C                                                                       CCS09600
C                                                                       C0S09700
      DIMENSION QD(3)                                                   CCS09800
C                                                                       C0SC9900
C                                                                       CCS10000
      N = 1                                                             CCS10100
      L = 1                                                             C0S10200
C     CYCLE THRU DATA FOR ALL ECDIES                                    C0S10300
      DO 1  K=1,NBCD                                                    C0S10400
C     CHECK TO SEE IF BODY MASS IS POSITIVE                             CCS10500
      IF(XMAS(K).GT.0) GO TO 10                                         C0S10600
      PRINT 200, K,XMAS(K)                                              C0S10700
      FG2 = .FALSE.                                                     C0S10800
C     CHECK TO SEE IF TOPOLOGICAL TREE PROPERLY LABELED                 C0S10900
   10 IF(JCON(K).LT.K) GO TO 11                                         CCS11000
      PRINT 201, JCON(K),K                                              C0S11100
      FG2 = .FALSE.                                                     C0S11200
C     CHECK NUMBER OF CONSTRAINED AXES, AT MOST 3                       C0S11300
   11 IF(PCON(K).GE.0.AND.PCON(K).LE.3) GO TO 12                        C0S11400
      K1 = K-1                                                          C0S11500
      PRINT 202, K,PCON(K),K1                                           CCS11600
      FG2 = .FALSE.                                                     C0S11700
   12 IF(.NOT.RBLO(K)) GO TO 13                                         C0S11800
      DO 2  I=1,3                                                       C0S11900
      DO 2  J=1,3                                                       C0S12000
C     CHECK SYMMETRY OF INERTIA TENSOR                                  C0S12100
      IF(XI(I,J,K).EQ.XI(J,I,K)) GO TO 2                                C0S12200
      PRINT 203, K                                                      C0S12300
      FG2 = .FALSE.                                                     C0S12400
    2 CONTINUE                                                          C0S12500
C     DETERMINANT OF INERTIA TENSOR MUST BE POSITIVE FOR IT             CCS12600
C       TO HAVE REAL PRINCIPAL MOMENTS OF INERTIA                       CCS12700
      DET = XI(1,1,K)*(XI(2,2,K)*XI(3,3,K) - XI(2,3,K)*XI(3,2,K))       C0S12800
     *     -XI(1,2,K)*(XI(2,1,K)*XI(3,3,K) - XI(3,1,K)*XI(2,3,K))       CCS12900
     *     +XI(1,3,K)*(XI(2,1,K)*XI(3,2,K) - XI(2,2,K)*XI(3,1,K))       C0S13000
      IF(DET.GT.0) GO TO 13                                             C0S13100
      PRINT 204, K,DET                                                  C0S13200
      FG2 = .FALSE.                                                     C0S13300
C     CHECK ORTHOGONALITY OF TRANSFORMATION MATRIX                      C0S13400
   13 DO 3  I=1,3                                                       C0S13500
      DO 3  J=1,3                                                       C0S13600
      TEST = 0                                                          C0S13700
      DO 4  L=1,3                                                       C0S13800
    4 TEST = TEST + XMT(L,I,K)*XMT(L,J,K)                               C0S13900
      IF(I.EQ.J) GO TO 14                                               C0S14000
      IF(DABS(TEST).LT..1D-03) GO TO 3                                  C0S14100
      PRINT 205, I,J,TEST,K                                             C0S14200
      FG2 = .FALSE.                                                     C0S14300
      GO TO 3                                                           C0S14400
   14 IF(DABS(DABS(TEST)-1.D0).LT..1D-03) GO TO 3                       C0S14500
      PRINT 205, I,J,TEST,K                                             00S14600
      FG2 = .FALSE.                                                     C0S14700
    3 CONTINUE                                                          C0S14800
    1 CONTINUE                                                          C0S14900
  200 FORMAT (10X,'MASS OF BODY',I3,' =',D15.5,' ZERO OR NEGATIVE MASS EC0S15000
     *LEMENTS UNACCEPTABLE ',/)                                         C0S15100
  201 FORMAT (10X,'JCON(K) MUST BE LESS THAN K BODY',I3,' CANNOT LIE BETC0S15200
     *WEEN BODY 1 AND BODY',I3,' IN TOPOLOGICAL TREE',/)                C0S15300
```

```
  202 FORMAT (10X,'BODY',I3,' CANNOT HAVE',I3,' CONSTRAINED AXES AT HING  C0515400
     *E POINT',I3,/)                                                      C0515500
  203 FORMAT (10X,'INERTIA TENSOR OF BODY',I3,' NON-SYMMETRIC ')          C0515600
  204 FORMAT (10X,'DETERMINANT OF THE INERTIA TENSOR OF BODY',I3,' =',D1  C0515700
     *5.5,' IT MUST BE POSITIVE FOR REAL PRINCIPAL MOMENTS OF INERTIA ',  C0515800
     */)                                                                  C0515900
  205 FORMAT (10X,'EIGENVECTOR',I3,' DOT',I3,' =',D15.5,' FOR THE TRANSF  C0516000
     *ORMATION MATRIX OF BODY',I3,/)                                       C0516100
      RETURN                                                              C0516200
      END                                                                 C0516300
C                                                                         C0600000
      SUBROUTINE SETS                                                     C0600100
C                                                                         C0600200
C                                                                         C0600300
      IMPLICIT REAL*8(A-H,O-Z,$)                                          C0600400
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLC, LEQU, LINIT(1)       C0600500
      LOGICAL          NSTART, LRTAPE                                     C0600600
C                                                                         C0600700
C                                                                         C0600800
      INTEGER S1(10),S2(10),S2NNM1                                        C0600900
      INTEGER                                                             C0601000
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  ,   C0601100
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,   C0601200
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,   C0601300
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,   C0601400
     * SQL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,   C0601500
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,   C0601600
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,   C0601700
     * IINIT(1)      , IZINIT(1)     , SD                                C0601800
C                                                                         C0601900
C                                                                         C0602000
      REAL*8                                                              C0602100
     * ANGL  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,   C0602200
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                   C0602300
     * THADD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)        C0602400
C                                                                         C0602500
C                                                                         C0602600
C                                                                         C0602700
      COMMON /CHEKS/           NSTART, LRTAPE                             C0602800
C                                                                         C0602900
C                                                                         C0603000
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLC(10)             C0603100
C                                                                         C0603200
C                                                                         C0603300
      COMMON /INTG/    AWORK (200)   ,                                    C0603400
     * CT1             , CT2           , CT3           , CT4           ,  C0603500
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C0603600
     * MC    (10)      , NB1           , NBOD          , NCTC          ,  C0603700
     * NFER            , NFKC          , NFRC          , NLOR          ,  C0603800
     * NMV             , NMO           , NMOA          , NSVP          ,  C0603900
     * NSVC            , PCON  (11)    , SD            , SFR   (33)    ,  C0604000
     * SG              , SI    (55)    , SIG           , SL            ,  C0604100
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,  C0604200
     * SQL   (11)      , SMV           , SR            , SSCN          ,  C0604300
     * SSIX            , SVA           , SVB           , SVD           ,  C0604400
```

```
     * SVI              , SVM              , SVP   (22)     , SVQ   (33)     ,  C0E04500
     * SXM   (3,10)     , SXT              , TORQ  (97)     , SMAL           ,  C0E04600
     * SEU              , NTQ              , SC    (33)     , SCG            ,  C0EC4700
     * NFLXB            , SFLX             , SFXM  (10)     , NMODS          ,  C0EC4800
     * SFCC             , SCC   (10)                                          ,  C0E04900
C                                                                              C0EC5000
C                                                                              C0E05100
      COMMEN /INTGZ/                                                           C0E05200
     * SCNDUM           , SCN   (9)      , SCRDUM          , SCR   (9)      ,  C0E05300
     * SFKDUM           , SFK   (9)      , SIXDUM          , SIX   (9)      ,  C0E05400
     * SKDUM            , SK    (9)      , SPIDUM          , SPI   (9)      ,  CCE05500
     * SMCDUM           , SMC   (9)                                           ,  C0E05600
C                                                                              C0E05700
C                                                                              CCE05800
      COMMON /REAL/                                                            C0E05900
     * CA     (3,10)    , CAC   (3,10)   , CLM   (10)      , CCMC  (3,11)   ,  C0E06000
     * DCMC   (3,11)    , ETC   (3,11)   , ETM   (33)      , FCMC  (3,11)   ,  CCE06100
     * GAM    (3,66)    , H              , HM    (3,10)    , HMC   (3,10)   ,  C0E06200
     * HMCM   (10)      , PHI   (3,11)   , PLM   (10)      , QF    (3,33)   ,  C0E06300
     * QFC    (3,33)    , QL    (3,22)   , QLC   (3,22)    , RDMC  (3,11)   ,  C0E06400
     * T                ,                  THA   (33)      , THAD  (33)     ,  C0E06500
     * THADM  (10)      , THAM  (10)     , XDIC  (3,3,66), XI    (3,3,10),  C0E06600
     * XIC    (3,3,10), XMAS  (10)     , XMN   (33,33)   , XMT   (3,3,10),  CCE06700
     * TUG    (33)      , FLA   (3,20)   , FLB   (3,20)    , FLC   (3,20)   ,  CCE06800
     * FLD    (3,3,20), FLJ   (3,3,20), CAU   (3,10)    , XIU   (3,3,10),  C0E06900
     * FLIRC  (3,10)    , FLCRC (3,10)   , FLAC  (3,20)    , FLUC  (3,20)   ,  CCE07000
     * FLCM   (20)      , ZETA  (20)     , FCF   (3,3,40), FCK   (3,40)   ,  C0EC7100
     * TIMEND                                                                  C0E07200
C                                                                              C0E07300
C                                                                              C0E07400
      COMMEN /REALZ/                                                           C0E07500
     * CEDUM (1,3)      , CB    (3,10)   , CBCDUM(1,3)     , CBC   (3,10)   ,  C0E07600
     * XMCDUM(1,1,9)    , XMC   (3,3,10), CBN(3)                              C0E07700
C                                                                              C0E07800
C                                                                              CC607900
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      ,       C0E08000
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,       C0E08100
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,       C0E08200
     *            (FLB(1,1),FLC(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),        C0E08300
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                C0EC8400
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))      ,       C0E08500
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,       CCEC8600
     *            (SCNDUM,IZINIT(1))                                           C0EC870C
C                                                                              C0EC8800
C                                                                              C0EC8900
C                                                                              CC609000
C                                                                              C0E09100
C                                                                              C0E09200
C                                                                              C0E09300
C     GET THE SETS SR AND SL                                                   C0E09400
C       SR = SET OF BODY LABELS OF RIGID BODIES (COMPACTED)                    C0E09500
C       SL = SET OF BODY LABELS OF LINEAR OSCILLATORS (COMPACTED)              CCE09600
C       SG = SET OF BODY LABELS OF GYROSTATS (COMPACTED)                       CCE09700
C                                                                              C0E09800
C                                                                              CCEC9900
      NR = 0                                                                   C0E10000
      NL = 0                                                                   C0E10100
      DO 1  I=1,NBOD                                                           C0E10200
      IF(RELU(I)) GO TO 14                                                     CCE10300
      NL = NL + 1                                                              C0E10400
```

```
      S2(NL) = I                                                        C0E10500
      GO TO 1                                                           C0E10600
   14 NR = NR + 1                                                       C0E10700
      S1(NR) = I                                                        C0E10800
    1 CONTINUE                                                          C0E10900
C     COMPACT S1(NR) INTO SR AND                                        C0E11000
C           S2(NL) INTO SL                                              C0E11100
      CALL COMPAC(S1,NR,SR)                                             C0E11200
      CALL COMPAC(S2,NL,SL)                                             C0E11300
C                                                                       C0E11400
C     GET SET SG OF GYROSTAT BODY LABELS                                C0E11500
      IF(NMG.EQ.0) GO TO 19                                             C0E11600
      DO 16  I=1,10                                                     C0E11700
      S1(I) = 0                                                         C0E11800
   16 S2(I) = 0                                                         C0E11900
      DO 17  M=1,NMG                                                    C0E12000
      L = MG(M)                                                         C0E12100
   17 S1(L) = 1                                                         C0E12200
      K = 0                                                             C0E12300
      DO 18  I=1,10                                                     C0E12400
      IF(S1(I).EQ.0) GO TO 18                                           C0E12500
      K = K+1                                                           C0E12600
      S2(K) = I                                                         C0E12700
   18 CONTINUE                                                          C0E12800
      CALL COMPAC(S2,K,SG)                                              C0E12900
      GO TO 15                                                          C0E13000
   19 SG = 0                                                            C0E13100
      GO TO 15                                                          C0E13200
C                                                                       C0E13300
C     GET THE SETS OF BODIES OUTBOARD OF HINGE POINT K                  C0E13400
C        (COMPACTED FORMS IN SK(K)                                      C0E13500
   15 NM1 = NBOD - 1                                                    C0E13600
      DO 2 L=1,NBOD                                                     C0E13700
      K = L-1                                                           C0E13800
      DO 4   I=1,NBOD                                                   C0E13900
    4 S1(I) = 0                                                         C0E14000
      NC = 0                                                            C0E14100
C     CYCLE THRU BODIES WHICH CAN BE OUTBOARD OF K                      C0E14200
      ITEST = K+1                                                       C0E14300
      DO 3 J=ITEST,NBOD                                                 C0E14400
      IN = J                                                            C0E14500
    5 IF(IN-1-K) 6,7,8                                                  C0E14600
C     BODY J NOT ON PATH FROM K                                         C0E14700
    6 GO TO 3                                                           C0E14800
C     BODY J OUTBOARD OF HINGE K ON CHAIN FROM K                        C0E14900
    7 NC = NC + 1                                                       C0E15000
      S1(NC) = J                                                        C0E15100
      GO TO 3                                                           C0E15200
C     BODY J CAN BE ON CHAIN FROM K                                     C0E15300
    8 IN = JCON(IN)                                                     C0E15400
      GO TO 5                                                           C0E15500
    3 CONTINUE                                                          C0E15600
C     COMPACT SET SK(K)                                                 C0E15700
      CALL COMPAC(S1,NC,SK(K))                                          C0E15800
    2 CONTINUE                                                          C0E15900
C     GET SETS OF BODIES ON PATH FROM HINGE POINT I                     C0E16000
C       TO C.M. OF BODY N+1; SI(KTO(NBOD,I,N)                           C0E16100
C     KTO = FUNCTION TO PUT TRIANGULAR ARRAYS IN ONE DIMENSIONAL ARRAY  C0E16200
      DO 9 L=1,NBOD                                                     C0E16300
      I = L-1                                                           C0E16400
```

```
      CALL UNPAC(S2,NSET,SK(I))                                         CC616500
      DO 10 N=1,NM1                                                     C0616600
   10 SI(KTO(NBOD,I,N)) = 0                                             C0616700
      DO 11  NN=1,NSET                                                  CC616800
      DO 13  K=1,NBOD                                                   CC616900
   13 SI(K) = 0                                                         00617000
      NC = 0                                                            C0617100
      IN = S2(NN)                                                       CC617200
   12 IF(IN-1.LT.I) GO TO 20                                            C0617300
      NC = NC + 1                                                       C0617400
      SI(NC) = IN                                                       C0617500
      IN = JCCN(IN)                                                     CC617600
      GO TO 12                                                          00617700
   20 CONTINUE                                                          C0617800
C                                                                       CC617900
      CALL COMPAC(S1,NC,SI(KTO(NBOD,I,S2(NN)-1)))                       CC618000
   11 CONTINUE                                                          C0618100
    9 CONTINUE                                                          C0618200
C                                                                       C0618300
C     COMPUTE REQUIRED SUMMATION SETS                                   CC618400
      MM = 1                                                            C0618500
      LL = 1                                                            C0618600
      DO 22  K=1,NB1                                                    C0618700
      IF(K.EQ.1) GO TO 23                                               C0618800
      MM = MM+3*PCON(K-1)                                               C0618900
      LL = LL+PCON(K-1)                                                 CC619000
   23 IF(PCON(K).NE.3) GO TO 24                                         C0619100
      SQF(K) = 0                                                        C0619200
      SQL(K) = LL                                                       C0619300
      GO TO 22                                                          C0619400
   24 IF(PCON(K).NE.0) GO TO 25                                         C0619500
      SQF(K) = MM                                                       C0619600
      SQL(K) = 0                                                        C0619700
      GO TO 22                                                          C0619800
   25 SQF(K) = MM                                                       C0619900
      SQL(K) = LL                                                       CC620000
   22 CONTINUE                                                          C0620100
      RETURN                                                            CC620200
      END                                                               CC620300
```

```
C                                                                       CC700000
      SUBROUTINE INOPT                                                  C0700100
C     ACCEPTS ALL INFORMATION NEEDED TO DEFINE THE COMPUTATIONAL        C0700200
C       OPTIONS WHICH CAN BE EXECUTED BY THE USER                       C0700300
C     THE FOLLOWING AUGMENTED SET DEFINITIONS ARE MADE                  C0700400
C       'NEST I' = ALL BODIES OUTBOARD OF HINGE POINT I+1 BODY I+1      C0700500
C          SPI(I) = ALL BODIES OF NEST I CONTRIBUTING SIGNIFICANTLY     C0700600
C                   TO PSEUDO-INERTIA TENSORS OF NEST I                 C0700700
C          SIX(I) = ALL BODIES OF NEST I CONTRIBUTING SIGNIFICANTLY     C0700800
C                   TO INERIA CROSS COUPLING EFFECTS IN NEST I MOTION   C0700900
C                   EQUATION                                            C0701000
C          SCN(I) = ALL BODIES OF NEST I CONTRIBUTING SIGNIFICANTLY     C0701100
C                   TO CENTRIPITAL CROSS COUPLING EFFECTS IN NEST I     C0701200
C                   MOTION EQUATION                                     C0701300
C          SCR(I) = ALL BODIES OF NEST I CONTRIBUTING SIGNIFICANTLY     C0701400
C                   EQUATION                                            C0701500
```

```
C                  TO CORIOLIS CROSS COUPLING EFFECTS IN NEST I MOTION  C0701600
C         SMC(I) = ALL CONSTANT AND VARIABLE SPEED MOMENTUM WHEELS       C0701700
C                  IN NEST I                                            C0701800
C             SD = ALL CONTIGUOUS PAIRS OF BODIES (JCON(K),K) FOR WHICH C0701900
C                  DIRECTION COSINE TECHNIQUES ARE TO BE USED TO COMPUTE C0702000
C                  RELATIVE TRANSFORMATION MATRICES                     C0702100
C                  SD SHOULD CONTAIN ONLY LABELS FOR RIGID OR FLEXIBLE  C0702200
C                  BODIES                                               C0702300
C           SMAL = ALL CONTIGUOUS PAIRS OF BODIES (JCONK),K) FOR WHICH  C0702400
C                  SMALL ANGLE TECHNIQUES ARE TO BE USED TO COMPUTE     C0702500
C                  RELATIVE TRANSFORMATION MATRICES                     C0702600
C            SEU = ALL CONTIGUOUS PAIRS OF BODIES (JCON(K),K) FOR WHICH C0702700
C                  EULER ANGLE TECHNIQUES ARE TO BE USED TO COMPUTE     C0702800
C                  RELATIVE TRANSFORMATION MATRICES                     C0702900
C       SXM(I,K) = COMPUTED CODE WORD ARRAY                             C0703000
C                  THE COORDINATE AXIS IN BODY K ALONG WHICH THE I-TH   C0703100
C                  FREE OR LOCKED GIMBAL AXIS IS ALIGNED, SIGN IMPLIES  C0703200
C                  DIRECTION                                            C0703300
C                    'NOTE'  INTERNAL LOGIC ASSUMES THAT THE FREE       C0703400
C                    COORDINATE VECTORS AT HINGE POINTS AT WHICH SMALL  C0703500
C                    ANGLE OR EULER ANGLE TECHNIQUES ARE APPLIED ARE    C0703600
C                          PARALLEL TO THE BODY FIXED AXES DEFINED      C0703700
C                          AT THAT POINT                                C0703800
C            SXT = ALL COLUMNS OF THE SYSTEM INERTIA MATRIX OF PSUEDO-   C0703900
C                  INERTIA TENSORS WHICH HAVE TIME VARYING ELEMENTS.    C0704000
C                  (SINCE INERTIA MATRIX SYMMETRIC COLUMNS DESIGNATED   C0704100
C                  EXTEND DOWN ONLY TO DIAGONAL ELEMENT,COLUMN N+1 IS   C0704200
C                  ALWAYS ASSUMED TIME VARYING)                         C0704300
C                  ELEMENTS IN COLUMN K,K=1,2,...,N GIVE THE INERTIA    C0704400
C                  CONTRIBUTIONS OF THE NEST K-1 TO THE SYSTEM EQUATIONS C0704500
C                  OF MOTION                                            C0704600
C            SMV = ALL VARIABLE SPEED MOMENTUM WHEELS                   C0704700
C            SVD = ALL BODIES IN WHICH BODY FIXED VECTORS AND DYADS ARE C0704800
C                  TIME VARYING IN THE COMPUTATIONAL REFERENCE FRAME     C0704900
C         SFR(I) = SET OF FREE COORDINATE VECTOR LABELS OF THOSE VECTORS C0705000
C                  ABOUT OR ALONG WHICH DISPLACEMENT IS COMPUTED         C0705100
C         SLK(I) = SET OF LOCKED COORDINATE VECTOR LABELS OF THOSE VECTO C0705200
C                  ABOUT OR ALONG WHICH CONSTRAINT TORQUE IS COMPUTED    C0705300
C                  THIS CAPABILITY NOT AVAILABLE IN N-BOD2               C0705400
C         SMA(I) = SET OF MOMENTUM WHEEL LABELS OF THOSE WHEELS FOR      C0705500
C                  WHICH ANGULAR DISPLACEMENT IS TO BE COMPUTED          C0705600
C                                                                        C0705700
C         INERF  = .TRUE.  COMPUTATION FRAME IS INERTIAL REFERENCE       C0705800
C                  .FALSE. COMPUTATION FRAME FIXED IN BODY I             C0705900
C                                                                        C0706000
C          SC(I) = ALL CAGED DEGREES OF FREEDOM                          C0706100
C            SCG = TOTAL NUMBER OF CAGED DEGREES OF FREEDOM              C0706200
C         TUG(J) = TIME AT WHICH DEGREE OF FREEDOM J UNCAGED             C0706300
C                                                                        C0706400
C                                                                        C0706500
C          NFLXB = TOTAL NUMBER OF FLEXIBLE BODIES                       C0706600
C           SFLX = SET OF ALL FLEXIBLE BODIES                            C0706700
C        SFXM(I) = TOTAL NUMBER OF FLEXIBLE BODY MODES USED FOR BODY I   C0706800
C                  (ZERO IF BODY I RIGID)                                C0706900
C          NMODS = TOTAL NUMBER OF FLEXIBLE BODY MODES USED FOR ENTIRE   C0707000
C                  SIMULATION                                            C0707100
C                                                                        C0707200
C     RESULTANT FLEXIBLE BODY MODAL PARAMETERS                           C0707300
C      FLA(I,MN) = MODE MN CENTER OF MASS                                C0707400
C      FLB(I,MN) = MASS MODAL MOMENT ABOUT HINGE POINT FOR               C0707500
```

```
C                 MODE MN DUE TC TRANSLATION OF MASS POINTS                     CC7C7600
C       FLC(I,MN) = INERTIA MODAL MOMENT FOR MODE MN DUE TO                     C07C7700
C                   ROTATION OF MASS POINTS                                     CC707800
C     FLD(I,J,MN) = PSUEDO-INERTIA COUPLING TENSOR FOR MODE MN                  C07C7900
C     FLJ(I,J,MN) = CENTRIPITAL COUPLING TENSOR FOR MODE MN                     C0708000
C                                                                               C0708100
C               FOR MODE M OF BODY N,MODAL NUMBER MN IS                         C0708200
C                MN = SFXM(1)+...+SFXM(N-1)+M                                   C0708300
C                                                                               C0708400
C                 FOR ELASTIC COORDINATE MN                                     C0708500
C    THA(NFER+MN) = INITIAL GENERALIZED DISPLACEMENT, ELASTIC COORD. MN         CC7C8600
C   THAD(NFER+MN) = INITIAL GENERALIZED RATE, ELASTIC COORD. MN                 C0708700
C            NFER = TOTAL NUMBER OF FREE COORDINATE VECTORS                     C0708800
C                                                                               C0708900
C          SFCC = SET OF ALL FLEXIBLE BODIES FOR WHICH CENTRIPETAL AND          C0709000
C                 CORIOLIS EFFECTS SIGNIFICANT IN DEFORMATION EQUATION          C0709100
C       SCXC(MN) = SET OF ALL MODES N WHICH SIGNIFICANTLY CONTRIBUTE            C0709200
C                  TO THE CENTRIPETAL OR CORIOLIS CROSS COUPLING EFFECTS        C0709300
C                  IN THE EQUATION FOR MODE WITH MODE NUMBER MN                 C0709400
C                                                                               C0709500
C            RESULTANT MODE COUPLING COEFFICIENTS FOR MODE M AND N              CC7C9600
C       FCF(M,N) = MODE CROSS COUPLING TENSOR CENTRIPETAL ACC. EFFECTS          C0709700
C                  (IN MOST CASES FCF(M,N); M=N CAN HAVE SIGNIFICANT            C0709800
C                  NON-ZERO TERMS)                                              CC7C9900
C       FCK(M,N) = MODE CROSS COUPLING VECTOR FOR CORIOLIS ACC. EFFECTS         C0710000
C                                                                               C0710100
C                                                                               CC710200
C                                                                               C0710300
      IMPLICIT REAL*8(A-H,O-Z,$)                                                C0710400
      LOGICAL* FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)             C0710500
      LOGICAL          NSTART, LRTAPE                                           C0710600
C                                                                               C0710700
C                                                                               C0710800
      INTEGER*                                                                  C0710900
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , FCCN  ,         C0711000
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,         C0711100
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,         C0711200
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,         C0711300
     * SGL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,         C0711400
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,         C0711500
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMCDS , SFCC  , SCC   ,         C0711600
     * IINIT(1)      , IZINIT(1)     , SD    , SCXC(20)                        C0711700
C                                                                               C0711800
C                                                                               C0711900
      REAL*8                                                                    C0712000
     * ANGC  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,         C0712100
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                         C0712200
     * THACD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)              C0712300
C                                                                               C0712400
C                                                                               C0712500
      COMMON /CHEKS/          NSTART, LRTAPE                                    C0712600
C                                                                               C0712700
C                                                                               C0712800
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)                   C0712900
C                                                                               C0713000
C                                                                               C0713100
      COMMON /INTG/    AWORK (200)    ,                                         C0713200
     * CT1           , CT2           , CT3           , CT4           ,         C0713300
     * CT5           , FCON  (33)    , JCON  (10)    , LCON  (22)    ,         C0713400
     * MC    (10)    , NB1           , NBOD          , NCTC          ,         C0713500
```

```
     *  NFER              , NFKC              , NFRC              , NLCR              ,  C0713600
     *  NMV               , NMO               , NMOA              , NSVP              ,  C0713700
     *  NSVG              , PCON    (11)      , SD                , SFR     (33)      ,  C0713800
     *  SG                , SI      (55)      , SIG               , SL                ,  C0713900
     *  SLK     (33)      , SMA     (10)      , SCK     (11)      , SQF     (11)      ,  C0714000
     *  SQL     (11)      , SMV               , SR                , SSCN              ,  C0714100
     *  SSIX              , SVA               , SVB               , SVD               ,  C0714200
     *  SVI               , SVM               , SVP     (22)      , SVQ     (33)      ,  C0714300
     *  SXM     (3,10)    , SXT               , TORQ    (97)      , SMAL              ,  C0714400
     *  SEU               , NTQ               , SC      (33)      , SCG               ,  C0714500
     *  NFLXB             , SFLX              , SFXM    (10)      , NMODS             ,  C0714600
     *  SFCC              , SCC     (10)                                              ,  C0714700
C                                                                                       ,  C0714800
C                                                                                       ,  C0714900
      COMMON /INTGZ/                                                                    ,  C0715000
     *  SCNDUM            , SCN     (9)       , SCRDUM            , SCR     (9)       ,  C0715100
     *  SFKDUM            , SFK     (9)       , SIXDUM            , SIX     (9)       ,  C0715200
     *  SKDUM             , SK      (9)       , SPIDUM            , SPI     (9)       ,  C0715300
     *  SMCDUM            , SMC     (9)                                               ,  C0715400
C                                                                                       ,  C0715500
C                                                                                       ,  C0715600
      COMMON /REAL/                                                                     ,  C0715700
     *  CA      (3,10)    , CAC     (3,10)    , CLM     (10)      , CCMC    (3,11)    ,  C0715800
     *  DCMC    (3,11)    , ETC     (3,11)    , ETM     (33)      , FOMC    (3,11)    ,  C0715900
     *  GAM     (3,66)    , H                 , HM      (3,10)    , HMC     (3,10)    ,  C0716000
     *  HMOM    (10)      , PHI     (3,11)    , PLM     (10)      , QF      (3,33)    ,  C0716100
     *  QFC     (3,33)    , QL      (3,22)    , QLC     (3,22)    , RCMC    (3,11)    ,  C0716200
     *  T                 ,                   THA       (33)      , THAD    (33)      ,  C0716300
     *  THADW   (10)      , THAW    (10)      , XDIC    (3,3,66), XI      (3,3,10),  C0716400
     *  XIC     (3,3,10), XMAS    (10)      , XMM     (33,33)   , XMT     (3,3,10),  C0716500
     *  TLG     (33)      , FLA     (3,20)    , FLB     (3,20)    , FLC     (3,20)    ,  C0716600
     *  FLD     (3,3,20), FLJ     (3,3,20), CAD     (3,10)    , XID     (3,3,10),  C0716700
     *  FLIRC   (3,10)    , FLCRC   (3,10)    , FLAC    (3,20)    , FLQC    (3,20)    ,  C0716800
     *  FLOM    (20)      , ZETA    (20)      , FCF     (3,3,40), FCK     (3,40)    ,  C0716900
     *  TIMEND                                                                         ,  C0717000
C                                                                                       ,  C0717100
C                                                                                       ,  C0717200
      COMMON /REAL2/                                                                    ,  C0717300
     *  CBDUM   (1,3)     , CB      (3,10)    , CBCDUM(1,3)       , CBC     (3,10)    ,  C0717400
     *  XMCDUM(1,1,9)     , XMC     (3,3,10), CBN(3)                                  ,  C0717500
C                                                                                       ,  C0717600
C                                                                                       ,  C0717700
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))      ,               ,  C0717800
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,               ,  C0717900
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,               ,  C0718000
     *            (FLB(1,1),FLD(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),               ,  C0718100
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                       ,  C0718200
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,               ,  C0718300
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,               ,  C0718400
     *            (SCNDUM,IZINIT(1))         ,(TORQ(78),SCXC(1))                      ,  C0718500
C                                                                                       ,  C0718600
C                                                                                       ,  C0718700
      DIMENSION MESS(18)                                                                ,  C0718800
      INTEGER MCN(4), SFXMI, SIK                                                        ,  C0718900
      DIMENSION QFK(3,3)                                                                ,  C0719000
      INTEGER S1(10),S2(10),S3(10),SML,S4(33)                                           ,  C0719100
      LOGICAL CTAIN                                                                     ,  C0719200
C                                                                                       ,  C0719300
C                                                                                       ,  C0719400
      DATA INR,IBD,IAS/'INER','ECCY','ASET'/,                                           ,  C0719500
```

```
     *      IXM,IIC,ICE,ICO/'X   ','IXC ','CEN ','CCR '/,                  C0719600
     *      IEC,ICL,IVT,IFR,ILO/'ELLE','COLX','VTIN','FREE','LOCK'/,       C0719700
     *      ICF,IMA/'CFCR','MCMA'/,                                        C0719800
     *      IVA/'VARW'/,                                                   C0719900
     *      ISM/'SMAL'/,                                                   C0720000
     *      IEN/' END'/,                                                   C0720100
     *      NO,IBL,ICG/' NO ',' ','CAGE'/,                                 C0720200
     *      IFL,IMD/'FLEX','MODE'/                                         C0720300
C                                                                           C0720400
C     DEFAULT CONDITIONS                                                    C0720500
C                                                                           C0720600
C     SET UP OPTION DATA SETS MAKING NO ENGINEERING ASSUMPTIONS AND THAT    C0720700
C        COMPUTATION IS RELATIVE TO INERTIAL FRAME                          C0720800
C                                                                           C0720900
      INERF = .FALSE.                                                       C0721000
      DO 1 L=1,NBOD                                                         C0721100
      I = L-1                                                               C0721200
      SFK(I) = 0                                                            C0721300
      SPI(I) = SK(I)                                                        C0721400
      SIX(I) = SK(I)                                                        C0721500
      SCN(I) = SK(I)                                                        C0721600
    1 SCR(I) = SK(I)                                                        C0721700
      SXT = SK(0)                                                           C0721800
      SVC = SK(0)                                                           C0721900
      SMAL = 0                                                              C0722000
      SEC = 0                                                               C0722100
      NFRC = NFER                                                           C0722200
      NCTC = 0                                                              C0722300
      NMCA = 0                                                              C0722400
      NFKC = 0                                                              C0722500
      SCG = 0                                                               C0722600
      NB3 = 3*NB1                                                           C0722700
      DO 84 I=1,NB3                                                         C0722800
      SLK(I) = 0                                                            C0722900
   84 SC(I) = 0                                                             C0723000
      SFLX = 0                                                              C0723100
      NMCDS = 0                                                             C0723200
      DO 79 I=1,NBOD                                                        C0723300
   79 SFXM(I) = 0                                                           C0723400
      NFLXB = 0                                                             C0723500
      SFCC = 0                                                              C0723600
      NB2 = 2*NBOD                                                          C0723700
      DO 89 I=1,NB2                                                         C0723800
   89 SCXC(I) = 0                                                           C0723900
      DO 2  I=1,NFER                                                        C0724000
    2 SFR(I) = I                                                            C0724100
      DO 60  I=1,NMO                                                        C0724200
      SMA(I) = 0                                                            C0724300
   60 S1(I) = I                                                             C0724400
      CALL COMPAC(S1,NMC,SMV)                                               C0724500
      NMV = NMC                                                             C0724600
C                                                                           C0724700
C     NOTE                                                                  C0724800
C        CORIOLIS EFFECTS ONLY FOR LINEAR OSCILLATORS                       C0724900
C        INERTIA CROSS COUPLING ONLY FOR RIGID BODIES                       C0725000
C        FOR LINEAR OSCILLATOR TRANSFORMATION MATRICES FIXED                C0725100
C     SUBTRACT THESE FROM NON-AUGMENTED SETS CONSTRUCTED                    C0725200
      DO 5 L=1,NBOD                                                         C0725300
      I = L-1                                                               C0725400
      CALL UNPAC(S1,N1,SCR(I))                                              C0725500
```

```
      CALL UNPAC(S2,N2,SL)                                              C0725600
      N3= C                                                             C0725700
      IF(N1.EC.0) GO TO 666                                             C0725800
      CO 6 J=1,N1                                                       C0725900
      IF(N2.EC.0) GO TO 667                                             C0726000
      CO 668 K=1,N2                                                     C0726100
      IF(S1(J).NE.S2(K))  GO TC 668                                     C0726200
      N3 = N3 + 1                                                       C0726300
      S3(N3) = S1(J)                                                    C0726400
  668 CONTINUE                                                          C0726500
  667 CONTINUE                                                          C0726600
    6 CONTINUE                                                          C0726700
  666 CONTINUE                                                          C0726800
      CALL COMPAC(S3,N3,SCR(I))                                         C0726900
    5 CONTINUE                                                          C0727000
C                                                                       C0727100
      CO 1C L=1,NBOD                                                    C0727200
      I = L-1                                                           C0727300
      CALL UNPAC(S1,N1,SIX(I))                                          C0727400
      CALL UNPAC(S2,N2,SR)                                              C0727500
      N3 = C                                                            C0727600
      IF(N1.EC.C) GO TO 1111                                            C0727700
      CO 11  J=1,N1                                                     C0727800
      IF(N2.EC.0) GO TO 1112                                            C0727900
      CO 1113 K=1,N2                                                    C0728000
      IF(S1(J).NE.S2(K)) GO TO 1113                                     C0728100
      N3 = N3 + 1                                                       C0728200
      S3(N3) = S1(J)                                                    C0728300
 1113 CONTINUE                                                          C0728400
 1112 CONTINUE                                                          C0728500
   11 CONTINUE                                                          C0728600
 1111 CONTINUE                                                          C0728700
      CALL CCMPAC(S3,N3,SIX(I))                                         C0728800
   1C CONTINUE                                                          C0728900
C                                                                       C0729000
C                                                                       C0729100
   12 READ 1CC, ICD1,ICD2,ICD3,I,NSET,(S1(J),J=1,10)                    C0729200
      CT1 = CT1 + 1                                                     C0729300
      IF(ICD1.EQ.IBL) GC TO 2C                                          C0729400
      IF(ICD1.EQ.NU)  GO TO 21                                          C0729500
      IF(ICD1.EQ.IEN) GC TO 21                                          C0729600
   2C IF(ICD2.EQ.IAS) GC TO 22                                          C0729700
      IF(ICD2.EQ.IEU) GO TO 23                                          C0729800
      IF(ICD2.EQ.ICL) GC TO 24                                          C0729900
      IF(ICD2.EQ.IVT) GC TO 25                                          C0730000
      IF(ICD2.EQ.IFR) GC TO 26                                          C0730100
      IF(ICD2.EQ.ILG) GC TO 27                                          C0730200
      IF(ICD2.EQ.IMA) GC TO 28                                          C0730300
      IF(ICD2.EQ.ICF) GC TO 29                                          C0730400
      IF(ICD2.EQ.INR) GC TO 30                                          C0730500
      IF(ICD2.EQ.IBD) GC TO 31                                          C0730600
      IF(ICD2.EQ.IVA) GC TO 62                                          C0730700
      IF(ICD2.EQ.ISM) GC TO 69                                          C0730800
      IF(ICD2.EQ.ICG) GC TO 77                                          C0730900
      IF(ICD2.EQ.IFL) GC TO 80                                          C0731000
      IF(ICD2.EQ.IMD) GC TO 90                                          C0731100
   37 PRINT 20C, CT1,ICD1,ICD2,ICD3                                     C0731200
      FG3 = .FALSE.                                                     C0731300
      RETURN                                                            C0731400
   3C INERF = .TRUE.                                                    C0731500
```

```
      GO TO 12                                                          C0731600
   31 INERF = .FALSE.                                                   C0731700
      GO TO 12                                                          C0731800
   22 IF(ICD3.EQ.IXM) GO TO 32                                          C0731900
      IF(ICD3.EQ.IIC) GO TO 33                                          C0732000
      IF(ICD3.EQ.ICE) GO TO 34                                          C0732100
      IF(ICD3.EQ.ICO) GO TO 35                                          C0732200
      GO TO 37                                                          C0732300
   23 CALL COMPAC(S1,NSET,SEL)                                          C0732400
      GO TO 12                                                          C0732500
C                                                                       C0732600
   80 NFLXB = NSET                                                      C0732700
      DO 81 I=1,NBOD                                                    C0732800
      SFXM(I) = S1(I)                                                   C0732900
   81 NMODS = NMODS + SFXM(I)                                           C0733000
C     CREATE THE SET SFLX                                               C0733100
      II = 0                                                            C0733200
      DO 88 I=1,NBOD                                                    C0733300
      S1(I) = 0                                                         C0733400
      IF(SFXM(I).EQ.0) GO TO 88                                         C0733500
      II = II + 1                                                       C0733600
      S1(II) = I                                                        C0733700
   88 CONTINUE                                                          C0733800
      CALL COMPAC(S1,NFLXB,SFLX)                                        C0733900
C     READ IN ALL FLEXIBLE BODY MODAL DATA                              C0734000
      CALL UNPAC(S1,NS1,SFLX)                                           C0734100
      MN = 0                                                            C0734200
      MMM = 0                                                           C0734300
      PRINT 201                                                         C0734400
      PRINT 242, (S1(I),I=1,NS1)                                        C0734500
      DO 85 II=1,NS1                                                    C0734600
      I = S1(NS1+1-II)                                                  C0734700
      READ 103, N,(MESS(J),J=1,18)                                      C0734800
      PRINT 241, N, (MESS(J),J=1,18)                                    C0734900
      IF(I.EQ.N) GO TO 86                                               C0735000
      PRINT 232                                                         C0735100
      EG3 = .FALSE.                                                     C0735200
      RETURN                                                            C0735300
   86 SFXMI = SFXM(I)                                                   C0735400
      DO 87 M=1,SFXMI                                                   C0735500
      MMM = MMM + 1                                                     C0735600
      MN = MN+1                                                         C0735700
      READ 104, FLOM(MN),ZETA(MN)                                       C0735800
      READ 104, THA(NFER+MN),THAD(NFER+MN)                              C0735900
      READ 104, (FLA(K,MN),K=1,3)                                       C0736000
      READ 104, (FLB(K,MN),K=1,3)                                       C0736100
      READ 104, (FLC(K,MN),K=1,3)                                       C0736200
      READ 104, ((FLD(K,L,MN),L=1,3),K=1,3)                             C0736300
      READ 104, ((FLJ(K,L,MN),L=1,3),K=1,3)                             C0736400
      PRINT 233, MN, MN,FLOM(MN),MN,ZETA(MN)                            C0736500
      NFMN = NFER + MN                                                  C0736600
      PRINT 235, NFMN,THA(NFMN),NFMN,THAD(NFMN)                         C0736700
      PRINT 234, MN,(FLA(K,MN),K=1,3),I                                 C0736800
      PRINT 243                                                         C0736900
      PRINT 235, MN,(FLE(K,MN),K=1,3),I                                 C0737000
      PRINT 243                                                         C0737100
      PRINT 236, MN,(FLC(K,MN),K=1,3),I                                 C0737200
      PRINT 243                                                         C0737300
      PRINT 238,     (FLD(1,L,MN),L=1,3)                                C0737400
      PRINT 237, MN,(FLD(2,L,MN),L=1,3),I                               C0737500
```

```
      PRINT 238,     (FLC(3,L,MM),L=1,3)                                  C0737600
      PRINT 243                                                           C0737700
      PRINT 238,     (FLJ(1,L,MM),L=1,3)                                  C0737800
      PRINT 240, MM,(FLJ(2,L,MM),L=1,3),I                                 C0737900
      PRINT 238,     (FLJ(3,L,MM),L=1,3)                                  C0738000
      PRINT 215                                                           C0738100
      IF(MMM.NE.2) GO TO 87                                               C0738200
      PRINT 201                                                           C0738300
      MMM = 0                                                             C0738400
   87 CONTINUE                                                            C0738500
   88 CONTINUE                                                            C0738600
      GO TO 12                                                            C0738700
C                                                                         C0738800
C     CONSTRUCT CENTRIPETAL AND CORIOLIS MODE COUPLING CODE WORD          C0738900
   90 CONTINUE                                                            C0739000
C     CHECK IF 'FLEXIBLE' READ YET                                        C0739100
      IF(NMODS.NE.0) GO TO 91                                             C0739200
  154 PRINT 232                                                           C0739300
      FG3=.FALSE.                                                         C0739400
      RETURN                                                              C0739500
   91 DO 92 I=1,NMODS                                                     C0739600
   92 SCXC(I) = 0                                                         C0739700
      DO 93 I=1,NBOD                                                      C0739800
   93 S2(I) = 0                                                           C0739900
      NS2 = 0                                                             C0740000
      DO 94 J=1,NBOD                                                      C0740100
      IF(S1(J).EQ.0) GO TO 94                                             C0740200
      NS2 = NS2+1                                                         C0740300
      S2(NS2) = J                                                         C0740400
   94 CONTINUE                                                            C0740500
      CALL COMPAC(S2,NS2,SFCC)                                            C0740600
      PRINT 201                                                           C0740700
      PRINT 244, (S2(I),I=1,NS2)                                          C0740800
C     READ IN MODE COUPLING DATA, SET UP COUNTERS                         C0740900
      MN = 0                                                              C0741000
      KF = 0                                                              C0741100
      DO 150 K=1,NBOD                                                     C0741200
      IF(SFXM(K).EQ.0) GO TO 150                                          C0741300
      IF(S1(K).NE.0) GO TO 151                                            C0741400
C     NO CROSS COUPLING TERMS, UPDATE MODE NUMBER COUNTER MN              C0741500
      MN = MN+SFXM(K)                                                     C0741600
      GO TO 150                                                           C0741700
  151 S1K = S1(K)                                                         C0741800
      DO 152 I=1,S1K                                                      C0741900
      READ 105, MB,NB,KE                                                  C0742000
      IF(K.NE.KB) GO TO 154                                               C0742100
      MNN = MN + NB                                                       C0742200
      CALL UNPAC(S2,NS2,SCXC(MNN))                                        C0742300
      NS2 = NS2+ 1                                                        C0742400
      S2(NS2) = MB                                                        C0742500
      CALL COMPAC(S2,NS2,SCXC(MNN))                                       C0742600
      KF = KF+1                                                           C0742700
      READ 104, ((FCF(II,JJ,KF),JJ=1,3),II=1,3)                           C0742800
      PRINT 243                                                           C0742900
      PRINT 246,           (FCF(1,JJ,KF),JJ=1,3)                          C0743000
      PRINT 247, MB,NB,KE,(FCF(2,JJ,KF),JJ=1,3),K,KF                     C0743100
      PRINT 246,           (FCF(3,JJ,KF),JJ=1,3)                          C0743200
      PRINT 243                                                           C0743300
      READ 104, (FCK(II,KF),II=1,3)                                       C0743400
      PRINT 248, MB,NB,KE,(FCK(JJ,KF),JJ=1,3),K,KF                       C0743500
```

```
      PRINT 243                                                         C0743600
  152 CONTINUE                                                          C0743700
C       CHECK AND OUTPUT INTEGER ARRAYS FOR BODY K                      C0743800
      KKF = KF-S1(K)                                                    C0743900
      SFXMI = SFXM(K)                                                   C0744000
      DO 153 I=1,SFXMI                                                  C0744100
      MN = MN+1                                                         C0744200
      CALL UNPAC(S2,NS2,SCXC(MN))                                       C0744300
      IF(NS2.NE.0) GO TO 155                                            C0744400
      PRINT 243                                                         C0744500
      PRINT 245, MN                                                     C0744600
      GO TO 153                                                         C0744700
  155 DO 156 J=1,NS2                                                    C0744800
      MB = S2(NS2+1-J)                                                  C0744900
      KKF = KKF+1                                                       C0745000
      PRINT 250, K,MB,MN,KKF                                            C0745100
  156 CONTINUE                                                          C0745200
  153 CONTINUE                                                          C0745300
  150 CONTINUE                                                          C0745400
      GO TO 12                                                          C0745500
C                                                                       C0745600
   24 CALL COMPAC(S1,NSET,SXT)                                          C0745700
      GO TO 12                                                          C0745800
   25 CALL COMPAC(S1,NSET,SVC)                                          C0745900
      GO TO 12                                                          C0746000
   62 CALL COMPAC(S1,NSET,SMV)                                          C0746100
      NMV = NSET                                                        C0746200
      GO TO 12                                                          C0746300
   65 CALL COMPAC(S1,NSET,SMAL)                                         C0746400
      GO TO 12                                                          C0746500
   77 SCG = NSET                                                        C0746600
      DO 99 I=1,SCG                                                     C0746700
      READ 101, SC(I),TEM                                               C0746800
   99 TUC(SC(I)) = TEM                                                  C0746900
C       CHECK IC ON THAC(SC(I)); MUST BE ZERO, RESET IF NOT.            C0747000
C         CODING FOR IMPULSE EFFECT DUE TO NON-ZERO THAO NOT INCLUDED   C0747100
      DO 78 I=1,SCG                                                     C0747200
      IF(THAO(SC(I)).EQ.0.0) GO TO 78                                   C0747300
      THAC(SC(I)) = 0.0                                                 C0747400
      PRINT 230, SC(I)                                                  C0747500
   78 CONTINUE                                                          C0747600
      GO TO 12                                                          C0747700
   26 CT2 = CT2+1                                                       C0747800
      JDONE = CT2+NSET-1                                                C0747900
      IF(CT2.GT.JDONE) GO TO 5001                                       C0748000
      DO 36 J=CT2,JDONE                                                 C0748100
   36 SFR(J) = S1(J+1-CT2)                                              C0748200
 5001 CONTINUE                                                          C0748300
      CT2 = CT2+NSET-1                                                  C0748400
      GO TO 12                                                          C0748500
   27 CT3 = CT3+1                                                       C0748600
      JDONE = CT3+NSET-1                                                C0748700
      IF(CT3.GT.JDONE) GO TO 5002                                       C0748800
      DO 39 J=CT3,JDONE                                                 C0748900
   39 SLK(J) = S1(J+1-CT3)                                              C0749000
 5002 CONTINUE                                                          C0749100
      CT3 = CT3+NSET-1                                                  C0749200
      GO TO 12                                                          C0749300
   28 CT4 = CT4+1                                                       C0749400
      JDONE = CT4+NSET-1                                                C0749500
```

```
      IF(CT4.GT.JDONE) GO TO 5003                                       C0749600
      DO 40 J=CT4,JDONE                                                 C0749700
   40 SMA(J) = SI(J+1-CT4)                                              C0749800
 5003 CONTINUE                                                          C0749900
      CT4 = CT4+NSET-1                                                  C0750000
      GO TO 12                                                          C0750100
   29 CTS = CTS+1                                                       C0750200
      JDONE = CTS+NSET-1                                                C0750300
      IF(CTS.GT.JDONE) GO TO 5004                                       C0750400
      DO 41 J=CTS,JDONE                                                 C0750500
   41 SPK(J) = SI(J+1-CTS)                                              C0750600
 5004 CONTINUE                                                          C0750700
      CTS = CTS+NSET-1                                                  C0750800
      GO TO 12                                                          C0750900
   32 CALL COMPAC(SI,NSET,SPI(I))                                       C0751000
      GO TO 12                                                          C0751100
   33 CALL COMPAC(SI,NSET,SIX(I))                                       C0751200
      GO TO 12                                                          C0751300
   34 CALL COMPAC(SI,NSET,SCN(I))                                       C0751400
      GO TO 12                                                          C0751500
   35 CALL COMPAC(SI,NSET,SCR(I))                                       C0751600
      GO TO 12                                                          C0751700
   21 CT1 = 0                                                           C0751800
      IF(CT2.EQ.0) GO TO 42                                             C0751900
      NPRC = CT2                                                        C0752000
      CT2 = 0                                                           C0752100
   42 IF(CT3.EQ.0) GO TO 43                                             C0752200
      NCTC = CT3                                                        C0752300
      CT3 = 0                                                           C0752400
   43 IF(CT4.EQ.0) GO TO 44                                             C0752500
      NMCA = CT4                                                        C0752600
      CT4 = 0                                                           C0752700
   44 IF(CTS.EQ.0) GO TO 45                                             C0752800
      NPKC = CTS                                                        C0752900
      CTS = 0                                                           C0753000
   45 CONTINUE                                                          C0753100
C                                                                       C0753200
C                                                                       C0753300
C     OBTAIN ELEMENTS OF ARRAY SXM                                      C0753400
C      INITIALIZE BOTH SXM AND TH ARRAYS                                C0753500
      DO 3  K=1,NBOD                                                    C0753600
      DO 3  I=1,3                                                       C0753700
      SXM(I,K) = 0                                                      C0753800
    3 CONTINUE                                                          C0753900
C     DEFINE CONTIGUOUS PAIRS FOR WHICH DIRECTION COSINES ARE TO BE USEDC0754000
      SD = SR-SMAL-SEU                                                  C0754100
C      DEFINE CONTIGUOUS PAIRS FOR WHICH SMALL ANGLE OR EULER ANGLE     C0754200
C        TECHNIQUES TO BE USED                                          C0754300
      SML = SR - SD                                                     C0754400
      CALL UNPAC(SI,NSI,SML)                                            C0754500
C     CYCLE THROUGH ELEMENTS OF SML TO DEFINE SXM ARRAY                 C0754600
      IF(NSI.EQ.0) GO TO 5080                                           C0754700
      DO 4  KK=1,NSI                                                    C0754800
      K = SI(KK)                                                        C0754900
C     DEFINE NOMINAL STATE TRANSFORMATION FROM BODY JCON(K) TO BODY K   C0755000
      CALL TRNSPS  (XMT(1,1,K))                                         C0755100
C     PICK OUT GIMBAL CONFIGURATION AT HINGE POINT K-1                  C0755200
      IGOTO = PCON(K) + 1                                               C0755300
      GO TO (13,14,15,16),IGOTO                                         C0755400
   16 CALL TRNSPS  (XMT(1,1,K))                                         C0755500
```

```
      GO TO 4                                                          C0755600
   15 NGA = 3                                                          C0755700
      M = SQF(K)                                                       C0755800
      L = SQL(K)                                                       C0755900
      CALL VECTRN (QF(1,M),XMT(1,1,K),QFK(1,1))                        C0756000
      DO 75  I=1,3                                                     C0756100
   75 QFK(I,2) = QL(I,L)                                               C0756200
      CALL VECROS  (QFK(1,1),QFK(1,2),QFK(1,3))                        C0756300
      GO TO 7                                                          C0756400
   14 NGA = 3                                                          C0756500
      M = SQF(K)                                                       C0756600
      M1 = M+1                                                         C0756700
      CALL VECTRN (QF(1,M),XMT(1,1,K),QFK(1,1))                        C0756800
      DO 8  I=1,3                                                      C0756900
    8 QFK(I,2) = QF(I,M1)                                              C0757000
      CALL VECROS (QFK(1,1),QFK(1,2),QFK(1,3))                         C0757100
      GO TO 7                                                          C0757200
   13 NGA = 3                                                          C0757300
      M = SQF(K)                                                       C0757400
      M1 = M+1                                                         C0757500
      M2 = M+2                                                         C0757600
      CALL VECTRN (QF(1,M),XMT(1,1,K),QFK(1,1))                        C0757700
      DO 9  I=1,3                                                      C0757800
    9 QFK(I,3) = QF(I,M2)                                              C0757900
      CALL VECROS (QFK(1,3),QFK(1,1),QFK(1,2))                         C0758000
      GO TO 7                                                          C0758100
    7 IF(NGA.EQ.0) GO TO 16                                            C0758200
      DO 17 N=1,NGA                                                    C0758300
      DO 17  I=1,3                                                     C0758400
      IF(QFK(I,N).NE.0) GO TO 18                                       C0758500
      GO TO 17                                                         C0758600
   18 IF(QFK(I,N).NE.1) GO TO 19                                       C0758700
      SXM(N,K) = I                                                     C0758800
      GO TO 17                                                         C0758900
   19 IF(QFK(I,N).NE.-1) GO TO 58                                      C0759000
      SXM(N,K) = -I                                                    C0759100
      GO TO 17                                                         C0759200
   58 CONTINUE                                                         C0759300
C     COME HERE ONLY IF FREE COORDINATE AXES NOT ALIGNED WITH          C0759400
C     BODY K FIXED AXES IN NOMINAL STATE.                              C0759500
C         SMALL ANGLE OR EULER ANGLE METHODS CANNOT BE USED. INCLUSION C0759600
C         OF THIS CAPABILITY OF LIMITED VALUE SINCE IT WOULD SACRIFICE C0759700
C         COMPUTATION SPEED AND MEMORY STORAGE.                        C0759800
C     PUT K BACK IN SD AND DELETE IT FROM SEU OR SMAL                  C0759900
      S2(1) = K                                                        C0760000
      CALL COMPAC(S2,1,NK)                                             C0760100
      SD = SD + NK                                                     C0760200
      CALL UNPAC(S3,NS3,SEU)                                           C0760300
      IF(NS3.EQ.0) GO TO 5020                                          C0760400
      DO 71  JJ=1,NS3                                                  C0760500
      J = S3(JJ)                                                       C0760600
      IF(J.NE.K) GO TO 71                                              C0760700
      SEL = SEL - NK                                                   C0760800
      GO TO 16                                                         C0760900
   71 CONTINUE                                                         C0761000
 5020 CONTINUE                                                         C0761100
      CALL UNPAC(S3,NS3,SMAL)                                          C0761200
      IF(NS3.EQ.0) GO TO 16                                            C0761300
      DO 72  JJ=1,NS3                                                  C0761400
      J = S3(JJ)                                                       C0761500
```

```
      IF(J.NE.K) GO TO 72                                              C0761600
      SMAL = SMAL - NK                                                 C0761700
      GO TO 16                                                         C0761800
   72 CONTINUE                                                         C0761900
      GO TO 16                                                         C0762000
   17 CONTINUE                                                         C0762100
      GO TO 16                                                         C0762200
    4 CONTINUE                                                         C0762300
 5080 CONTINUE                                                         C0762400
C                                                                      C0762500
C     MAKE SURE ALL ANGLES CALLED FOR BY SAM WILL BE COMPUTED          C0762600
      DO 36 K=1,NBOD                                                   C0762700
      IF(SAM(1,K).EQ.0) GO TO 36                                       C0762800
      MDONE = 3-PCON(K)                                                C0762900
      DO 49 LL=1,MDONE                                                 C0763000
      L = LL-1                                                         C0763100
      M = SQF(K) + L                                                   C0763200
      IF(NFRC.EQ.0) GO TO 5021                                         C0763300
      DO 57  N=1,NFRC                                                  C0763400
      IF(M.EQ.SFR(N)) GO TO 49                                         C0763500
   57 CONTINUE                                                         C0763600
 5021 CONTINUE                                                         C0763700
C     COME HERE IF M NOT IN SFR                                        C0763800
      NFRC = NFRC+1                                                    C0763900
      SFR(NFRC) = M                                                    C0764000
   49 CONTINUE                                                         C0764100
   36 CONTINUE                                                         C0764200
C                                                                      C0764300
C     MAKE SURE TRANSLATION COMPONENTS OF C.M. OF BODY 1 (HINGE POINT 0)C0764400
C      ARE COMPUTED SO AS TO BE ABLE TO DEFINE CB(I), NEEDED FOR INERTIAC0764500
C      ANGULAR MOMENTUM AND ENERGY CALCULATION                         C0764600
      MDONI = SQF(NB1)                                                 C0764700
      MDONE = SQF(NB1)+2-PCON(NB1)                                     C0764800
      IF(MDONI.EQ.0) GO TO 5033                                        C0764900
      DO 73 M=MDONI,MDONE                                              C0765000
      IF(NFRC.EQ.0) GO TO 5022                                         C0765100
      DO 74  N=1,NFRC                                                  C0765200
      IF(M.EQ.SFR(N)) GO TO 73                                         C0765300
   74 CONTINUE                                                         C0765400
 5022 CONTINUE                                                         C0765500
      NFRC = NFRC+1                                                    C0765600
      SFR(NFRC) = M                                                    C0765700
   73 CONTINUE                                                         C0765800
 5033 CONTINUE                                                         C0765900
C                                                                      C0766000
C     MAKE SURE ALL POINT MASS COORDINATES IN SQF ARRAY                C0766100
      CALL UNPAC(S1,N1,SL)                                             C0766200
      IF(N1.EQ.0) GO TO 5018                                           C0766300
      DO 96 II=1,N1                                                    C0766400
      I = S1(II)                                                       C0766500
      MDONI = SQF(I)                                                   C0766600
      MDONE = SQF(I) + 2 - PCON(I)                                     C0766700
      IF(MDONI.EQ.0) GO TO 5016                                        C0766800
      DO 97 M=MDONI,MDONE                                              C0766900
      IF(NFRC.EQ.0) GO TO 5017                                         C0767000
      DO 98 N=1,NFRC                                                   C0767100
      IF(M.EQ.SFR(N)) GO TO 97                                         C0767200
   98 CONTINUE                                                         C0767300
 5017 CONTINUE                                                         C0767400
      NFRC = NFRC + 1                                                  C0767500
```

```
      SFR(NFRC) = M                                                    C0767600
   97 CONTINUE                                                         C0767700
 5016 CONTINUE                                                         C0767800
   96 CONTINUE                                                         C0767900
 5018 CONTINUE                                                         C0768000
C                                                                      C0768100
C     FIND ALL MOMENTUM WHEELS IN THE NEST K-1; K=1,2,....,NBOD        C0768200
      DO 66 II=1,NBOD                                                  C0768300
      I=II-1                                                           C0768400
      CALL UNPAC(S1,N1,SK(I))                                          C0768500
      DO 67  J=1,10                                                    C0768600
   67 S2(J) = 0                                                        C0768700
      N2 = 0                                                           C0768800
      IF(NMO.EQ.0) GO TO 5023                                          C0768900
      DO 68  J=1,NMO                                                   C0769000
      IF(.NOT.CTAIN(MO(J),S1,N1)) GO TO 68                             C0769100
      N2 = N2 + 1                                                      C0769200
      S2(N2) = J                                                       C0769300
   68 CONTINUE                                                         C0769400
 5023 CONTINUE                                                         C0769500
      CALL COMPAC(S2,N2,SMC(I))                                        C0769600
   66 CONTINUE                                                         C0769700
C                                                                      C0769800
C     IF COMPUTING FRAME BODY I DELETE LABEL I FROM SVD                C0769900
      CALL UNPAC(S1,NSET,SVD)                                          C0770000
      IF(INERF.OR.S1(NSET).EQ.0) GO TO 76                              C0770100
      NSET = NSET - 1                                                  C0770200
      CALL COMPAC(S1,NSET,SVD)                                         C0770300
   76 CONTINUE                                                         C0770400
C                                                                      C0770500
C                        ALL TRUNCATED SUMMATIONS DEFINED              C0770600
C                             PRINT THEM OUT                           C0770700
C                                                                      C0770800
      PRINT 201                                                        C0770900
      IF(INERF) GO TO 46                                               C0771000
      PRINT 203                                                        C0771100
      GO TO 47                                                         C0771200
   46 PRINT 202                                                        C0771300
   47 PRINT 204                                                        C0771400
      PRINT 205                                                        C0771500
      PRINT 206                                                        C0771600
      DO 48 II=1,NBOD                                                  C0771700
      I=II-1                                                           C0771800
      CALL UNPAC(S1,NSET,SK(I))                                        C0771900
      PRINT 207, I,(S1(J),J=1,NSET)                                    C0772000
      CALL UNPAC(S1,NSET,SMC(I))                                       C0772100
      PRINT 222, I,(S1(J),J=1,NSET)                                    C0772200
      PRINT 224                                                        C0772300
      CALL UNPAC(S1,NSET,SPI(I))                                       C0772400
      PRINT 209,    (S1(J),J=1,NSET)                                   C0772500
      CALL UNPAC(S1,NSET,SIX(I))                                       C0772600
      PRINT 210,    (S1(J),J=1,NSET)                                   C0772700
      CALL UNPAC(S1,NSET,SCN(I))                                       C0772800
      PRINT 211,    (S1(J),J=1,NSET)                                   C0772900
      CALL UNPAC(S1,NSET,SCR(I))                                       C0773000
      PRINT 212,    (S1(J),J=1,NSET)                                   C0773100
   48 PRINT 225, I                                                     C0773200
      PRINT 201                                                        C0773300
      CALL UNPAC(S1,NSET,SD)                                           C0773400
      IF(NSET.EQ.0) GO TO 5005                                         C0773500
```

```
      DO 5C  I=1,NSET                                                  C0773600
    5C PRINT 214, S1(I),JCON(S1(I))                                    C0773700
      PRINT 215                                                        C0773800
 5005 CONTINUE                                                         C0773900
      CALL UNPAC(S1,NSET,SMAL)                                         C0774000
      IF(NSET.EQ.0) GO TO 5006                                         C0774100
      DO 55  I =1,NSET                                                 C0774200
    55 PRINT 226, S1(I),JCON(S1(I))                                    C0774300
      PRINT 215                                                        C0774400
 5006 CONTINUE                                                         C0774500
      CALL UNPAC(S1,NSET,SEU)                                          C0774600
      IF(NSET.EQ.0) GO TO 5007                                         C0774700
      DO 7C  I=1,NSET                                                  C0774800
    7C PRINT 229, S1(I),JCON(S1(I))                                    C0774900
      PRINT 215                                                        C0775000
 5007 CONTINUE                                                         C0775100
      CALL UNPAC(S1,NSET,SXT)                                          C0775200
      IF(NSET.EQ.0) GO TO 5008                                         C0775300
      DO 51  I=1,NSET                                                  C0775400
    51 PRINT 216,S1(I)                                                 C0775500
      PRINT 215                                                        C0775600
 5008 CONTINUE                                                         C0775700
      CALL UNPAC(S1,NSET,SVD)                                          C0775800
      IF(NSET.EQ.0) GO TO 5009                                         C0775900
      DO 52  I=1,NSET                                                  C0776000
    52 PRINT 217,S1(I)                                                 C0776100
      PRINT 215                                                        C0776200
 5009 CONTINUE                                                         C0776300
C     PUT ARRAY SFR IN SEQUENTIAL ORDER                                C0776400
      DO 61  I=1,NB3                                                   C0776500
    61 S4(I) = C                                                       C0776600
      IF(NFRC.EQ.0) GO TO 501C                                         C0776700
      DO 63  I=1,NFRC                                                  C0776800
    63 S4(SFR(I)) = 1                                                  C0776900
 501C CONTINUE                                                         C0777000
      K = C                                                            C0777100
      DO 64  I=1,NB3                                                   C0777200
      IF(S4(I).EQ.C) GO TO 64                                          C0777300
      K = K+1                                                          C0777400
      SFR(K) = I                                                       C0777500
    64 CONTINUE                                                        C0777600
      IF(NFRC.EQ.0) GO TO 5011                                         C0777700
      DO 53  I=1,NFRC                                                  C0777800
    53 PRINT 218, SFR(I)                                               C0777900
      PRINT 215                                                        C0778000
 5011 CONTINUE                                                         C0778100
      IF(NCTC.EQ.0) GO TO 5012                                         C0778200
      DO 54  I=1,NCTC                                                  C0778300
    54 PRINT 219, SLK(I)                                               C0778400
      PRINT 215                                                        C0778500
 5012 CONTINUE                                                         C0778600
      IF(NFKC.EQ.0) GO TO 5013                                         C0778700
      DO 55  I=1,NFKC                                                  C0778800
    55 PRINT 220, SFK(I)                                               C0778900
      PRINT 215                                                        C0779000
 5013 CONTINUE                                                         C0779100
      CALL UNPAC(S1,NSET,SMV)                                          C0779200
      IF(NSET.EQ.0) GO TO 5014                                         C0779300
      DO 65  I=1,NSET                                                  C0779400
    65 PRINT 213, S1(I)                                                C0779500
```

```
      PRINT 215                                                         CC779600
 5014 CONTINUE                                                          CC779700
      IF(NMOA.EQ.0) GO TO 5015                                          CC779800
      DO 56  I=1,NMOA                                                   CC779900
   56 PRINT 221, SMA(I)                                                 CC780000
      PRINT 215                                                         CC780100
 5015 CONTINUE                                                          CC780200
      IF(SCG.EQ.0) GO TO 83                                             CC780300
      DO 82 I=1,SCG                                                     CC780400
   82 PRINT 231, SC(I),TUG(SC(I))                                       CC780500
   83 CONTINUE                                                          CC780600
      CT1 = 0                                                           CC780700
  100 FORMAT (3A4,I3,11I5)                                              CC780800
  101 FORMAT (I5,D15.5)                                                 CC780900
  102 FORMAT (I5,2D15.5)                                                CC781000
  103 FORMAT (I5,18A4)                                                  CC781100
  104 FORMAT (3D15.5)                                                   CC781200
  105 FORMAT (5I5)                                                      CC781300
  200 FORMAT (' IDENTIFICATION CODE NOT RECOGNIZED IN SUBROUTINE INCPT. CC781400
     *INPUT OPTION CARD',I4,' CODE READ IS ',3A4)                       CC781500
  201 FORMAT ('1')                                                      CC781600
  202 FORMAT (10X,'COMPUTING FRAME TAKEN TO BE THAT FIXED INERTIALLY',//CC781700
     */)                                                                CC781800
  203 FORMAT (10X,'COMPUTING FRAME TAKEN TO BE THAT FIXED IN BODY I',///CC781900
     *)                                                                 CC782000
  204 FORMAT (30X,'TO SPEED UP COMPUTATION VARIOUS TRUNCATED SETS OF BODCC782100
     *Y LABELS HAVE BEEN DEFINED')                                      CC782200
  205 FORMAT (34X,'THE SPECIFICATION OF THESE SETS PERMITS ENGINEERING JCC782300
     *UDGEMENT TO')                                                     CC782400
  206 FORMAT (38X,'BE INTRODUCED INTO THE FORMALISM AND MODELLING ',///)CC782500
  207 FORMAT (30X,'BODY LABELS OF BODIES IN NEST',I3,3X,10I5)           CC782600
  208 FORMAT (20X,10I5,/)                                               CC782700
  209 FORMAT (30X,'PSUEDO INERTIA TENSORS',13X,10I5)                    CC782800
  210 FORMAT (30X,'INERTIA CROSS COUPLING',13X,10I5)                    CC782900
  211 FORMAT (30X,'CENTRIPITAL CROSS COUPLING',9X,10I5)                 CC783000
  212 FORMAT (30X,'CORICLIS CROSS COUPLING',12X,10I5)                   CC783100
  213 FORMAT (10X,'MOMENTUM WHEEL',I3,' IS ASSUMED TO BE VARIABLE SPEED CC783200
     *')                                                                CC783300
  214 FORMAT (10X,'RELATIVE ANGULAR DISPLACEMENT BETWEEN BODIES',I3,' ANCC783400
     *D',I3,' IS COMPUTED VIA INTEGRATION OF DIRECTION COSINE EQUATIONS'CC783500
     *)                                                                 CC783600
  215 FORMAT (////)                                                     CC783700
  216 FORMAT (10X,'THE ELEMENTS OF COLUMN',I3,' DOWN TO THE DIAGONAL IN CC783800
     *THE SYSTEM INERTIA MATRIX OF DYADS ARE ASSUMED TIME VARYING')     CC783900
  217 FORMAT (10X,'VECTORS AND DYADS FIXED IN BODY',I3,' ARE ASSUMED TIMCC784000
     *E VARYING IN THE FRAME OF COMPUTATION ')                          CC784100
  218 FORMAT (10X'DISPLACEMENT ABOUT OR ALONG FREE VECTOR',I3,' COMPUTEDCC784200
     *')                                                                CC784300
  219 FORMAT (10X,'CONSTRAINT TORQUE ABOUT OR ALONG LOCKED VECTOR',I3,' CC784400
     *COMPUTED ')                                                       CC784500
  220 FORMAT (10X,'CONSTRAINT FORCE AT HINGE POINT',I3,' COMPUTED    ') CC784600
  221 FORMAT (10X,'ANGULAR POSITION OF MOMENTUM WHEEL',I3,' COMPUTED ') CC784700
  222 FORMAT (30X,'MOMENTUM WHEEL LABELS IN NEST',I3,3X,10I5)           CC784800
  224 FORMAT (20X,'PRIME CONTRIBUTORS TO COMPUTATION OF ')              CC784900
  225 FORMAT (20X,'FOR THE EQUATION OF MOTION OF NEST',I3,////)         CC785000
  226 FORMAT (10X,'RELATIVE ANGULAR DISPLACEMENT BETWEEN BODIES',I3,' ANCC785100
     *C',I3,' IS COMPUTED VIA SMALL ANGLE ASSUMPTIONS ')                CC785200
  229 FORMAT (10X,'RELATIVE ANGULAR DISPLACEMENT BETWEEN BODIES',I3,' ANCC785300
     *D',I3,' IS COMPUTED VIA EULER ANGLE TECHNIQUES ')                 CC785400
  230 FORMAT (10X,'NOTE: THAC(',I2,') = 0   INITIAL RATE CONDITION RESETCC785500
```

```
     * TC ZERO, SEE SUB INOPT AND MAIN ')                                C0785600
  231 FORMAT (10X,'MOTION ABOUT FREE VECTOR',I3,' UNCAGED AT T=',D15.5) C0785700
  232 FORMAT ('   MODAL DATA OUT OF SEQUENCE ')                          C0785800
  233 FORMAT ('  MODE',I3,2X,' FLOM(',I2,') =',D12.5,5X,' ZETA(',I2,') =C0785900
     *',D12.5)                                                          C0786000
  234 FORMAT ('  FLA(',I2,') =',3D12.5,3X,' (BODY',I2,' FIXED COORDINATEC0786100
     *S) ')                                                             C0786200
  235 FORMAT ('  FLB(',I2,') =',3D12.5,3X,' (BODY',I2,' FIXED COORDINATEC0786300
     *S) ')                                                             C0786400
  236 FORMAT ('  FLC(',I2,') =',3D12.5,3X,' (BODY',I2,' FIXED COORDINATEC0786500
     *S) ')                                                             C0786600
  237 FORMAT ('  FLD(',I2,') =',3D12.5,3X,' (BODY',I2,' FIXED COORDINATEC0786700
     *S) ')                                                             C0786800
  238 FORMAT (11X,3D12.5)                                               C0786900
  239 FORMAT (11X,'  THA(',I2,') =',D12.5,5X,' THAD(',I2,') =',D12.5,/) C0787000
  240 FORMAT ('  FLJ(',I2,') =',3D12.5,3X,' (BODY',I2,' FIXED COORDINATEC0787100
     *S) ')                                                             C0787200
  241 FORMAT (///,10X,'  MODAL DATA FOR BODY',I5,18A4,/)                C0787300
  242 FORMAT (10X,' THE FOLLOWING BODIES HAVE BEEN REDEFINED TO BE FLEXIC0787400
     *BLE ',10I5)                                                       C0787500
  243 FORMAT ('  ')                                                     C0787600
  244 FORMAT (10X,' THE FOLLOWING FLEXIBLE BODIES HAVE SIGNIFICANT MODE C0787700
     *COUPLING IN THEIR DEFORMATION EQUATIONS',10I5)                    C0787800
  246 FORMAT (17X,3D12.5)                                               C0787900
  247 FORMAT ('  FCF(',I2,',',I2,',',I2,') =',3D12.5,' (BODY',I2,' FIXEDC0788000
     * COORDINATES)   LOCATED AT FCF(1,1,',I2,')')                     C0788100
  248 FORMAT ('  FCK(',I2,',',I2,',',I2,') =',3D12.5,' (BODY',I2,' FIXEDC0788200
     * COORDINATES)   LOCATED AT FCK(1,',I2,')')                       C0788300
  249 FORMAT ('  SCXC(',I2,') = 0')                                     C0788400
  250 FORMAT ('  BODY ',I2,' MODE',I3,' CROSS COUPLES IN COORDINATE EQUAC0788500
     *TION ',I2,' COEFFICIENTS AT KF =',I3)                             C0788600
      RETURN                                                            C0788700
      END                                                               C0788800
```

```
C                                                                       CC800000
      SUBROUTINE INTCR                                                  C0800100
C                                                                       CC800200
C                                                                       CC800300
      IMPLICIT REAL*8(A-H,O-Z,$)                                        CC800400
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C0800500
      LOGICAL          RSTART, LRTAPE                                   C0800600
C                                                                       CC800700
C                                                                       CC800800
      INTEGER                                                           C0800900
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , FCON  ,  C0801000
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C0801100
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C0801200
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,  CC801300
     * SQL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,  CC801400
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  CC801500
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  C0801600
     * IINIT(1)      , IZINIT(1)     , SD                              C0801700
C                                                                       CC801800
C                                                                       C0801900
      REAL*8                                                            C0802000
     * ANGC  (3,3)   , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  CC802100
```

```
      * FLQ    (3,20)   , FLE    (3,3,20), FLH    (3,3,20),                    CC802200
      * THADD  (33)     , YMCD   (3,2,11), RINIT  (1)      , RZINIT(1)        CC802300
C                                                                              CC802400
      COMMON /CHEKS/              NSTART, LRTAPE                               CC802500
C                                                                              CC802600
C                                                                              CC802700
C                                                                              CC802800
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INEFF, RBLU(10)                  CC802900
C                                                                              CC803000
C                                                                              CC803100
      COMMON /INTG/     AWORK (200)    ,                                       CC803200
      * CT1             , CT2           , CT3           , CT4           ,      CC803300
      * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,      CC803400
      * MC    (10)      , NBI           , NBOD          , NCTC          ,      CC803500
      * NFER            , NFKC          , NFRC          , NLCR          ,      CC803600
      * NMV             , NMO           , NMCA          , NSVP          ,      CC803700
      * NSVC            , PCON  (11)    , SD            , SFR   (33)    ,      CC803800
      * SG              , SI    (55)    , SIG           , SL            ,      CC803900
      * SLK   (33)      , SMA   (10)    , SCK   (11)    , SQF   (11)    ,      CC804000
      * SQL   (11)      , SMV           , SR            , SSCN          ,      CC804100
      * SSI>            , SVA           , SVB           , SVC           ,      CC804200
      * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,      CC804300
      * SXM   (3,10)    , SXT           , TORQ  (97)    , SMAL          ,      CC804400
      * SEU             , NTQ           , SC    (33)    , SCG           ,      CC804500
      * NFLXB           , SFLX          , SFXM  (10)    , NMODS         ,      CC804600
      * SFCC            , SCC   (10)                                           CC804700
C                                                                              CC804800
C                                                                              CC804900
      COMMON /INTGZ/                                                           CC805000
      * SCNDUM          , SCN   (9)     , SCRDUM        , SCR   (9)     ,      CC805100
      * SFKDUM          , SFK   (9)     , SIXDUM        , SIX   (9)     ,      CC805200
      * SKDUM           , SK    (9)     , SPIDUM        , SPI   (9)     ,      CC805300
      * SMCDUM          , SMC   (9)                                            CC805400
C                                                                              CC805500
C                                                                              CC805600
      COMMON /REAL/                                                            CC805700
      * CA    (3,10)    , CAC   (3,10)  , CLM   (10)    , COMC  (3,11)  ,      CC805800
      * DCMC  (3,11)    , ETC   (3,11)  , ETM   (33)    , FOMC  (3,11)  ,      CC805900
      * GAM   (3,66)    , H             , HM    (3,10)  , HMC   (3,10)  ,      CC806000
      * HMOM  (10)      , PHI   (3,11)  , PLM   (10)    , QF    (3,33)  ,      CC806100
      * QFC   (3,33)    , QL    (3,22)  , QLC   (3,22)  , RCMC  (3,11)  ,      CC806200
      * T               ,                 THA   (33)    , THAD  (33)    ,      CC806300
      * THADW (10)      , THAW  (10)    , XDIC  (3,3,66), XI    (3,3,10),      CC806400
      * XIC   (3,3,10), XMAS  (10)    , XMN   (33,33) , XMT   (3,3,10),      CC806500
      * TLG   (33)      , FLA   (3,20)  , FLB   (3,20)  , FLC   (3,20)  ,      CC806600
      * FLD   (3,3,20), FLJ   (3,3,20), CAO   (3,10)  , XIO   (3,3,10),      CC806700
      * FLIRC (3,10)    , FLCRC (3,10)  , FLAC  (3,20)  , FLQC  (3,20)  ,      CC806800
      * FLCM  (20)      , ZETA  (20)    , FCF   (3,40)  , FCK   (3,3,40),      CC806900
      * TIMEND                                                                 CC807000
C                                                                              CC807100
C                                                                              CC807200
      COMMON /REALZ/                                                           CC807300
      * CBDUM (1,3)     , CB    (3,10)  , CBCDUM(1,3)   , CBC   (3,10)  ,      CC807400
      * XMCDUM(1,1,9) , XMC   (3,3,10), CBN(3)                                CC807500
C                                                                              CC807600
      COMMON /SATELL/ DUMMY(1000)                                              CC807700
C                                                                              CC807800
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))     ,       CC807900
      *           (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))    ,       CC808000
      *           (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))  ,       CC808100
```

```
     *              (FLB(1,1),FLC(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),    CCE08200
     *              (FLH(1,1,1),FLJ(1,1,1))    ,                            C0808300
     *              (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))      ,   CCEC8400
     *              (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,   C0808500
     *              (SCNDUM,IZINIT(1))                                       CCEC8600
C                                                                             C0808700
C                                                                             CC808800
      RETURN                                                                  C0808900
      END                                                                     CCE09000



C                                                                             CC$00000
      SUBROUTINE TRNSIV(XMT,QF,THA,JCON,PCCN,NBOD,RBLO,INERF,XMCDUM,XMC)      C0$00100
C     COMPUTE INITIAL TRANSFORMATION MATRICES                                 C0$00200
C           BODY K TO COMPUTING FRAME COORDINATES                             C0$00300
C                                                                             C0$00400
C     ENTER SUBROUTINE WITH                                                   C0$00500
C          XMT = NOMINAL STATE TRANSFORMATION MATRICES BODY K TO BODY JC      C0$00600
C           QF = FREE COORDINATE VECTOR, EIGENVECTORS                         C0$00700
C          THA = ROTATION ABOUT RESPECTIVE EIGENVECTORS                       C0$00800
C     RETURN FROM SUBROUTINE WITH                                             C0$00900
C          XMC = INITIAL TRANSFORMATION MATRICES BODY K TO COMPUTING FRA      C0$01000
C                                                                             C0$01100
      IMPLICIT REAL*8(A-H,C-Z,$)                                              C0$01200
      LOGICAL    RBLO( 1),INERF, LEQU                                         C0$01300
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,           C0$01400
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,           C0$01500
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ              C0$01600
      INTEGER    JCON( 1),PCCN( 1)                                            C0$01700
      DIMENSION  XMT(3,3, 1),QF(3, 1),THA( 1)                                 C0$01800
      DIMENSION XMCDUM(1,1,9),XMC(3,3, 1)                                     C0$01900
      DIMENSION  XTMP(3,3),XTM1(3,3),XTM2(3,3)                                C0$02000
      DIMENSION  QT1(3),ZT1(4),ZT2(4),ZT3(4),ZT4(4),QT2(3)                    C0$02100
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,           C0$02200
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,           C0$02300
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ              C0$02400
      EQUIVALENCE (LTRNSI,LEQU)                                               C0$02500
C                                                                             C0$02600
C     COMPUTE TRANSFORMATION MATRICES WHICH TAKE VECTORS FROM BODY JCCN(      C0$02700
C          TO BODY K COORDINATES                                              C0$02800
C                                                                             C0$02900
C     NOTE XMT TAKES VECTORS BODY K TO JCCN(K) IN NOMINAL STATE               C0$03000
      M = 1                                                                   C0$03100
      IF(.NOT. LEQU) GO TO 1001                                               C0$03200
      PRINT 200                                                               C0$03300
      PRINT 230                                                               C0$03400
      DO 19  K=1,NBOD                                                         C0$03500
      PRINT 201                                                               C0$03600
      PRINT 202, (XMT(1,J,K),J=1,3)                                           C0$03700
      PRINT 203, K,(XMT(2,J,K),J=1,3)                                         C0$03800
   19 PRINT 202, (XMT(3,J,K),J=1,3)                                           C0$03900
 1001 DO 1  K=1,NBOD                                                          C0$04000
      IF(LEQU) K1 = K-1                                                       C0$04100
C        XMT(I,J,K) - 3X3 TRANSFORMATION MATRIX BODY K TO JCON(K)             C0$04200
C        XTMP(I,J)  - 3X3 TRANSFORMATION MATRIX BODY JCCN(K) TO K             C0$04300
      DO 2  I=1,3                                                             C0$04400
      DO 2  J=1,3                                                             C0$04500
```

```
      2 XTMP(I,J) = XMT(J,I,K)                                          C0S04600
        IF(.NOT. LEQU) GO TO 1002                                       C0S04700
        PRINT 204                                                       C0S04800
        PRINT 231, K1                                                   C0S04900
        PRINT 205, K                                                    C0S05000
C                                                                       C0S05100
C       COMPLT FREE COORDINATE LABEL                                    C0S05200
 1002 IF(K.EQ.1) GO TO 3                                                C0S05300
        M = M + 3 - PCON(K-1)                                           C0S05400
      3 M1 = M + 1                                                      C0S05500
        M2 = M + 2                                                      C0S05600
C                                                                       C0S05700
C       CHECK RIGID BODY OR LINEAR OSCILLATOR                           C0S05800
        IF(RELJ(K).AND.PCON(K).NE.3) GO TO 4                            C0S05900
C       LINEAR OSCILLATOR OR THREE CONSTRAINED AXES, JCON(K) TO K       C0S06000
        DO 5  I=1,3                                                     C0S06100
        DO 5  J=1,3                                                     C0S06200
      5 XMC(I,J,K) = XTMP(I,J)                                          C0S06300
        IF(LEQU) PRINT 206, K                                           C0S06400
        GO TO 1                                                         C0S06500
      4 CONTINUE                                                        C0S06600
C       BODY K IS A RIGID BODY WHICH IS CONNECTED TO BODY JCON(K)        C0S06700
C          BY EITHER A ONE, TWO OR THREE AXIS GIMBAL                    C0S06800
C                                                                       C0S06900
C       PUT FREE VECTOR M IN COORDINATE FRAME I SUB N                   C0S07000
        CALL VECTRN (QF(1,M),XTMP,QT1)                                  C0S07100
        IF(LEQU) PRINT 207, M                                           C0S07200
C       FORM ROTATION QUATERNION EQUATION FRAME I SUB N INTO I SUB 1    C0S07300
        CALL QUATOP(QT1,THA(M),ZT1)                                     C0S07400
        IF(LEQU) PRINT 208,M,(ZT1(I),I=1,4)                             C0S07500
C                                                                       C0S07600
        IF(PCON(K).NE.2) GO TO 6                                        C0S07700
C       BODY K CONNECTED TO BODY JCON(K) BY A ONE AXIS GIMBAL           C0S07800
C          COORDINATE FRAME I SUB 1 IS BODY K FIXED FRAME               C0S07900
C                                                                       C0S08000
        CALL TRANSO(ZT1,XTM1)                                           C0S08100
        IF(LEQU) PRINT 209                                              C0S08200
C       FORM INITIAL TRANSFORMATION MATRIX JCON(K) TO K                 C0S08300
        CALL MATMUL(XTM1,XTMP,XMC(1,1,K),3)                             C0S08400
        IF(LEQU) PRINT 210, K                                           C0S08500
        GO TO 1                                                         C0S08600
C                                                                       C0S08700
      6 IF(PCON(K).NE.1) GO TO 7                                        C0S08800
C       BODY K CONNECTED TO BODY JCON(K) BY A TWO AXIS GIMBAL           C0S08900
C       FREE VECTOR M1 GIVEN IN BODY K COORDINATES, FORM ROTATION QUATERNIC0S09000
        CALL QUATOP(QF(1,M1),THA(M1),ZT2)                               C0S09100
        IF(LEQU) PRINT 211, M1,M1,(ZT2(I),I=1,4)                        C0S09200
C       FORM RESULTANT ROTATION QUATERNION                              C0S09300
        CALL QUTMUL (ZT1,ZT2,ZT4)                                       C0S09400
        IF(LEQU) PRINT 212, (ZT4(I),I=1,4)                              C0S09500
        CALL TRANSO(ZT4,XTM1)                                           C0S09600
        IF(LEQU) PRINT 213                                              C0S09700
C       FORM INITIAL TRANSFORMATION MATRIX JCON(K) TO K                 C0S09800
        CALL MATMUL(XTM1,XTMP,XMC(1,1,K),3)                             C0S09900
        IF(LEQU) PRINT 214, K                                           C0S10000
        GO TO 1                                                         C0S10100
C                                                                       C0S10200
C       BODY K CONNECTED TO BODY JCON(K) BY A THREE AXIS GIMBAL         C0S10300
      7 CALL VECROS (QF(1,M2),QT1,QT2)                                  C0S10400
        CALL VECNRM(QT2)                                                C0S10500
```

```
      IF(LEQU) PRINT 215,M2                                              C0S10600
C     THE COMPONENTS OF FREE VECTOR M1 FOR A THREE AXIS GIMBAL IN THE    C0S10700
C       INTERMEDIATE FRAME I SUB 1 ARE IDENTIAL TO ITS COMPONENTS COMPUT C0S10800
C       WHEN SYSTEM IN NOMINAL STATE                                     C0S10900
      CALL QUATOP(QT2,THA(M1),ZT2)                                       C0S11000
      IF(LEQU) PRINT 216, M1,(ZT2(I),I=1,4)                              C0S11100
      CALL QUATOP(QF(1,M2),THA(M2),ZT3)                                  C0S11200
      IF(LEQU) PRINT 217, M2,M2,(ZT3(I),I=1,4)                           C0S11300
C     FORM RESULTANT QUATERNION BY SUCCESSIVE QUATERNION MULTIPLICATION  C0S11400
      CALL QUTMUL (ZT1,ZT2,ZT4)                                          C0S11500
      IF(LEQU) PRINT 218                                                 C0S11600
      CALL QUTMUL (ZT4,ZT3,ZT1)                                          C0S11700
      IF(LEQU) PRINT 219, (ZT1(I),I=1,4)                                 C0S11800
      CALL TRANSQ(ZT1,XTM1)                                              C0S11900
      IF(LEQU) PRINT 220                                                 C0S12000
C     FORM INITIAL TRANSFORMATION MATRIX JCON(K) TO K                    C0S12100
      CALL MATMUL(XTM1,XTMP,XMC(1,1,K),3)                                C0S12200
      IF(LEQU) PRINT 214, K                                              C0S12300
    1 CONTINUE                                                           C0S12400
      IF(LEQU) PRINT 204                                                 C0S12500
C                                                                        C0S12600
C     ALL CONTIGUOUS BODY TRANSFORMATION MATRICES COMPUTED               C0S12700
C       XMC(I,J,K) = 3X3 TRANSFORMATION MATRIX BODY JCON(K) TO BODY K AT C0S12800
      IF(.NOT. LEQU) GO TO 1000                                          C0S12900
      PRINT 232                                                          C0S13000
      DO 20  K=1,NBOD                                                    C0S13100
      PRINT 201                                                          C0S13200
      PRINT  221,    (XMC(1,J,K),J=1,3)                                  C0S13300
      PRINT 222, K,(XMC(2,J,K),J=1,3)                                    C0S13400
   20 PRINT  221,    (XMC(3,J,K),J=1,3)                                  C0S13500
      PRINT 204                                                          C0S13600
C                                                                        C0S13700
C     COMPUTE TRANSFORMATION MATRICES, COMPUTING FRAME TO BODY K         C0S13800
C                                                                        C0S13900
 1000 IF(INERF) GO TO 8                                                  C0S14000
      IF(LEQU) PRINT 223                                                 C0S14100
      IF(LEQU) PRINT 201                                                 C0S14200
      IC = 1                                                             C0S14300
      GO TO 9                                                            C0S14400
    8 IC = 0                                                             C0S14500
      IF(LEQU) PRINT 224                                                 C0S14600
      IF(LEQU) PRINT 201                                                 C0S14700
    9 ICB1 = IC + 1                                                      C0S14800
      DO 10 KKK=ICB1,NBOD                                                C0S14900
      K = NBOD - (KKK-ICB1)                                              C0S15000
      IF(LEQU) PRINT 201                                                 C0S15100
      KK = K                                                             C0S15200
   12 JK = JCON(KK)                                                      C0S15300
      IF(JK.EQ.IC) GO TO 10                                              C0S15400
      DO 11 I=1,3                                                        C0S15500
      DO 11 J=1,3                                                        C0S15600
      XTM1(I,J) = XMC(I,J,K)                                             C0S15700
   11 XTM2(I,J) = XMC(I,J,JK)                                            C0S15800
      CALL MATMUL(XTM1,XTM2,XMC(1,1,K),3)                                C0S15900
      IF(LEQU) PRINT 225, K,K,JK                                         C0S16000
      KK = JK                                                            C0S16100
      GO TO 12                                                           C0S16200
   10 CONTINUE                                                           C0S16300
      IF(LEQU) PRINT 204                                                 C0S16400
C                                                                        C0S16500
```

```
C     GET COMPUTING FRAME TO BODY 1 AND TO INERTIAL REFERENCE                  CCS16600
      IF(IC.EQ.1) GO TO 14                                                    CCS16700
      DO 15  I=1,3                                                            CCS16800
      DO 15  J=1,3                                                            CCS16900
   15 XMC(I,J,C) = 0                                                          CCS17000
      XMC(1,1,C) = 1                                                          CCS17100
      XMC(2,2,C) = 1                                                          CCS17200
      XMC(3,3,C) = 1                                                          CCS17300
      IF(LEQU) PRINT 226                                                      CCS17400
      GO TO 16                                                                CCS17500
   14 DO 17  I=1,3                                                            CCS17600
      DO 17  J=1,3                                                            CCS17700
   17 XMC(J,I,C) = XMC(I,J,1)                                                 CCS17800
      IF(LEQU) PRINT 227                                                      CCS17900
      DO 18  I=1,3                                                            CCS18000
      DO 18  J=1,3                                                            CCS18100
   18 XMC(I,J,1) = 0                                                          CCS18200
      XMC(1,1,1) = 1                                                          CCS18300
      XMC(2,2,1) = 1                                                          CCS18400
      XMC(3,3,1) = 1                                                          CCS18500
      IF(LEQU) PRINT 228                                                      CCS18600
   16 CONTINUE                                                                CCS18700
C                                                                             CCS18800
C     TRANSPOSE TO GET TRANSFORMATION MATRICES BODY K TO COMPUTING FRAME      CCS18900
      NBOD1=NBOD + 1                                                          CCS19000
      DO 13 KKK=1,NBOD1                                                       CCS19100
      K=KKK - 1                                                               CCS19200
      CALL TRNSPS  (XMC(1,1,K))                                               CCS19300
      IF(LEQU) PRINT 229, K,K                                                 CCS19400
   13 CONTINUE                                                                CCS19500
      IF(.NOT.LEQU) RETURN                                                    CCS19600
      PRINT 204                                                               CCS19700
      PRINT 233                                                               CCS19800
      DO 21 KKK=1,NBOD1                                                       CCS19900
      K = KKK-1                                                               CCS20000
      PRINT 201                                                               CCS20100
      PRINT 221,   (XMC(1,J,K),J=1,3)                                         CCS20200
      PRINT 222, K,(XMC(2,J,K),J=1,3)                                         CCS20300
   21 PRINT 221,   (XMC(3,J,K),J=1,3)                                         CCS20400
      PRINT 204                                                               CCS20500
  200 FORMAT ('1  SUBROUTINE TRNSIV  ENTERED ',//)                            CCS20600
  201 FORMAT ('     ')                                                        CCS20700
  202 FORMAT (12X,3D15.5)                                                     CCS20800
  203 FORMAT ('  XMT(',I2,') = ',3D15.5)                                      CCS20900
  204 FORMAT (3(/))                                                           CCS21000
  205 FORMAT ('  XTMP = XMT(',I2,')**T ')                                     CCS21100
  206 FORMAT ('  XMC(',I2,') = XTMP ')                                        CCS21200
  207 FORMAT ('  QT1 = XTMP * QF(',I2,') ')                                   CCS21300
  208 FORMAT ('  ZT1 = QUATOF(QT1,THA(',I2,'))',8X,'= ',4D15.5)               CCS21400
  209 FORMAT ('  XTM1 = TRANSQ(ZT1) ')                                        CCS21500
  210 FORMAT ('  XMC(',I2,') = XTM1 * XTMP  ')                                CCS21600
  211 FORMAT ('  ZT2 = QUATOF(QF(',I2,'),THA(',I2,'))',5X,'= ',4D15.5)        CCS21700
  212 FORMAT ('  ZT4 = ZT1 * ZT2   ',15X,'= ',4D15.5)                         CCS21800
  213 FORMAT ('  XTM1 = TRANSQ(ZT4) ')                                        CCS21900
  214 FORMAT ('  XMC(',I2,') = XTM1 * XTMP ')                                 CCS22000
  215 FORMAT ('  QT2 = NORM(QF(',I2,') X QT1)')                               CCS22100
  216 FORMAT ('  ZT2 = QUATOF(QT2,THA(',I2,'))',8X,'= ',4D15.5)               CCS22200
  217 FORMAT ('  ZT3 = QUATOF(QF(',I2,'),THA(',I2,'))',5X,'= ',4D15.5)        CCS22300
  218 FORMAT ('  ZT4 = ZT1 * ZT2  ')                                          CCS22400
  219 FORMAT ('  ZT1 = ZT4 * ZT3   ',15X,'= ',4D15.5)                         CCS22500
```

```
 220 FORMAT (' XTM1 = TRANSC(ZT1).')                                    C0S22600
 221 FORMAT (12X,3D15.5)                                                C0S22700
 222 FORMAT (' XMC(',I2,') = ',3D15.5)                                  C0S22800
 223 FORMAT (' COMPUTING FRAME FIXED IN BODY I ')                       C0S22900
 224 FORMAT (' COMPUTING FRAME FIXED INERTIALLY')                       C0S23000
 225 FORMAT (' XMC(',I2,') = XMC(',I2,') * XMC(',I2,') ')               C0S23100
 226 FORMAT (' XMC( 0) = 1 ')                                           C0S23200
 227 FORMAT (' XMC( 0) = XMC( 1)**T ')                                  C0S23300
 228 FORMAT (' XMC( 1) = 1 ')                                           C0S23400
 229 FORMAT (' XMC(',I2,') = XMC(',I2,')**T ')                          C0S23500
 230 FORMAT (10X,' TRANSFORMATION MATRICES, NOMINAL STATE BODY K TO BOD C0S23600
    *Y JCON(K) ',//)                                                    C0S23700
 231 FORMAT (10X,' HINGE POINT ',I2,/)                                  C0S23800
 232 FORMAT (10X,' TRANSFORMATION MATRICES, TIME ZERO BODY JCON(K) TO B C0S23900
    *ODY K ')                                                           C0S24000
 233 FORMAT (10X,' TRANSFORMATION MATRICES, TIME ZERO BODY K TO COMPUTI C0S24100
    *NG FRAME ',/)                                                      C0S24200
     RETURN                                                             C0S24300
     END                                                                C0S24400
```

```
C                                                                       C1C00000
      SUBROUTINE VDIV                                                   C1C00100
C     USED TO TRANSFORM ALL VECTORS AND DYADS TO COMPUTING FRAME        C1C00200
C       DEFINES FREE AND LOCKED VECTORS NOT INPUTED                     C1C00300
C       SETS UP DO LOOP SETS FOR TRANVD                                 C1C00400
C                                                                       C1C00500
C                                                                       C1C00600
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1C00700
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, KBLO, LEQU, LINIT(1)     C1C00800
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1C00900
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1C01000
     *                 LOFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1C01100
C                                                                       C1C01200
C                                                                       C1C01300
      INTEGER                                                           C1C01400
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCCN   , PCCN   ,  C1C01500
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,  C1C01600
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,  C1C01700
     * SMA    , SMCDUM, SMC    , SMV    , SOK    , SPIDUM, SPI    , SQF    ,  C1C01800
     * SQL    , SR     , SSCN   , SSIX   , SVA    , SVB    , SVD    , SVI    ,  C1C01900
     * SVM    , SVP    , SVQ    , SXM    , SXT    , TORQ   , SMAL   , SEU    ,  C1C02000
     * SC     , SCG    , NFLXB  , SFLX   , SFXM   , NMODS  , SFCC   , SCC    ,  C1C02100
     * IINIT(1)        , IZINIT(1)      , SD                            C1C02200
C                                                                       C1C02300
C                                                                       C1C02400
C                                                                       C1C02500
      REAL*8                                                            C1C02600
     * ANGC  (33)     , CNF   (3,10)   , ETIC  (3,10)  , ETMC  (3,10)  ,  C1C02700
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                 C1C02800
     * TFACD (33)     , YMCD  (3,2,11), RINIT (1)      , RZINIT(1)     C1C02900
C                                                                       C1C03000
C                                                                       C1C03100
C                                                                       C1C03200
C                                                                       C1C03300
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1C03400
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1C03500
```

```
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ                C1C03600
C                                                                              C1C03700
C                                                                              C1C03800
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)                  C1C03900
C                                                                              C1C04000
C                                                                              C1C04100
      COMMON /INTG/    AWORK (200)   ,                                         C1C04200
     * CT1             , CT2           , CT3           , CT4           ,       C1C04300
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,       C1C04400
     * MC    (10)      , NB1           , NBOD          , NCTC          ,       C1C04500
     * NFER            , NFKC          , NFRC          , NLOR          ,       C1C04600
     * NMV             , NMO           , NMOA          , NSVP          ,       C1C04700
     * NSVC            , PCON  (11)    , SD            , SFR   (33)    ,       C1C04800
     * SG              , SI    (55)    , SIG           , SL            ,       C1C04900
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,       C1C05000
     * SQL   (11)      , SMV           , SR            , SSCN          ,       C1C05100
     * SSIX            , SVA           , SVB           , SVD           ,       C1C05200
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,       C1C05300
     * SXM   (3,10)    , SXT           , TURQ  (97)    , SMAL          ,       C1C05400
     * SEU             , NTQ           , SC    (33)    , SCG           ,       C1C05500
     * NFLXB           , SFLX          , SFXM  (10)    , NMODS         ,       C1C05600
     * SFCC            , SCC   (10)                                           C1C05700
C                                                                              C1C05800
C                                                                              C1C05900
      COMMON /INTGZ/                                                           C1C06000
     * SCNDUM          , SCN   (9)     , SCRDUM        , SCR   (9)     ,       C1C06100
     * SFKDUM          , SFK   (9)     , SIXDUM        , SIX   (9)     ,       C1C06200
     * SKDUM           , SK    (9)     , SPIDUM        , SPI   (9)     ,       C1C06300
     * SMCDUM          , SMC   (9)                                            C1C06400
C                                                                              C1C06500
C                                                                              C1C06600
      COMMON /REAL/                                                            C1C06700
     * CA    (3,10)    , CAC   (3,10)  , CLM   (10)    , CCMC  (3,11)  ,       C1C06800
     * DCMC  (3,11)    , ETC   (3,11)  , ETM   (33)    , FCMC  (3,11)  ,       C1C06900
     * GAM   (3,66)    , H             , HM    (3,10)  , HMC   (3,10)  ,       C1C07000
     * HMOM  (10)      , PHI   (3,11)  , PLM   (10)    , QF    (3,33)  ,       C1C07100
     * QFC   (3,33)    , QL    (3,22)  , QLC   (3,22)  , RCMC  (3,11)  ,       C1C07200
     * T               ,                 THA   (33)    , THAD  (33)    ,       C1C07300
     * THACW (10)      , THAW  (10)    , XDIC  (3,3,66), XI    (3,3,10),       C1C07400
     * XIC   (3,3,10)  , XMAS  (10)    , XMN   (33,33) , XMT   (3,3,10),       C1C07500
     * TLG   (33)      , FLA   (3,20)  , FLB   (3,20)  , FLC   (3,20)  ,       C1C07600
     * FLD   (3,3,20)  , FLJ   (3,3,20), CAO   (3,10)  , XIO   (3,3,10),       C1C07700
     * FLIRC (3,10)    , FLCRC (3,10)  , FLAC  (3,20)  , FLGC  (3,20)  ,       C1C07800
     * FLOM  (20)      , ZETA  (20)    , FCF   (3,3,40), FCK   (3,40)  ,       C1C07900
     * TIMEND                                                                  C1C08000
C                                                                              C1C08100
C                                                                              C1C08200
      COMMON /REALZ/                                                           C1C08300
     * CBDUM (1,3)     , CE    (3,10)  , CBCDUM(1,3)   , CBC   (3,10)  ,       C1C08400
     * XMCDUM(1,1,9)   , XMC   (3,3,10), CBN(3)                               C1C08500
C                                                                              C1C08600
C                                                                              C1C08700
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))       ,       C1C08800
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNE(1,1))      ,       C1C08900
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))    ,       C1C09000
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1))),      C1C09100
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                               C1C09200
     *            (FG1,LINIT(1))            ,(CA(1,1),RINIT(1))       ,       C1C09300
     *            (CBDUM(1,1),RZINIT(1))    ,(AWORK(1),IINIT(1))      ,       C1C09400
     *            (SCNDUM,IZINIT(1))                                          C1C09500
```

```
C                                                                       C1C09600
      LOGICAL CTAIN                                                     C1C09700
      INTEGER SET(18),S(18),SS(18)                                      C1C09800
      INTEGER ST1(10),ST2(10),ST3(10),ST4(10),SFXMN                     C1C09900
      REAL*8 TEM(3), TEM1(3,3), TEM2(3,3), TEM3(3,3)                    C1C10000
      EQUIVALENCE (LVDIV,LEQU)                                          C1C10100
C                                                                       C1C10200
C                                                                       C1C10300
C                                                                       C1C10400
      IF(LEQU)PRINT 200                                                 C1C10500
C                                                                       C1C10600
      IF(INERF) GO TO 1                                                 C1C10700
      IC = 1                                                            C1C10800
      IF(LEQU)PRINT 201                                                 C1C10900
      GO TO 7                                                           C1C11000
    1 IC = 0                                                            C1C11100
      IF(LEQU)PRINT 202                                                 C1C11200
    7 CONTINUE                                                          C1C11300
C                                                                       C1C11400
C     INCLUDE FLEXIBILITY EFFECTS                                       C1C11500
      IF(NFLXB.EQ.0) GO TO 75                                           C1C11600
      CALL UNPAC(SET,NSET,SFLX)                                         C1C11700
      IF(LEQU) PRINT 231, SFLX,(SET(I),I=1,NSET)                        C1C11800
C     SAVE UNDEFORMED CM VECTOR AND INERTIA TENSOR DATA                 C1C11900
      MN = 0                                                            C1C12000
      DO 76 NN=1,NSET                                                   C1C12100
      N = SET(NSET+1-NN)                                                C1C12200
      IF(LEQU) PRINT 257, N                                             C1C12300
      DO 77 I=1,3                                                       C1C12400
      CAC(I,N) = CA(I,N)                                                C1C12500
      DO 77 J=1,3                                                       C1C12600
   77 XIC(I,J,N) = XI(I,J,N)                                            C1C12700
      IF(.NOT.LEQU) GO TO 1001                                          C1C12800
      PRINT 232, N,N                                                    C1C12900
      PRINT 234, N,N                                                    C1C13000
      PRINT 233                                                         C1C13100
 1001 CONTINUE                                                          C1C13200
      SFXMN = SFXM(N)                                                   C1C13300
      DO 78 M=1,SFXMN                                                   C1C13400
      MN = MN+1                                                         C1C13500
      CALL VECTRN(FLA(1,MN),XMC(1,1,N),FLAC(1,MN))                      C1C13600
      IF(.NOT.LEQU) GO TO 1002                                          C1C13700
      PRINT 238, N,MN                                                   C1C13800
      PRINT 248, MN,N,MN,(FLAC(I,MN),I=1,3)                             C1C13900
      PRINT 233                                                         C1C14000
 1002 CONTINUE                                                          C1C14100
C     USE TEMPORARY LOCATIONS SO THAT                                   C1C14200
C      EQUIVALENCE MAY BE USED TO SAVE STORAGE                          C1C14300
      DO 79 I=1,3                                                       C1C14400
      DO 79 J=1,3                                                       C1C14500
      TEM1(I,J) = FLD(I,J,MN)                                           C1C14600
      TEM2(I,J) = FLD(J,I,MN)                                           C1C14700
   79 TEM3(I,J) = FLJ(I,J,MN)                                           C1C14800
      CALL DYADD(TEM1,TEM2,FLE(1,1,MN))                                 C1C14900
      CALL DYADD(TEM1,TEM3,FLH(1,1,MN))                                 C1C15000
      IF(.NOT.LEQU) GO TO 1003                                          C1C15100
      PRINT 244,          (FLE(1,I,MN),I=1,3)                           C1C15200
      PRINT 242,MN,MN,MN,(FLE(2,I,MN),I=1,3)                           C1C15300
      PRINT 244,          (FLE(3,I,MN),I=1,3)                           C1C15400
      PRINT 233                                                         C1C15500
```

```
      PRINT 244,          (FLB(1,I,MN),I=1,3)                          C1C15600
      PRINT 243,MN,MN,MN,(FLB(2,I,MN),I=1,3)                          C1C15700
      PRINT 244,          (FLB(3,I,MN),I=1,3)                          C1C15800
      PRINT 233                                                        C1C15900
 1003 CONTINUE                                                         C1C16000
C                                                                      C1C16100
C     COMPUTE G VECTOR IN BODY N FIXED FRAME                           C1C16200
      CALL VECROS(CAD(1,N),FLA(1,MN),TEM)                              C1C16300
      CALL VECSUB(FLB(1,MN),TEM,TEM)                                   C1C16400
      CALL SCLV(XMAS(N),TEM,TEM)                                       C1C16500
      CALL VECADD(FLC(1,MN),TEM,FLQ(1,MN))                             C1C16600
      CALL VECTRN(FLQ(1,MN),XMC(1,1,N),FLQC(1,MN))                     C1C16700
      IF(.NOT.LEQU) GO TO 1004                                         C1C16800
      PRINT 245, MN,MN,N,MN,N,MN,(FLQ(I,MN),I=1,3)                    C1C16900
      PRINT 233                                                        C1C17000
      PRINT 249, MN,N,MN,(FLQC(I,MN),I=1,3)                            C1C17100
      PRINT 233                                                        C1C17200
 1004 CONTINUE                                                         C1C17300
      CALL SCLV(THA(NFER+MN),FLA(1,MN),TEM)                            C1C17400
      CALL VECADD(CA(1,N),TEM,CA(1,N))                                 C1C17500
      CALL SCLD(THA(NFER+MN),FLE(1,1,MN),TEM1)                         C1C17600
      CALL DYADD(XI(1,1,N),TEM1,XI(1,1,N))                             C1C17700
      IF(.NOT.LEQU) GO TO 1006                                         C1C17800
      NFCM = NFER+MN                                                   C1C17900
      PRINT 246, N,N,NFCM,MN                                           C1C18000
      PRINT 247, N,N,NFCM,MN                                           C1C18100
      PRINT 233                                                        C1C18200
 1006 CONTINUE                                                         C1C18300
   78 CONTINUE                                                         C1C18400
      IF(.NOT.LEQU) GO TO 1007                                         C1C18500
      PRINT 251, N                                                     C1C18600
      PRINT 252, N,(CA(I,N),I=1,3)                                     C1C18700
      PRINT 233                                                        C1C18800
      PRINT 254,    (XI(1,I,N),I=1,3)                                  C1C18900
      PRINT 255, N,(XI(2,I,N),I=1,3)                                   C1C19000
      PRINT 254,    (XI(3,I,N),I=1,3)                                  C1C19100
      PRINT 233                                                        C1C19200
 1007 CONTINUE                                                         C1C19300
   76 CONTINUE                                                         C1C19400
   75 CONTINUE                                                         C1C19500
C                                                                      C1C19600
C                                                                      C1C19700
C        CENTER OF MASS VECTORS                                        C1C19800
      IF(LEQU)PRINT 204                                                C1C19900
      DO 2  J=1,NBOD                                                   C1C20000
    2 CALL VECTRN (CA(1,J),XMC(1,1,J),CAC(1,J))                        C1C20100
      IF(.NOT. LEQU) GO TO 1020                                        C1C20200
      DO 8  J=1,NBOD                                                   C1C20300
    8 PRINT 203, J,J,J,(CAC(I,J),I=1,3)                                C1C20400
C                                                                      C1C20500
C        INERTIA TENSORS                                               C1C20600
      PRINT 205                                                        C1C20700
 1020 CALL UNPAC(SET,NSET,SR)                                          C1C20800
      DO 46  J=1,NSET                                                  C1020900
      K = SET(J)                                                       C1C21000
   46 CALL TENTRN (XI(1,1,K),XMC(1,1,K),XIC(1,1,K))                    C1C21100
      IF(.NOT. LEQU) GO TO 1000                                        C1021200
      DO 49  J=1,NSET                                                  C1021300
      PRINT 233                                                        C1C21400
      K=SET(J)                                                         C1C21500
```

```
      PRINT 206,          (XIC(1,L,K),L=1,3)                                C1C21600
      PRINT 207, K,K,K,K,(XIC(2,L,K),L=1,3)                                C1021700
   45 PRINT 206,          (XIC(3,L,K),L=1,3)                                C1C21800
C                                                                          C1C21900
 1000 CONTINUE                                                             C1C22000
C                                                                          C1022100
C     GET ELEMENTS OF SVA; THAT IS, SVD MINUS ZERO CA VECTORS              C1C22200
C     IF BODY FLEXIBLE BOTH CM VECTOR AND INERTIA TENSOR MUST BE           C1C22300
C      TRANSFORMED EVERY STEP                                              C1C22400
      CALL UNPAC(SET,NSET,SVD)                                             C1C22500
      CALL UNPAC(ST1,NS1,SFLX)                                             C1C22600
      IF(LEQU)PRINT 208, SVD,(SET(I),I=1,NSET)                             C1C22700
      NS = 0                                                               C1C22800
      DO 51 I=1,18                                                         C1C22900
   51 S(I) = 0                                                             C1C23000
      IF(NSET.EQ.0) GO TO 5024                                             C1C23100
      DO 50  J=1,NSET                                                      C1C23200
      K = SET(J)                                                           C1C23300
      IF(CTAIN(K,ST1,NS1)) GO TO 50                                        C1C23400
      IF(CAC(1,K).EQ.0.AND.CAC(2,K).EQ.0.AND.CAC(3,K).EQ.0) GO TO 50       C1C23500
   80 CONTINUE                                                             C1C23600
      NS = NS+1                                                            C1C23700
      S(NS) = K                                                            C1C23800
   50 CONTINUE                                                             C1C23900
 5024 CONTINUE                                                             C1C24000
      CALL COMPAC(S,NS,SVA)                                                C1C24100
      IF(LEQU)PRINT 241, SVA,(S(I),I=1,NS)                                 C1C24200
      DO 47 I=1,18                                                         C1C24300
   47 S(I) = 0                                                             C1C24400
      CALL UNPAC(SS,NS,SVD)                                                C1C24500
      CALL UNPAC(SET,NSET,SR)                                              C1C24600
      K = 0                                                                C1C24700
      DO 48  J=1,NBOD                                                      C1C24800
      IF(.NOT.(CTAIN(J,SET,NSET).AND.CTAIN(J,SS,NS))) GO TO 48             C1C24900
   81 CONTINUE                                                             C1025000
      K = K+1                                                              C1025100
      S(K) = J                                                             C1025200
   48 CONTINUE                                                             C1C25300
      CALL COMPAC(S,K,SVI)                                                 C1025400
      IF(LEQU)PRINT 209, SVI,(S(I),I=1,K)                                  C1025500
C     DO ON SVI ELEMENTS IN TRANVC TO GET XIC                              C1C25600
C                                                                          C1C25700
C       HINGE VECTORS                                                      C1C25800
      IF(LEQU)PRINT 210                                                    C1C25900
      DO 4 J=1,18                                                          C1026000
    4 S(J) = 0                                                             01026100
      CALL UNPAC(SET,NSET,SVD)                                             C1026200
      K = 0                                                                C1C26300
      DO 5  J=1,NBOD                                                       C1C26400
      JJ = JCEN(J)                                                         C1C26500
      CALL VECTRN (CB(1,J),XMC(1,1,JJ),CBC(1,J))                           C1C26600
      IF(LEQU)PRINT 211, J,JJ,J,(CBC(I,J),I=1,3)                           C1C26700
      IF(CEC(1,J).EQ.0.AND.CBC(2,J).EQ.0.AND.CBC(3,J).EQ.0) GO TO 5        C1C26800
      IF(.NOT.CTAIN(JJ,SET,NSET)) GO TO 5                                  C1C26900
      K = K+1                                                              C1C27000
      S(K) = J                                                             C1027100
    5 CONTINUE                                                             C1C27200
C     CB(I) ALONG WITH CBC(I) MUST BE COMPUTED AT EACH INTEGRATION STEP.   C1C27300
C      PUT I IN SVB                                                        C1C27400
      K = K+1                                                              C1C27500
```

```
      S(K) = 1                                                          C1C27600
    6 CALL COMPAC(S,K,SVE)                                              C1C27700
      IF(LEQU)PRINT 212, SVB,(S(I),I=1,K)                               C1C27800
C      DO CN SVB ELEMENTS IN TRANVC                                     C1C27900
C                                                                       C1C28000
C      NOTE CBC(I,0) IS COMPOSITE CENTER OF MASS, MUST BE CALCULATED    C1C28100
C                                                                       C1C28200
C        FREE AND LOCKED COORDINATE VECTORS                             C1C28300
      IF(LEQU)PRINT 213                                                 C1C28400
      M = 1                                                             C1C28500
      L = 1                                                             C1C28600
      DO 9  K=1,NBOD                                                    C1C28700
      IF(K.EQ.1) GO TO 10                                               C1C28800
      M = M+3-FCON(K-1)                                                 C1C28900
      L = L+PCON(K-1)                                                   C1C29000
   10 M1 = M+1                                                          C1C29100
      M2 = M+2                                                          C1C29200
      L1 = L+1                                                          C1C29300
      L2 = L+2                                                          C1C29400
      IGOTO = FCON(K) + 1                                               C1C29500
      GO TO(11,12,13,14),IGOTO                                          C1C29600
C                                                                       C1C29700
C        THREE DEGREES OF FREEDOM                                       C1C29800
   11 CALL VECTRN (QF(1,M),XMC(1,1,FCON(M)),QFC(1,M))                   C1C29900
      CALL VECTRN (QF(1,M2),XMC(1,1,FCON(M2)),QFC(1,M2))                C1C30000
      IF(LEQU)PRINT 214, M,FCON(M),M                                    C1C30100
      IF(LEQU)PRINT 214, M2,FCON(M2),M2                                 C1C30200
      IF(FCON(M1).LT.0) GO TO 15                                        C1C30300
      CALL VECROS (QF(1,M2),QF(1,M),QF(1,M1))                           C1C30400
      IF(LEQU)PRINT 236, M1,M2,M                                        C1C30500
      CALL VECTRN (QF(1,M1),XMC(1,1,FCON(M1)),QFC(1,M1))                C1C30600
      IF(LEQU)PRINT 214, M1,FCON(M1),M1                                 C1C30700
      GO TO 9                                                           C1C30800
   15 CALL VECROS (QFC(1,M2),QFC(1,M ),QFC(1,M1))                       C1C30900
      CALL VECNRM (QFC(1,M1))                                           C1C31000
      DO 45  I=1,3                                                      C1C31100
   45 QF(I,M1) = 0                                                      C1C31200
      IF(LEQU)PRINT 216, M1, M2, M                                      C1C31300
      GO TO 9                                                           C1C31400
C                                                                       C1C31500
C        TWO DEGREES OF FREEDOM                                         C1C31600
   12 CALL VECTRN (QF(1,M),XMC(1,1,FCON(M)),QFC(1,M))                   C1C31700
      CALL VECTRN (QF(1,M1),XMC(1,1,FCON(M1)),QFC(1,M1))                C1C31800
      IF(LEQU)PRINT 214, M, FCON(M), M                                  C1C31900
      IF(LEQU)PRINT 214, M1, FCON(M1), M1                               C1C32000
      IF(LCON(L).LT.0) GO TO 16                                         C1C32100
      CALL VECROS (QF(1,M),QF(1,M1),QL(1,L))                            C1C32200
      IF(LEQU)PRINT 237, L,M,M1                                         C1C32300
      CALL VECTRN (QL(1,L),XMC(1,1,LCON(L)),QLC(1,L))                   C1C32400
      IF(LEQU)PRINT 217, L, LCON(L),  L                                 C1C32500
      GO TO 9                                                           C1C32600
   16 CALL VECROS (QFC(1,M),QFC(1,M1),QLC(1,L))                         C1C32700
      CALL VECNRM (QLC(1,L))                                            C1C32800
      DO 25  I=1,3                                                      C1C32900
   25 QL(I,L) = 0                                                       C1C33000
      IF(LEQU)PRINT 235, L,M,M1                                         C1C33100
      GO TO 9                                                           C1C33200
C                                                                       C1C33300
C        ONE DEGREE OF FREEDOM                                          C1C33400
   13 CALL VECTRN (QF(1,M),XMC(1,1,FCON(M)),QFC(1,M))                   C1C33500
```

```
      CALL VECTRN (QL(1,L),XMC(1,1,LCON(L)),QLC(1,L))                   C1C33600
      IF(LEQU)PRINT 214, M,FCON(M),M                                    C1C33700
      IF(LEQU)PRINT 217, L,LCON(L),L                                    C1C33800
      IF(RBLO(K)) GO TO 17                                              01C33900
      CALL VECROS (QF(1,M),QL(1,L),QL(1,L1))                            01C34000
      IF(LEQU)PRINT 238, L1,M,L                                         C1C34100
      CALL VECTRN (QL(1,L1),XMC(1,1,LCON(L1)),QLC(1,L1))                C1C34200
      IF(LEQU)PRINT 217, L1,LCON(L1),L1                                 C1C34300
      GO TO 9                                                           C1C34400
   17 CALL TRNSPS  (XMT(1,1,K))                                         C1C34500
      CALL VECTRN (QF(1,M),XMT(1,1,K),TEM)                              01C34600
      CALL TRNSPS  (XMT(1,1,K))                                         C1C34700
      CALL VECROS (TEM,QL(1,L),QL(1,L1))                                C1C34800
      IF(LEQU)PRINT 239, L1,K,M,L                                       C1C34900
      CALL VECTRN (QL(1,L1),XMC(1,1,LCON(L1)),QLC(1,L1))                C1C35000
      IF(LEQU)PRINT 217, L1,LCON(L1),L1                                 C1C35100
      GO TO 9                                                           C1C35200
C                                                                       C1C35300
C     FCON(K) = 3 ZERO DEGREES OF FREEDOM                               C1C35400
   14 CALL VECROS (QL(1,L),QL(1,L1),QL(1,L2))                           C1C35500
      IF(LEQU)PRINT 217, L,LCON(L),L                                    C1C35600
      IF(LEQU)PRINT 217, L1,LCON(L1),L1                                 C1C35700
      IF(LEQU)PRINT 240, L2,L,L1                                        C1C35800
      IF(LEQU)PRINT 217, L2,LCON(L2),L2                                 C1C35900
      DO 18 II=1,3                                                      C1C36000
      I=II-1                                                            C1C36100
   18 CALL VECTRN (QL(1,L+I),XMC(1,1,LCON(L+I)),QLC(1,L+I))             C1C36200
C                                                                       C1C36300
    9 CONTINUE                                                          C1C36400
C                                                                       01C36500
C        AT INERTIAL ORIGIN                                             C1C36600
      M = M+3-FCON(NBOD)                                                C1C36700
      L = L+FCON(NBOD)                                                  C1C36800
      DO 19  I=1,3                                                      01C36900
      MB2 = M+2                                                         C1C37000
      DO 20 J=M,MB2                                                     01C37100
   20 QF(I,J) = 0                                                       C1C37200
      LB2 = L+2                                                         C1C37300
      DO 21 J=L,LB2                                                     C1C37400
   21 QL(I,J) = 0                                                       C1C37500
   19 CONTINUE                                                          C1C37600
      IF(LEQU)M0 = 0                                                    C1C37700
C                                                                       C1C37800
      IGOTO = FCON(NBOD+1)+1                                            C1C37900
      GO TO (22,23,24,25),IGOTO                                         C1C38000
C                                                                       C1C38100
C       THREE DEGREES OF FREEDOM                                        C1C38200
   22 M1 = M+1                                                          01C38300
      M2 = M+2                                                          C1C38400
      FCON(M) = 0                                                       C1C38500
      FCON(M1) = 0                                                      C1C38600
      FCON(M2) = 0                                                      C1C38700
      QF(1,M) = 1                                                       C1C38800
      QF(2,M1) = 1                                                      C1C38900
      QF(3,M2) = 1                                                      C1C39000
      IF(IC.EQ.0) GO TO 26                                              C1C39100
      DO 27 II=1,3                                                      C1C39200
      I = II-1                                                          C1C39300
      IF(LEQU)JM = M+I                                                  C1C39400
      IF(LEQU)PRINT 214, JM,M0,JM                                       01C39500
```

```
   27 CALL VECTRN (QF(1,M+I),XMC(1,1,0),QFC(1,M+I))                    C1C39600
      GO TO 26                                                         01C39700
C                                                                      C1C39800
C        TWO DEGREES OF FREEDOM                                        C1C39900
   23 M1 = M+1                                                         C1C40000
      FCON(M) = 0                                                      C1C40100
      FCON(M1) = 0                                                     01040200
      LCON(L) = 0                                                      C1C40300
      QF(1,M) = 1                                                      C1C40400
      QF(2,M1) = 1                                                     C1C40500
      QL(3,L) = 1                                                      C1C40600
      IF(IC.EQ.0) GO TO 26                                             C1C40700
      IF(LEQU)PRINT 214, M,MC,M                                        C1C40800
      IF(LEQU)PRINT 214, M1,M0,M1                                      C1C40900
      IF(LEQU)PRINT 217, L,M0,L                                        C1C41000
      CALL VECTRN (QF(1,M),XMC(1,1,0),QFC(1,M))                        C1C41100
      CALL VECTRN (QF(1,M1),XMC(1,1,0),QFC(1,M1))                      C1041200
      CALL VECTRN (QL(1,L),XMC(1,1,0),QLC(1,L))                        C1C41300
      GO TO 26                                                         C1C41400
C                                                                      C1C41500
C        ONE DEGREE  OF FREEDOM                                        C1C41600
   24 L1 = L+1                                                         C1C41700
      FCON(M) = 0                                                      C1C41800
      LCON(L) = 0                                                      C1C41900
      LCON(L1) = 0                                                     C1C42000
      QF(1,M) = 1                                                      C1042100
      QL(2,L) = 1                                                      C1C42200
      QL(3,L1) = 1                                                     C1042300
      IF(IC.EQ.0) GO TO 26                                             C1C42400
      IF(LEQU)PRINT 214, M,MC,M                                        C1C42500
      IF(LEQU)PRINT 217, L,MC,L                                        C1C42600
      IF(LEQU)PRINT 217, L1,M0,L1                                      01C42700
      CALL VECTRN (QF(1,M),XMC(1,1,0),QFC(1,M))                        01C42800
      CALL VECTRN (QL(1,L),XMC(1,1,0),QLC(1,L))                        C1C42900
      CALL VECTRN (QL(1,L1),XMC(1,1,0),QLC(1,L1))                      C1C43000
      GO TO 26                                                         C1C43100
C                                                                      C1C43200
C        ZERO DEGREES OF FREEDOM                                       C1C43300
   25 L1 = L+1                                                         C1C43400
      L2 = L+2                                                         C1C43500
      LCON(L) = 0                                                      C1C43600
      LCON(L1) = 0                                                     C1C43700
      LCON(L2) = 0                                                     C1043800
      QL(1,L) = 1                                                      C1C43900
      QL(2,L1) = 1                                                     C1044000
      QL(3,L2) = 1                                                     C1044100
      IF(IC.EQ.0) GO TO 26                                             C1044200
      DO 28 II=1,3                                                     C1044300
      I = II-1                                                         C1C44400
      IF(LEQU)JL = L+I                                                 C1C44500
      IF(LEQU)PRINT 217, JL,MC,JL                                      C1C44600
   28 CALL VECTRN (QL(1,L+I),XMC(1,1,0),QLC(1,L+I))                    C1C44700
      GO TO 26                                                         C1C44800
C                                                                      C1C44900
   26 CONTINUE                                                         C1045000
      IF(IC.NE.0) GO TO 1025                                           C1C45100
      DO 30  I=1,3                                                     C1C45200
      MB2=M+2                                                          C1045300
      DO 32 J=M,MB2                                                    C1C45400
   31 QFC(I,J) = QF(I,J)                                               C1C45500
```

```
      LB2=L+2                                                          C1C45600
      CO 32 J=L,LB2                                                    C1C45700
   32 QLC(I,J) = QL(I,J)                                               C1C45800
   30 CONTINUE                                                         C1C45900
 1025 CONTINUE                                                         C1046000
      IF(.NOT. LEQU) GO TO 1015                                        C1C46100
      PRINT 221                                                        C1C46200
      CO 60  J=1,NFER                                                  C1C46300
   60 PRINT 222, J,(QF(I,J),I=1,3),J,(QFC(I,J),I=1,3)                  C1C46400
      PRINT 221                                                        C1046500
      IF(NLOR.EQ.0)GC TC 1015                                          C1C46600
      CO 61  J=1,NLOR                                                  C1C46700
   61 PRINT 223, J,(QL(I,J),I=1,3),J,(QLC(I,J),I=1,3)                  C1C46800
      PRINT 221                                                        C1C46900
C                                                                      C1C47000
C     CYCLE THROUGH FREE VECTORS PICK OUT ONES TO BE TRANSFORMED IN TRAN C1C47100
 1015 CO 33  J=1,NFER                                                  C1047200
   33 SVC(J) = 0                                                       C1C47300
      K = C                                                            C1C47400
      CC 34  M=1,NFER                                                  C1C47500
C     IS FREE VECTOR M FIXED INERTIALLY                                C1C47600
      IF(FCCN(M)) 35,36,37                                             C1C47700
   36 IF(IC.EG.0) GO TO 34                                             C1C47800
   35 K = K+1                                                          C1047900
      SVC(K) = M                                                       C1C48000
      GO TC 34                                                         C1C48100
C      ELEMENTS OF SVD IN SET(J)                                       C1C48200
   37 IF(CTAIN(FCON(M),SET,NSET)) GO TO 35                             C1C48300
   34 CONTINUE                                                         C1C48400
      NSVC = K                                                         C1C48500
      IF(.NOT. LEQU) GO TO 1005                                        C1048600
      IF(NSVC.EQ.0)GC TC 5016                                          C1C48700
      CO 64  I=1,NSVC                                                  C1048800
   64 PRINT 224, I,SVC(I)                                              C1048900
      PRINT 221                                                        C1C49000
C                                                                      C1C49100
C     CYCLE THROUGH LOCKED VECTORS PICK OUT ONES TO BE TRANSFORMED IN TRC1049200
 5016 IF(NLOR.EQ.0)GO TO 5017                                          C1C49300
 1005 CO 38  J=1,NLOR                                                  C1C49400
   38 SVF(J) = 0                                                       C1C49500
 5017 CONTINUE                                                         C1C49600
      K = C                                                            C1C49700
      IF(NLOR.EQ.0) GO TC 5025                                         C1C49800
      CO 39  L=1,NLOR                                                  C1C49900
C     IS CONSTRAINT TORQUE AECUT CL(I,L) COMPUTED                      C1C50000
      IF(.NOT.CTAIN(L,SLK,NCTC)) GO TO 39                              C1050100
      IF(LCON(L)) 40,41,42                                             C1050200
   41 IF(IC.EG.0) GO TO 39                                             C1C50300
   40 K = K+1                                                          C1C50400
      SVF(K) = L                                                       C1C50500
      GC TC 39                                                         C1050600
C      ELEMENTS OF SVD IN SET(J)                                       C1C50700
   42 IF(CTAIN(LCON(L),SET,NSET)) GO TO 40                             C1C50800
   39 CONTINUE                                                         C1C50900
 5025 CONTINUE                                                         C1C51000
      NSVP = K                                                         C1C51100
      IF(.NCT. LEQU) GO TO 1010                                        C1C51200
      IF(NSVP.EQ.0)GO TC 1010                                          C1C51300
      CO 65 I=1,NSVP                                                   C1C51400
   65 PRINT 225, I,SVP(I)                                              C1C51500
```

```
C                                                                       C1C51600
C       ZERO ALL ARRAY ELEMENTS OF GAM STORED UPPER TRIANGULAR          C1C51700
 101C DO 52  K=1,NBOD                                                   C1C51800
      IF(LEQU)PRINT 233                                                 C1C51900
      DO 52  L=K,NBOD                                                   C1C52000
      KL = KT3(NB1,K-1,L)                                               C1C52100
C     PUT IN LOGIC TO AVOID CLOBBERING CODE WORDS STORED BY EQUIV.      C1C52200
      IF(KL.EQ.1) GO TO 52                                              C1C52300
      IF(LEQU)K1 = K-1                                                  C1C52400
      IF(LEQU)PRINT 218,K1,L,KL                                         C1C52500
      DO 52 I=1,3                                                       C1C52600
      GAM(I,KL) = 0.D0                                                  C1C52700
   52 CONTINUE                                                          C1C52800
C                                                                       C1C52900
C       ZERO ALL ARRAY ELEMENTS OF XDIC STORED LOWER TRIANGULAR         C1C53000
      DO 53  K=1,NB1                                                    C1C53100
      IF(LEQU)PRINT 233                                                 C1C53200
      DO 53  I=K,NB1                                                    C1C53300
      IK = KT1(NB1,I,K)                                                 C1C53400
      IF(LEQU)PRINT 215, I,K,IK                                         C1C53500
      DO 53  M=1,3                                                      C1C53600
      DO 53  N=1,3                                                      C1C53700
   53 XDIC(M,N,IK) = 0.D0                                               C1C53800
      IF(LEQU)PRINT 221                                                 C1C53900
C                                                                       C1C54000
C                                                                       C1C54100
C                                                                       C1C54200
C         FIRST PASS THROUGH, FIND THE UNION OF ALL LABELS IN THE       C1C54300
C           SETS SIX(K),K=0,....,NEQD-1 AND THE SETS SCN(K),K=0,....,NBOD-1 C1C54400
      DO 55  I=1,NBOD                                                   C1C54500
      ST3(I) = 0                                                        C1C54600
   55 ST4(I) = 0                                                        C1C54700
      DO 56 JJJ=1,NBOD                                                  C1C54800
      J=JJJ-1                                                           C1C54900
      CALL UNPAC(ST1,NST1,SIX(J))                                       C1C55000
      CALL UNPAC(ST2,NST2,SCN(J))                                       C1C55100
      IF(NST1.EQ.0)GO TO 5018                                           C1C55200
      DO 57  I=1,NST1                                                   C1C55300
   57 ST3(ST1(I)) = 1                                                   C1C55400
 5018 IF(NST2.EQ.0)GO TO 5019                                           C1C55500
      DO 58  I=1,NST2                                                   C1C55600
   58 ST4(ST2(I)) = 1                                                   C1C55700
 5019 CONTINUE                                                          C1C55800
   56 CONTINUE                                                          C1C55900
      NST1 = 0                                                          C1C56000
      NST2 = 0                                                          C1C56100
      DO 59  J=1,NBOD                                                   C1C56200
      IF(ST3(J).EQ.0) GO TO 62                                          C1C56300
      NST1 = NST1 +1                                                    C1C56400
      ST1(NST1) = J                                                     C1C56500
   62 IF(ST4(J).EQ.0) GO TO 59                                          C1C56600
      NST2 = NST2 +1                                                    C1C56700
      ST2(NST2) = J                                                     C1C56800
   59 CONTINUE                                                          C1C56900
      CALL COMPAC(ST1,NST1,SSIX)                                        C1C57000
      CALL COMPAC(ST2,NST2,SSCN)                                        C1C57100
      IF(LEQU)PRINT 219, SSIX,(ST1(I),I=1,NST1)                         C1C57200
      IF(LEQU)PRINT 220, SSCN,(ST2(I),I=1,NST2)                         C1C57300
C                                                                       C1C57400
C                                                                       C1C57500
```

```
C                                                                       C1C57600
      DO 63  K=1,NBOD                                                   01C57700
   63 SOK(K) = SI(K)                                                    C1C57800
      CALL UNPAC(ST1,NST1,SKDUM)                                        C1C57900
      NST1 = NST1 +1                                                    C1C58000
      ST1(NST1) = NB1                                                   C1C58100
      CALL COMPAC(ST1,NST1,SCK(NB1))                                    C1C58200
C                                                                       C1C58300
C     COMPUTE ACTUAL POSITION OF POINT MASSES                           C1C58400
C          NOMINAL CAC(K) + DISPLACED THA(M)*QFC(M)                     C1C58500
      CALL UNPAC(SET,NSET,SL)                                           C1C58600
      IF(NSET.EQ.0)GO TO 5026                                           C1C58700
      DO 7C  KK=1,NSET                                                  C1C58800
      K = SET(KK)                                                       C1C58900
      MM=SCF(K)                                                         01C59000
      MMM=MM+2-PCUN(K)                                                  C1C59100
      DO 72 M=MM,MMM                                                    C1C59200
      DO 71  I=1,3                                                      C1C59300
   71 CAC(I,K) = CAC(I,K) + THA(M)*QFC(I,M)                             C1C59400
      IF(LEQU)PRINT 228, K,K,M,M                                        C1C59500
   72 CONTINUE                                                          C1C59600
   7C CONTINUE                                                          C1C59700
 5026 CONTINUE                                                          C1C59800
C                                                                       C1C59900
C                                                                       C1C60000
C       ANGULAR MOMENTUM WHEEL                                          C1C60100
      IF(LEQU)PRINT 226                                                 C1C60200
      IF(NMC.EQ.0)GO TO 5020                                            C1C60300
      DO 43  J=1,NMC                                                    C1C60400
   43 S(J) = C                                                          C1C60500
 5020 CONTINUE                                                          C1C60600
      K = C                                                             C1C60700
      IF(NMO.EQ.0) GO TO 5027                                           C1C60800
      DO 44  J=1,NMO                                                    C1C60900
C        HM(I,J) = COMPONENTS OF SPIN AXIS                              C1C61000
C        HMOM(J) = RELATIVE MOMENTUM I.V. ABOUT SPIN AXIS               C1C61100
      CALL VECTRN (HM(1,J),XMC(1,1,MO(J)),HMC(1,J))                     C1C61200
C       HMC(I,J) = SPIN AXIS COMPONENTS IN COMPUTING FRAME              C1C61300
      IF(LEQU)PRINT 227, J,MO(J),J                                      C1C61400
      JJ = MO(J)                                                        C1C61500
C      ELEMENTS OF SVD IN SET(J)                                        C1C61600
      IF(.NOT.CTAIN(JJ,SET,NSET)) GO TO 44                              C1C61700
      K = K+1                                                           C1C61800
      S(K) = J                                                          C1C61900
   44 CONTINUE                                                          C1C62000
 5027 CONTINUE                                                          C1C62100
      CALL COMPAC(S,K,SVM)                                              01C62200
      IF(LEQU)PRINT 229, SVM, (S(I),I=1,K)                              C1C62300
C          USE SVM ELEMENTS IN TRANVD TO GET HMC                        C1C62400
      IF(LEQU)PRINT 233                                                 C1C62500
      IF(.NOT. LEQU) RETURN                                             C1C62600
      IF(NMC.EQ.0) RETURN                                               C1C62700
      DO 66  J=1,NMC                                                    C1C62800
   66 PRINT 230, J,(HMC(I,J),I=1,3)                                     C1C62900
      PRINT 233                                                         C1C63000
  200 FORMAT ('1  SUBROUTINE VCIV ENTERED ',2(/))                       C1C63100
  201 FORMAT ('  COMPUTING FRAME IS BODY ',I ',/)                       C1C63200
  202 FORMAT ('  COMPUTING FRAME INERTIALLY FIXED ',/)                  C1C63300
  203 FORMAT ('  CAC(',I2,') = XMC(',I2,') * CA(',I2,') = ',3D15.5)     C1C63400
  204 FORMAT (2(/),                                                     C1C63500
```

```
     *          '          CENTER OF MASS VECTORS ',/)                    C1C63600
 205 FORMAT (2(/),                                                        C1C63700
     *          '                INERTIA TENSORS ',/)                     C1C63800
 206 FORMAT (44X,3D15.5)                                                  C1C63900
 207 FORMAT ('  XIC(',I2,') = XMC(',I2,') * XI(',I2,') * XMC(',I2,')**T C1C64000
    *= ',3D15.5)                                                         C1C64100
 208 FORMAT (/,'  SVD = ',Z8,' ELEMENTS OF SET ARE ',10I5,/)             C1C64200
 209 FORMAT (/,'  SVI = ',Z8,' ELEMENTS OF SET ARE ',10I5,/)             C1C64300
 210 FORMAT (2(/),                                                       C1C64400
     *          '              HINGE VECTORS ',/)                        C1C64500
 211 FORMAT ('  CBC(',I2,') = XMC(',I2,') * CB(',I2,') = ',3D15.5)      C1C64600
 212 FORMAT (/,'  SVB = ',Z8,' ELEMENTS OF SET ARE ',10I5,/)             C1C64700
 213 FORMAT (2(/),                                                       C1C64800
     *          '       FREE AND LOCKED COORDINATE VECTORS ',/)          C1C64900
 214 FORMAT ('  OFC(',I2,') = XMC(',I2,') * OF(',I2,') ')               C1C65000
 215 FORMAT ('  XDIC(',I2,',',I2,') = XDIC(',I2,') = 0 ')               C1C65100
 216 FORMAT ('  OFC(',I2,') = NORM(OFC(',I2,') X OFC(',I2,')) ')        C1C65200
 217 FORMAT (65X,'  OLC(',I2,') = XMC(',I2,') * OL(',I2,') ')           C1C65300
 218 FORMAT ('  GAM(',I2,',',I2,') = GAM(',I2,') = 0 ')                 C1C65400
 219 FORMAT (5X,'  SSIX = ',Z8,' UNION OF ALL LABELS IN THE SETS SIX(K)C1C65500
    *ARE ',10I5)                                                         C1C65600
 220 FORMAT (5X,'  SSCN = ',Z8,' UNION OF ALL LABELS IN THE SETS SCN(K)C1C65700
    * ARE ',10I5)                                                       C1C65800
 221 FORMAT (3(/))                                                       C1C65900
 222 FORMAT ('  OF(',I2,') = ',3D15.5,5X,'  OFC(',I2,') = ',3D15.5)     C1C66000
 223 FORMAT ('  OL(',I2,') = ',3D15.5,5X,'  OLC(',I2,') = ',3D15.5)     C1C66100
 224 FORMAT ('  SVO(',I2,') = ',I5)                                     C1C66200
 225 FORMAT ('  SVP(',I2,') = ',I5)                                     C1C66300
 226 FORMAT (3(/),25X,' MOMENTUM WHEELS ',/)                            C1C66400
 227 FORMAT ('  HMC(',I2,') = XMC(',I2,') * HM(',I2,') ')               C1C66500
 228 FORMAT ('  CAC(',I2,') = CAC(',I2,') + THA(',I2,')*OFC(',I2,') = 'C1C66600
    *,3D15.5)                                                            C1C66700
 229 FORMAT (2(/),                                                       C1C66800
     *          '  SVM = ',Z8,' ELEMENTS OF SET ARE ',10I5)             C1C66900
 230 FORMAT ('  HMC(',I2,') = ',3D15.5)                                 C1C67000
 231 FORMAT (/,'  SFLX =',Z8,' ELEMENTS OF SET ARE ',10I5,/)             C1C67100
 232 FORMAT (39X,'  CAO(',I2,') = CA(',I2,') ')                         C1C67200
 233 FORMAT ('    ')                                                     C1C67300
 234 FORMAT (39X,'  XIC(',I2,') = XI(',I2,') ')                         C1C67400
 235 FORMAT (65X,'  OLC(',I2,') = NORM(OFC(',I2,') X OFC(',I2,')) ')   C1C67500
 236 FORMAT (35X,'    OF(',I2,') = OF(',I2,') X OF(',I2,') ')           C1C67600
 237 FORMAT (35X,'    OL(',I2,') = OF(',I2,') X OF(',I2,') ')           C1C67700
 238 FORMAT (35X,'    OL(',I2,') = OF(',I2,') X OL(',I2,') ')           C1C67800
 239 FORMAT (35X,'    OL(',I2,') = XMT(',I2,')**T * OF(',I2,') X OL(',I1C67900
    *2,') ')                                                             C1C68000
 240 FORMAT (35X,'    OL(',I2,') = OL(',I2,') X OL(',I2,') ')           C1C68100
 241 FORMAT (/,'  SVA = ',Z8,' ELEMENTS OF SET ARE ',10I5,/)             C1C68200
 242 FORMAT ('  FLE(',I2,') = FLD(',I2,') + FLD(',I2,')**T =',27X,3D12.C1C68300
    *5)                                                                  C1C68400
 243 FORMAT ('  FLH(',I2,') = FLD(',I2,') + FLJ(',I2,') =',30X,3D12.5)  C1C68500
 244 FORMAT (61X,3D12.5)                                                 C1C68600
 245 FORMAT ('  FLO(',I2,') = FLC(',I2,') + XMAS(',I2,')*(FLB(',I2,') -C1C68700
    * CAO(',I2,') X FLA(',I2,') =',3D12.5)                              C1C68800
 246 FORMAT (30X,'  CA(',I2,') = CA(',I2,') + THA(',I2,')*FLA(',I2,')')C1C68900
 247 FORMAT (30X,'  XI(',I2,') = XI(',I2,') + THA(',I2,')*FLE(',I2,')')C1C69000
 248 FORMAT ('  FLAC(',I2,') = XMC(',I2,')*FLA(',I2,') =',31X,3D12.5)  C1C69100
 249 FORMAT ('  FLOC(',I2,') = XMC(',I2,')*FLO(',I2,') =',31X,3D12.5)  C1C69200
 252 FORMAT (30X,'  CA(',I2,') = ',3D12.5)                              C1C69300
 254 FORMAT (41X,3D12.5)                                                 C1C69400
 255 FORMAT (30X,'  XI(',I2,') = ',3D12.5)                              C1C69500
```

```
 256 FORMAT (/,'  SVA = ',Z8,' ELEMENTS OF SET ARE ',10I5,/)                C1C69600
 257 FORMAT (/////,40X,'FLEXIELE BODY ',I3,//)                             C1C69700
 258 FORMAT (30X,'  BODY ',I3,' EFFECTS OF ELASTIC MODE ',I3,/)            C1C69800
 259 FORMAT (15X,'  ELASTICALLY DEFORMED CENTER OF MASS VECTOR AND INER    C1C69900
    *TIA TENSOR FOR BODY ',I3,/)                                           C1C70000
     RETURN                                                               C1C70100
     END                                                                  C1C70200
C                                                                         C1100000
      SUBROUTINE EQIV(Y,NEQ)                                              C1100100
C     USED TO SET UP INITIAL VALUES FOR RUNGE                             C1100200
C                                                                         C1100300
C                                                                         C1100400
      IMPLICIT REAL*8(A-H,O-Z,$)                                          C1100500
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)       C1100600
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,       C1100700
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,       C1100800
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ          C1100900
C                                                                         C1101000
C                                                                         C1101100
      INTEGER                                                             C1101200
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , FCCN  ,   C1101300
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,   C1101400
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,   C1101500
     * SMA   , SMCDUM, SMC   , SMV   , SQK   , SPIDUM, SPI   , SQF   ,   C1101600
     * SQL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,   C1101700
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,   C1101800
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,   C1101900
     * IINIT(1)      , IZINIT(1)     , SQ                                 C1102000
C                                                                         C1102100
C                                                                         C1102200
      REAL*8                                                              C1102300
     * ANGD  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,   C1102400
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                   C1102500
     * THACD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)        C1102600
C                                                                         C1102700
C                                                                         C1102800
C                                                                         C1102900
C                                                                         C1103000
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,       C1103100
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,       C1103200
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ          C1103300
C                                                                         C1103400
C                                                                         C1103500
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)             C1103600
C                                                                         C1103700
C                                                                         C1103800
      COMMON /INTG/    AWORK (200)  ,                                     C1103900
     * CT1   (33)    , CT2   (33)    , CT3   (33)    , CT4   (22)    ,   C1104000
     * CT5           , FCON  (33)    , JCON  (10)    , LCON  (22)    ,   C1104100
     * MC    (10)    , NB1           , NBOD          , NCTC          ,   C1104200
     * NFER          , NFKC          , NFRC          , NLOR          ,   C1104300
     * NMV           , NMO           , NMOA          , NSVP          ,   C1104400
     * NSVC          , FCON  (11)    , SD            , SFR   (33)    ,   C1104500
     * SG            , SI    (55)    , SIG           , SL            ,   C1104600
     * SLK   (33)    , SMA   (10)    , SQK   (11)    , SQF   (11)    ,   C1104700
```

```
     * SCL    (11)      , SMV              , SR               , SSCN             ,  C1104800
     * SSIX              , SVA              , SVB              , SVD              ,  C1104900
     * SVI               , SVM              , SVP    (22)      , SVQ    (33)      ,  C1105000
     * SXM    (3,10)     , SXT              , TURQ   (97)      , SMAL             ,  C1105100
     * SEU               , NTQ              , SC     (33)      , SCG              ,  C1105200
     * NFLXB             , SFLX             , SFXM   (10)      , NMODS            ,  C1105300
     * SFCC              , SCC    (10)                                             C1105400
C                                                                                  C1105500
C                                                                                  C1105600
      COMMON /INTGZ/                                                               C1105700
     * SCNDUM            , SCN    (9)       , SCRDUM           , SCR    (9)       ,  C1105800
     * SFKDUM            , SFK    (9)       , SIXDUM           , SIX    (9)       ,  C1105900
     * SKDUM             , SK     (9)       , SPIDUM           , SPI    (9)       ,  C1106000
     * SMCDUM            , SMC    (9)                                              C1106100
C                                                                                  C1106200
C                                                                                  C1106300
      COMMON /REAL/                                                                C1106400
     * CA     (3,10)     , CAC    (3,10)    , CLM    (10)      , CCMC   (3,11)    ,  C1106500
     * DCMC   (3,11)     , ETC    (3,11)    , ETM    (33)      , FUMC   (3,11)    ,  C1106600
     * GAM    (3,66)     , H                , HM     (3,10)    , HMC    (3,10)    ,  C1106700
     * HMCM   (10)       , PHI    (3,11)    , PLM    (10)      , QF     (3,33)    ,  C1106800
     * QFC    (3,33)     , QL     (3,22)    , QLC    (3,22)    , RCMC   (3,11)    ,  C1106900
     * T                 ,                    THA    (33)      , THAD   (33)      ,  C1107000
     * THADW  (10)       , THAW   (10)      , XDIC   (3,3,66)  , XI     (3,3,10)  ,  C1107100
     * XIC    (3,3,10)   , XMAS   (10)      , XMN    (33,33)   , XMT    (3,3,10)  ,  C1107200
     * TUG    (33)       , FLA    (3,20)    , FLB    (3,20)    , FLC    (3,20)    ,  C1107300
     * FLD    (3,3,20)   , FLJ    (3,3,20)  , CAD    (3,10)    , XID    (3,3,10)  ,  C1107400
     * FLIRC  (3,10)     , FLCRC  (3,10)    , FLAC   (3,20)    , FLQC   (3,20)    ,  C1107500
     * FLCM   (20)       , ZETA   (20)      , FCF    (3,3,40)  , FCK    (3,40)    ,  C1107600
     * TIMEND                                                                      C1107700
C                                                                                  01107800
C                                                                                  C1107900
      COMMON /REAL2/                                                               C1108000
     * CEDUM  (1,3)      , CB     (3,10)    , CBCDUM(1,3)      , CBC    (3,10)    ,  C1108100
     * XMCDUM(1,1,9)     , XMC    (3,3,10)  , CBN(3)                               C1108200
C                                                                                  C1108300
C                                                                                  C1108400
      EQUIVALENCE (ETM(1),THACC(1))          ,(XMN(1,1),ANGD(1))      ,           C1108500
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,          C1108600
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,          C1108700
     *            (FLB(1,1),FLQ(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),           C1108800
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                   C1108900
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,          C1109000
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,          C1109100
     *            (SCNDUM,IZINIT(1))                                             C1109200
C                                                                                  C1109300
      INTEGER ST1(11)                                                              C1109400
      EQUIVALENCE (LEQUIV,LEQU)                                                    C1109500
C                                                                                  C1109600
      DIMENSION Y(NEQ)                                                             C1109700
C                                                                                  C1109800
C                                                                                  C1109900
C     ANGULAR OR LINEAR RATE WITH RESPECT TO FREE AXES                             C1110000
      IF(LEQU)PRINT 100                                                            C1110100
      IF(LEQU)PRINT 101, NFER                                  ADDR:1-4            C1110200
      DO 1  N=1,NFER                                                               C1110300
      Y(N) = THAD(N)                                                               C1110400
      IF(LEQU)PRINT 102, N,N,Y(N)                                                  C1110500
    1 CONTINUE                                                                     C1110600
      NN = NFER                                                                    C1110700
```

```
C                                                                       C1110800
C                                                                       C1110900
C     GENERALIZED ELASTIC COORDINATE RATE EQUATIONS                     C1111000
      IF(LEQU) PRINT 112, NMODS                                         C1111100
      IF(NMODS.EQ.0) GO TO 7                                            C1111200
      DO 7 N=1,NMODS                                                    C1111300
      NA = NN + N                                                       C1111400
      Y(NA) = THAD(NA)                                                  C1111500
      IF(LEQU) PRINT 102, NA, NA, Y(NA)                                 C1111600
    7 CONTINUE                                                          C1111700
      NN = NN + NMODS                                                   C1111800
C                                                                       C1111900
C                                                                       C1112000
C     RELATIVE ANGULAR MOMENTUM OF WHEEL                                C1112100
      CALL UNPAC(ST1,NST1,SMV)                                          C1112200
      IF(LEQU)PRINT 103, NST1                                           C1112300
      IF(NST1.EQ.0) GO TO 5028                                          C1112400
      DO 2 NNN=1,NST1                                                   C1112500
      N=NST1-(NNN-1)                                                    C1112600
      NA = NN + NST1 + 1 - N                                            C1112700
      Y(NA) = THADW(ST1(N))                                             C1112800
      IF(LEQU)PRINT 104, NA,ST1(N),Y(NA)                                C1112900
    2 CONTINUE                                                          C1113000
 5028 CONTINUE                                                          C1113100
      NN = NN + NMV                                                     C1113200
C                                                                       C1113300
C     DISPLACEMENT ABOUT OR ALONG FREE AXES                             C1113400
      IF(LEQU)PRINT 105, NFRC                                           C1113500
      IF(NFRC.EQ.0) GO TO 5029                                          C1113600
      DO 3  N=1,NFRC                                                    C1113700
      M = SFR(N)                                                        C1113800
      NA = NN+N                                                         C1113900
      Y(NA) = THA(M)                                                    C1114000
      IF(LEQU)PRINT 106, NA,M,Y(NA)                                     C1114100
    3 CONTINUE                                                          C1114200
 5029 CONTINUE                                                          C1114300
      NN = NN+NFRC                                                      C1114400
C                                                                       C1114500
C                                                                       C1114600
C     GENERALIZED ELASTIC COORDINATE DISPLACEMENT EQUATIONS             C1114700
      IF(LEQU) PRINT 112, NMODS                                         C1114800
      IF(NMODS.EQ.0) GO TO 9                                            C1114900
      DO 9 N=1,NMODS                                                    C1115000
      NA = NN + N                                                       C1115100
      NST1 = NFER + N                                                   C1115200
      Y(NA) = THA(NST1)                                                 C1115300
      IF(LEQU) PRINT 106, NA,NST1,Y(NA)                                 C1115400
    9 CONTINUE                                                          C1115500
      NN = NN + NMODS                                                   C1115600
C                                                                       C1115700
C                                                                       C1115800
C     DISPLACEMENT ABOUT WHEEL SPIN AXIS                                C1115900
      IF(LEQU)PRINT 107, NMOA                                           C1116000
      IF(NMOA.EQ.0) GO TO 5030                                          C1116100
      DO 4  N=1,NMOA                                                    C1116200
      M = SMA(N)                                                        C1116300
      NA = NN+N                                                         C1116400
      Y(NA) = THAW(M)                                                   C1116500
      IF(LEQU)PRINT 108, NA,M,Y(NA)                                     C1116600
    4 CONTINUE                                                          C1116700
```

```
 5030 CONTINUE                                                          C1116800
      NN = NN+ NMOA                                                     C1116900
C                                                                       C1117000
C     DIRECTION COSINES                                                 C1117100
      CALL UNPAC(ST1,NST1,SD)                                           C1117200
      IF(LEQU)PRINT 109, SD,(ST1(I),I=1,NST1)                           C1117300
      IF(LEQU)PRINT 110, INERF                                          C1117400
      IF(NST1.EQ.0) GO TO 5031                                          C1117500
      N = NN                                                            C1117600
      M = C                                                             C1117700
      IF(INERF.OR.ST1(NST1).NE.1) GO TO 5                               C1117800
      NST1 = NST1 - 1                                                   C1117900
      DO 8  J=1,2                                                       C1118000
      DO 8  I=1,3                                                       C1118100
      NN = NN +1                                                        C1118200
      Y(NN) = XMC(I,J,M)                                                C1118300
      IF(LEQU)PRINT 111, NN,I,J,M,Y(NN)                                 C1118400
    8 CONTINUE                                                          C1118500
    5 IF(NST1.EQ.0) GO TO 5031                                          C1118600
      DO 6 N=1,NST1                                                     C1118700
      M = ST1(N)                                                        C1118800
      DO 6  J=1,2                                                       C1118900
      DO 6  I=1,3                                                       C1119000
      NN = NN + 1                                                       C1119100
      Y(NN) = XMC(I,J,M)                                                C1119200
      IF(LEQU)PRINT 111, NN,I,J,M,Y(NN)                                 C1119300
    6 CONTINUE                                                          C1119400
 5031 CONTINUE                                                          C1119500
      NEQ = NN                                                          C1119600
C                                                                       C1119700
  100 FORMAT ('1 ENTER SUBROUTINE EQIV ')                               C1119800
  101 FORMAT ('   NFER =',I5)                                           C1119900
  102 FORMAT ('  Y(',I2,') = THAD(',I2,') = ',D15.5)                    C1120000
  103 FORMAT ('  NST1 =',I5)                                           C1120100
  104 FORMAT ('  Y(',I2,') = THADW(',I2,') = ',D15.5)                   C1120200
  105 FORMAT ('  NFRC = ',I5)                                          C1120300
  106 FORMAT ('  Y(',I2,') = THA(',I2,') = ',D15.5)                     C1120400
  107 FORMAT ('  NMOA = ',I5)                                          C1120500
  108 FORMAT ('  Y(',I2,') = THAW(',I2,') = ',D15.5)                    C1120600
  109 FORMAT ('  SD =',Z8,' ELEMENTS IN ARRAY ST1 ARE',10I5)            C1120700
  110 FORMAT ('  INERF = ',L10)                                        C1120800
  111 FORMAT ('  Y(',I2,') = XMC(',I2,',',I2,',',I2,') = ',D15.5)      C1120900
  112 FORMAT ('  NMODS =',I5)                                          C1121000
C                                                                       C1121100
      RETURN                                                            C1121200
      END                                                               C1121300
```

```
C                                                                       C1200000
      SUBROUTINE TRAN                                                   C1200100
C     COMPUTE ALL TRANSFORMATION MATRICES .      (2)     X2 .           C1200200
C        1) USE ORTHOGONALITY TO GET MISSING ELEMENTS                   C1200300
C        2) CONSTRUCT OTHERS NOT IN SD                                  C1200400
C                                                                       C1200500
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1200600
C                                                                       C1200700
C                                                                       C1200800
```

```
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)      C1200900
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,      C1201000
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,      C1201100
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C1201200
C                                                                        C1201300
C                                                                        C1201400
      INTEGER                                                            C1201500
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , PCON   ,  C1201600
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,  C1201700
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,  C1201800
     * SMA    , SMCDUM, SMC    , SMV    , SOK   , SPIDUM, SPI    , SQF    ,  C1201900
     * SGL    , SR     , SSCN   , SSIX   , SVA   , SVB    , SVD    , SVI    ,  C1202000
     * SVM    , SVP    , SVQ    , SXM    , SXT   , TORQ   , SMAL   , SEU    ,  C1202100
     * SC     , SCG    , NFLXB  , SFLX   , SFXM  , NMODS  , SFCC   , SCC    ,  C1202200
     * IINIT(1)        , IZINIT(1)      , SD                              C1202300
C                                                                        C1202400
C                                                                        C1202500
      REAL*8                                                             C1202600
     * ANGC   (33)     , CNF    (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C1202700
     * FLQ    (3,20)   , FLE    (3,3,20), FLH   (3,3,20),                C1202800
     * THACD  (33)     , YMCD   (3,2,11), RINIT (1)     , RZINIT(1)      C1202900
C                                                                        C1203000
C                                                                        C1203100
C                                                                        C1203200
C                                                                        C1203300
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,      C1203400
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,      C1203500
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C1203600
C                                                                        C1203700
C                                                                        C1203800
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)            C1203900
C                                                                        C1204000
C                                                                        C1204100
      COMMON /INTG/ AWORK(200) ,                                         C1204200
     * CT1            , CT2            , CT3            , CT4          ,  C1204300
     * CT5            , FCON   (33)    , JCON   (10)    , LCON   (22)  ,  C1204400
     * MC     (10)    , NBI            , NBOD           , NCTC         ,  C1204500
     * NFEF           , NFKC           , NFRC           , NLOR         ,  C1204600
     * NMV            , NMO            , NMCA           , NSVP         ,  C1204700
     * NSVC           , PCON   (11)    , SD             , SFR    (33)  ,  C1204800
     * SG             , SI     (55)    , SIG            , SL           ,  C1204900
     * SLK    (33)    , SMA    (10)    , SOK    (11)    , SQF    (11)  ,  C1205000
     * SGL    (11)    , SMV            , SR             , SSCN         ,  C1205100
     * SSIX           , SVA            , SVB            , SVD          ,  C1205200
     * SVI            , SVM            , SVP    (22)    , SVQ    (33)  ,  C1205300
     * SXM    (3,10)  , SXT            , TORQ   (97)    , SMAL         ,  C1205400
     * SEU            , NTQ            , SC     (33)    , SCG          ,  C1205500
     * NFLXB          , SFLX           , SFXM   (10)    , NMODS        ,  C1205600
     * SFCC           , SCC    (10)                                      C1205700
C                                                                        C1205800
C                                                                        C1205900
      COMMON /INTGZ/                                                     C1206000
     * SCNDUM         , SCN    (9)     , SCRDUM         , SCR    (9)    ,  C1206100
     * SFKDUM         , SFK    (9)     , SIXDUM         , SIX    (9)    ,  C1206200
005000*1SKDUM         , SK     (9)     , SPIDUM         , SPI    (9)    ,  C1206300
     * SMCDUM         , SMC    (9)                                       C1206400
C                                                                        C1206500
C                                                                        C1206600
      COMMON /REAL/                                                      C1206700
     * CA    ,(3,10)  , CAC    (3,10)  , CLM    (10)    , CCMC   (3,11) ,  C1206800
```

```
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FCMC   (3,11)   ,  C1206900
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)   ,  C1207000
     * HMCM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)   ,  C1207100
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RCMC   (3,11)   ,  C1207200
     * T                ,                  THA    (33)     , THAD   (33)     ,  C1207300
     * THADM  (10)     , THAM   (10)     , XCIC   (3,3,66), XI     (3,3,10),  C1207400
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C1207500
     * TLG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)   ,  C1207600
     * FLD    (3,3,20), FLJ    (3,3,20), CAO    (3,10)   , XIG    (3,3,10),  C1207700
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)   ,  C1207800
     * FLOM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)   ,  C1207900
     * TIMEND                                                                   C1208000
C                                                                               C1208100
C                                                                               C1208200
      COMMON /REALZ/                                                            C1208300
     * CEDLM (1,3)     , CB     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)   ,  C1208400
     * XMCDLM(1,1,9)   , XMC    (3,3,10), CBN(3)                               C1208500
C                                                                               C1208600
C                                                                               C1208700
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))       ,       C1208800
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))      ,      C1208900
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))    ,      C1209000
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),        C1209100
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                C1209200
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))       ,      C1209300
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))      ,      C1209400
     *            (SCNDLM,IZINIT(1))                                           C1209500
C                                                                               C1209600
      LOGICAL CTAIN                                                             C1209700
      DIMENSION TEMP1(3,3)                                                      C1209800
      DIMENSION XMS(3,3),Q(2),GS(2),TEM(3,2),TEMP(3,3)                          C1209900
      INTEGER ST1(10),SET(10)                                                   C1210000
      EQUIVALENCE (LTRAN,LEQU)                                                  C1210100
C                                                                               C1210200
C                                                                               C1210300
C                                                                               C1210400
      IF(.NOT. LEQU) GO TO 100                                                  C1210500
      PRINT 100                                                                 C1210600
      J0 = 0                                                                    C1210700
      J1 = 1                                                                    C1210800
      J2 = 2                                                                    C1210900
      J3 = 3                                                                    C1211000
      PRINT 101, INERF                                                          C1211100
 1000 CALL UNPAC (SET,NSET,SD)                                                  C1211200
      IF(LEQU)PRINT 102, SD,(SET(I),I=1,NSET)                                   C1211300
      IF(NSET.EQ.0) GO TO 5022                                                  C1211400
      KST = 1                                                                   C1211500
      IF(INERF.OR.SET(NSET).NE.1) GO TO 1                                       C1211600
      NSET = NSET - 1                                                           C1211700
      KST = 0                                                                   C1211800
      K = 0                                                                     C1211900
    1 KK=KST                                                                    C1212000
      IF(KST.GT.NSET)GO TO 5022                                                 C1212100
 5021 IF(KK.EQ.0) GO TO 19                                                      C1212200
      K = SET(KK)                                                               C1212300
   19 CALL VECROS (XMC(1,1,K),XMC(1,2,K),XMC(1,3,K))                            C1212400
      IF(.NOT. LEQU) GO TO 2                                                    C1212500
      PRINT 104                                                                 C1212600
      PRINT 116,    (XMC(1,I,K),I=1,3)                                          C1212700
      PRINT 116,K,  (XMC(2,I,K),I=1,3)                                          C1212800
```

```
      PRINT 115,   (XMC(3,I,K),I=1,3)                                   C1212900
    2 CONTINUE                                                          C1213000
      KK=KK+1                                                           C1213100
      IF(KK.LE.NSET) GO TO 5021                                         C1213200
 5022 CONTINUE                                                          C1213300
C                                                                       C1213400
C         COMPUTE ELEMENTS OF TRANSFORMATION MATRICES OBTAINABLE        C1213500
C         VIA SMALL ANGLE ASSUMPTIONS OR EULER ANGLES                   C1213600
C                                                                       C1213700
      IS = SR - SD                                                      C1213800
      CALL UNPAC(SET,NSET,IS)                                           C1213900
      CALL UNPAC(ST1,NST1,SMAL)                                         C1214000
      IF(.NOT. LEQU) GO TO 3                                            C1214100
      PRINT 107, IS,(SET(I),I=1,NSET)                                   C1214200
      PRINT 104                                                         C1214300
    3 IF(NSET.EQ.0) GO TO 5032                                          C1214400
      DO 4 III=1,NSET                                                   C1214500
      JJJ=NSET-(III-1)                                                  C1214600
      JJ = SET(JJJ)                                                     C1214700
      J = JCON(JJ)                                                      C1214800
      DO 20 I=1,3                                                       C1214900
      DO 20 L=1,3                                                       C1215000
   20 TEMP(I,L) = XMT(I,L,JJ)                                           C1215100
C     CHECK FOR THREE CONSTRAINED AXES                                  C1215200
      IF(PCON(JJ).NE.3) GO TO 5026                                      C1215300
      DO 27 I=1,3                                                       C1215400
      DO 27 L=1,3                                                       C1215500
      XMS(I,L) = TEMP(I,L)                                              C1215600
   27 CONTINUE                                                          C1215700
 5026 CONTINUE                                                          C1215800
      MDONE = 3-PCON(JJ)                                                C1215900
      DO 5 II=1,MDONE                                                   C1216000
      I = II-1                                                          C1216100
C     CYCLE THROUGH FREE COORDINATE ROTATIONS AT HINGE POINT JJ-1       C1216200
      M = SOF(JJ) + I                                                   C1216300
      KB = SXM(I+1,JJ)                                                  C1216400
C     CHECK SMALL ANGLE ASSUMPTIONS                                     C1216500
      B = THA(M)                                                        C1216600
      C = 1.000                                                         C1216700
      IF(CTAIN(JJ,ST1,NST1)) GO TO 22                                   C1216800
      B = DSIN(THA(M))                                                  C1216900
      C = DCOS(THA(M))                                                  C1217000
   22 CONTINUE                                                          C1217100
      IF(.NOT.LEQU) GO TO 5027                                          C1217200
      PRINT 120, JJ, J,I,M,KB,B,C                                       C1217300
  120 FORMAT (' JJ =',I3,' J =',I3,' I =',I3,' M =',I3,' KB =',I3,' B =' C1217400
     *,D17.8,' C =',D17.8)                                              01217500
 5027 CONTINUE                                                          C1217600
      CALL ROT(B,C,KB,TEMP1)                                            C1217700
      CALL MATMUL(TEMP,TEMP1,XMS,3)                                     C1217800
      DO 25 L=1,3                                                       C1217900
      DO 25 LL=1,3                                                      C1218000
      TEMP(L,LL) = XMS(L,LL)                                            C1218100
   25 CONTINUE                                                          C1218200
    5 CONTINUE                                                          C1218300
      IF(LEQU) PRINT 118, ((TEMP(I,L),L=1,3),I=1,3)                     C1218400
      CALL MATMUL(XMT(1,1,JJ),TEMP,XMS,3)                               C1218500
      IF(.NOT. LEQU) GO TO 1002                                         C1218600
      PRINT 104                                                         C1218700
      PRINT 105, (XMS(1,I),I=1,3)                                       C1218800
```

```
      PRINT 110, JJ,(XMS(2,I),I=1,3)                                    C1218900
      PRINT 105, (XMS(3,I),I=1,3)                                       C1219000
      PRINT 104                                                         C1219100
 1002 IF(J.NE.0) GO TO 16                                               C1219200
      IF(INERF) GO TO 17                                                C1219300
      DO 15 I=1,3                                                       C1219400
      DO 15 L=1,3                                                       C1219500
   15 XMC(I,L,0) = XMS(L,I)                                             C1219600
      IF(.NOT. LEQU) GO TO 4                                            C1219700
      PRINT 112, (XMC(I,I,0),I=1,3)                                     C1219800
      PRINT 113, J,JJ,(XMC(2,I,0),I=1,3)                                C1219900
      PRINT 112, (XMC(3,I,0),I=1,3)                                     C1220000
      PRINT 104                                                         C1220100
      GO TO 4                                                           C1220200
   17 DO 18  I=1,3                                                      C1220300
      DO 18  L=1,3                                                      C1220400
   18 XMC(I,L,1)=XMS(I,L)                                               C1220500
      IF(.NOT. LEQU) GO TO 4                                            C1220600
      PRINT 112,(XMC(1,I,1),I=1,3)                                      C1220700
    1 PRINT 114, JJ,JJ,(XMC(2,I,1),I=1,3)                               C1220800
      PRINT 112,(XMC(3,I,1),I=1,3)                                      C1220900
      GO TO 4                                                           C1221000
   16 CALL MATMUL(XMC(1,1,J),XMS,XMC(1,1,JJ),3)                         C1221100
      IF(.NOT. LEQU) GO TO 4                                            C1221200
      PRINT 105,(XMC(1,I,JJ),I=1,3)                                     C1221300
      PRINT 106, JJ,J,J,(XMC(2,I,JJ),I=1,3)                             C1221400
      PRINT 105,(XMC(3,I,JJ),I=1,3)                                     C1221500
      PRINT 104                                                         C1221600
    4 CONTINUE                                                          C1221700
 5032 CONTINUE                                                          C1221800
C                                                                       C1221900
C     UPDATE TRANSFORMATION MATRICES ASSOCIATED WITH LINEAR OSCILLATORS C1222000
      CALL UNPAC(SET,NSET,SL)                                           C1222100
      IF(NSET.EQ.0) GO TO 5033                                          C1222200
      DO 13  JJJ=1,NSET                                                 C1222300
      JJ = SET(JJJ)                                                     C1222400
      J  = JCON(JJ)                                                     C1222500
      DO 14  I=1,3                                                      C1222600
      DO 14  L=1,3                                                      C1222700
   14 XMC(I,L,JJ) = XMC(I,L,J)                                          C1222800
      IF(LEQU) PRINT 111, JJ,J                                          C1222900
   13 CONTINUE                                                          C1223000
 5033 CONTINUE                                                          C1223100
      IF(LEQU) PRINT 104                                                C1223200
  100 FORMAT ('1  SUBROUTINE TRAN ENTERED ',//)                         C1223300
  101 FORMAT ('  INERF = ',L5,/)                                        C1223400
  102 FORMAT ('  SD = ',I8,' SET ELEMENTS ARE ',10I5,//)                 C1223500
  104 FORMAT ('   ')                                                    C1223600
  105 FORMAT (30X,3D17.8)                                               C1223700
  106 FORMAT ('  XMC(',I2,') = XMC(',I2,')*XMS(',I2,') = ',3D17.8)      C1223800
  107 FORMAT ('     SR-SD = ',I8,' SET ELEMENTS ARE ',10I5,//)          C1223900
  109 FORMAT ( 12X,3D17.8)                                              C1224000
  110 FORMAT ('  XMS(',I2,') = ',3D17.8)                                C1224100
  111 FORMAT (/,'  XMC(',I2,') = XMC(',I2,') ')                        C1224200
  112 FORMAT (25X,3D17.8)                                               C1224300
  113 FORMAT ('  XMC(',I2,') = XMS(',I2,')**T =',3D17.8)                C1224400
  114 FORMAT ('  XMC(',I2,') = XMS(',I2,')    = ',3D17.8)              C1224500
  115 FORMAT (12X,3D17.8)                                               C1224600
  116 FORMAT ('  XMC(',I2,') = ',3D17.8)                                C1224700
  118 FORMAT ('  TEMP =',3D17.8)                                        C1224800
```

```
      RETURN                                                            C1224900
      END                                                               C1225000
C                                                                       C1300000
      SUBROUTINE TRANVD                                                 C1300100
C     TRANSFORMS ONLY THOSE VECTORS AND DYADS TIME VARYING IN           C1300200
C      COMPUTING FRAME                                                  C1300300
C                                                                       C1300400
C                                                                       C1300500
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1300600
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C1300700
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1300800
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1300900
     *                 LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1301000
C                                                                       C1301100
C                                                                       C1301200
      INTEGER                                                           C1301300
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , PCON   ,  C1301400
     * SCNDUM, SCN    , SCRDUM , SCR    , SFKDUM , SFK    , SFR    , SG     ,  C1301500
     * SI     , SIG    , SIXDUM , SIX    , SKDUM  , SK     , SL     , SLK    ,  C1301600
     * SMA    , SMCDUM , SMC    , SMV    , SOK    , SPIDUM , SPI    , SQF    ,  C1301700
     * SGL    , SR     , SSCN   , SSIX   , SVA    , SVB    , SVD    , SVI    ,  C1301800
     * SVM    , SVP    , SVQ    , SXM    , SXT    , TORQ   , SMAL   , SEU    ,  C1301900
     * SC     , SCG    , NFLXB  , SFLX   , SFXM   , NMODS  , SFCC   , SCC    ,  C1302000
     * IINIT(1)        , IZINIT(1)      , SD                            C1302100
C                                                                       C1302200
C                                                                       C1302300
      REAL*8                                                            C1302400
     * ANGC  (33)     , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C1302500
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                 C1302600
     * THACD (33)     , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C1302700
C                                                                       C1302800
C                                                                       C1302900
C                                                                       C1303000
C                                                                       C1303100
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1303200
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1303300
     *                 LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1303400
C                                                                       C1303500
C                                                                       C1303600
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C1303700
C                                                                       C1303800
C                                                                       C1303900
      COMMON /INTG/ AWORK(200),                                         C1304000
     * CT1            , CT2            , CT3            , CT4          ,  C1304100
     * CT5            , FCON  (33)     , JCON  (10)     , LCON  (22)   ,  C1304200
     * MC     (10)    , NB1            , NBOD           , NCTC         ,  C1304300
     * NFER           , NFKC           , NFRC           , NLOR         ,  C1304400
     * NMV            , NMO            , NMOA           , NSVP         ,  C1304500
     * NSVC           , PCON  (11)     , SD             , SFR   (33)   ,  C1304600
     * SG             , SI    (55)     , SIG            , SL           ,  C1304700
     * SLK    (33)    , SMA   (10)     , SOK   (11)     , SQF   (11)   ,  C1304800
     * SGL    (11)    , SMV            , SR             , SSCN         ,  C1304900
     * SSIX           , SVA            , SVB            , SVD          ,  C1305000
     * SVI            , SVM            , SVP   (22)     , SVQ   (33)   ,  C1305100
     * SXM    (3,10)  , SXT            , TORQ  (97)     , SMAL         ,  C1305200
```

```
     * SEU             , NTQ             , SC     (33)     , SCG             ,  C1305300
     * NFLXB           , SFLX            , SFXM   (10)     , NMCDS           ,  C1305400
     * SFCC            , SCC    (10)                                           C1305500
C                                                                              C1305600
C                                                                              C1305700
      COMMON /INTGZ/                                                           C1305800
     * SCNDUM          , SCN    (9)      , SCRDUM          , SCR    (9)      ,  C1305900
     * SFKDUM          , SFK    (9)      , SIXDUM          , SIX    (9)      ,  C1306000
     * SKDUM           , SK     (9)      , SPIDUM          , SPI    (9)      ,  C1306100
     * SMCDUM          , SMC    (9)                                           C1306200
C                                                                              C1306300
C                                                                              C1306400
      COMMON /REAL/                                                            C1306500
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , CCMC   (3,11)   ,  C1306600
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FUMC   (3,11)   ,  C1306700
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)   ,  C1306800
     * HMOM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)   ,  C1306900
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RCMC   (3,11)   ,  C1307000
     * T                ,                  THA    (33)     , THAD   (33)     ,  C1307100
     * THADW  (10)     , THAW   (10)     , XDIC   (3,3,66), XI     (3,3,10),  C1307200
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C1307300
     * TUG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)   ,  C1307400
     * FLD    (3,3,20), FLJ    (3,3,20), CAU    (3,10)   , XIU    (3,3,10),  C1307500
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)   ,  C1307600
     * FLCM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)   ,  C1307700
     * TIMEND                                                                  C1307800
C                                                                              C1307900
C                                                                              C1308000
      COMMON /REALZ/                                                           C1308100
     * CEDUM (1,3)     , CE     (3,10)   , CBCDUM(1,3)     , CBC    (3,10)   ,  C1308200
     * XMCDUM(1,1,9)   , XMC    (3,3,10), CBN(3)                              C1308300
C                                                                              C1308400
C                                                                              C1308500
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANCD(1))      ,       C1308600
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,       C1308700
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,       C1308800
     *            (FLB(1,1),FLQ(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),        C1308900
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                C1309000
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))      ,       C1309100
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,       C1309200
     *            (SCNDUM,IZINIT(1))                                          C1309300
C                                                                              C1309400
      LOGICAL CTAIN                                                            C1309500
      DIMENSION TEM(3)                                                         C1309600
      DIMENSION TEM1(3,3)                                                      C1309700
      INTEGER ST1(10),SET(18),SFXMN                                            C1309800
      EQUIVALENCE (LTRANV,LEQU)                                                C1309900
C                                                                              C1310000
      IF(LEQU)PRINT 200                                                        C1310100
C                                                                              C1310200
C                                                                              C1310300
C     ELASTIC DEFORMATION PARAMETERS                                           C1310400
      IF(NFLXB.EQ.0) GO TO 15                                                  C1310500
      NN = 0                                                                   C1310600
      CALL UNPAC(SET,NSET,SFLX)                                                C1310700
      CALL UNPAC(ST1,NST1,SVD)                                                 C1310800
      IF(NSET.EQ.0) GO TO 5000                                                 C1310900
      DO 16 NN=1,NSET                                                          C1311000
      N = SET(NSET+1-NN)                                                       C1311100
      IFLX = 0                                                                 C1311200
```

```
      IF(CTAIN(N,STI,NSTI)) GO TO 19                                    C1311300
      IFLX = 1                                                          C1311400
   19 CONTINUE                                                          C1311500
C     IF IFLX=C MODAL VECTORS AND TENSORS FIXED IN                      C1311600
C      BODY N MUST BE TRANSFORMED                                       C1311700
C     COMPUTE DEFORMED CM VECTOR AND INERTIA TENSOR BODY N FIXED FRAME  C1311800
      IF(.NOT.LEQU) GO TO 1000                                          C1311900
      PRINT 226,N                                                       C1312000
      PRINT 205                                                         C1312100
 1000 CONTINUE                                                          C1312200
      DO 17 I=1,3                                                       C1312300
      CA(I,N) = CAC(I,N)                                                C1312400
      DO 17 J=1,3                                                       C1312500
   17 XI(I,J,N) = XIO(I,J,N)                                            C1312600
      IF(.NOT.LEQU) GO TO 1001                                          C1312700
      PRINT 220, N,N                                                    C1312800
      PRINT 221, N,N                                                    C1312900
      PRINT 205                                                         C1313000
 1001 CONTINUE                                                          C1313100
      SFXMN = SFXM(N)                                                   C1313200
      DO 18 M=1,SFXMN                                                   C1313300
      MN = MN+1                                                         C1313400
      CALL SCLV(THA(NFER+MN),FLA(1,MN),TEM)                             C1313500
      CALL VECADD(CA(1,N),TEM,CA(1,N))                                  C1313600
      CALL SCLD(THA(NFER+MN),FLE(1,1,MN),TEM1)                          C1313700
      CALL DYADD(XI(1,1,N),TEM1,XI(1,1,N))                              C1313800
      IF(.NOT.LEQU) GO TO 1002                                          C1313900
      NFXM = NFER+MN                                                    C1314000
      PRINT 216, N,N,NFXM,MN                                            C1314100
      PRINT 217, N,N,NFXM,MN                                            C1314200
      PRINT 205                                                         C1314300
 1002 CONTINUE                                                          C1314400
      IF(IFLX.EQ.1) GO TO 20                                            C1314500
      CALL VECTRN(FLA(1,MN),XMC(1,1,N),FLAC(1,MN))                      C1314600
      CALL VECTRN(FLQ(1,MN),XMC(1,1,N),FLQC(1,MN))                      C1314700
      IF(.NOT.LEQU) GO TO 1003                                          C1314800
      PRINT 218, MN,N,MN,(FLAC(I,MN),I=1,3)                             C1314900
      PRINT 219, MN,N,MN,(FLQC(I,MN),I=1,3)                             C1315000
      PRINT 205                                                         C1315100
 1003 CONTINUE                                                          C1315200
   20 CONTINUE                                                          C1315300
   18 CONTINUE                                                          C1315400
      IF(.NOT.LEQU) GO TO 1004                                          C1315500
      PRINT 223, N,(CA(I,N),I=1,3)                                      C1315600
      PRINT 205                                                         C1315700
      PRINT 224,   (XI(1,I,N),I=1,3)                                    C1315800
      PRINT 225, N,(XI(2,I,N),I=1,3)                                    C1315900
      PRINT 224,   (XI(3,I,N),I=1,3)                                    C1316000
 1004 CONTINUE                                                          C1316100
   16 CONTINUE                                                          C1316200
 5000 CONTINUE                                                          C1316300
   15 CONTINUE                                                          C1316400
C                                                                       C1316500
C                                                                       C1316600
C     COMPUTE CENTER OF MASS VECTORS                                    C1316700
      CALL UNPAC(SET,NSET,SVA)                                          C1316800
      IF(NSET.EQ.0) GO TO 5034                                          C1316900
      DO 1  J=1,NSET                                                    C1317000
      K = SET(J)                                                        C1317100
      IF(.NOT.RBLO(K)) GO TO 1                                          C1317200
```

```
      CALL VECTRN (CA(1,K),XMC(1,1,K),CAC(1,K))                         C1317300
      IF(LEQU)PRINT 201, K,K,K,(CAC(I,K),I=1,3)                         C1317400
    1 CONTINUE                                                          C1317500
 5034 CONTINUE                                                          C1317600
      IF(LEQU)PRINT 202                                                 C1317700
C                                                                       C1317800
C     . INERTIA TENSORS                                                 C1317900
      CALL UNPAC(SET,NSET,SVI)                                          C1318000
      IF(NSET.EQ.0) GO TO 5035                                          C1318100
      DO 10  J=1,NSET                                                   C1318200
      K = SET(J)                                                        C1318300
      CALL TENTRN (XI(1,1,K),XMC(1,1,K),XIC(1,1,K))                     C1318400
      IF(LEQU)PRINT 203,          (XIC(1,I,K),I=1,3)                    C1318500
      IF(LEQU)PRINT 204, K,K,K,K,(XIC(2,I,K),I=1,3)                     C1318600
      IF(LEQU)PRINT 203,          (XIC(3,I,K),I=1,3)                    C1318700
      IF(LEQU)PRINT 205                                                 C1318800
   10 CONTINUE                                                          C1318900
 5035 CONTINUE                                                          C1319000
      IF(LEQU)PRINT 202                                                 C1319100
C                                                                       C1319200
C       HINGE VECTORS                                                   C1319300
C     COMPUT INERTIAL POSITION OF CENTER OF MASS OF BODY 1              C1319400
      DO 14  I=1,3                                                      C1319500
   14 CB(I,1) = CBN(I)                                                  C1319600
      IF(LEQU)PRINT 214,(CB(I,1),I=1,3)                                 C1319700
      IF(SCF(NB1).EQ.0) GO TO 5042                                      C1319800
      ITERM1=SCF(NB1)                                                   C1319900
      ITERM2=ITERM1+2-PCCN(NB1)                                         C1320000
      DO 13 M=ITERM1,ITERM2                                             C1320100
      CALL SCLV(THA(M),QF(1,M),TEM)                                     C1320200
      CALL VECADD(CB(1,1),TEM,CB(1,1))                                  C1320300
      IF(LEQU)PRINT 215,M,M                                             C1320400
   13 CONTINUE                                                          C1320500
 5042 CONTINUE                                                          C1320600
      CALL UNPAC(SET,NSET,SVB)                                          C1320700
      IF(NSET.EQ.0) GO TO 5036                                          C1320800
      DO 2  J=1,NSET                                                    C1320900
      K = SET(J)                                                        C1321000
      JJ = JCCN(K)                                                      C1321100
      CALL VECTRN (CB(1,K),XMC(1,1,JJ),CBC(1,K))                        C1321200
      IF(LEQU)PRINT 206, K,JJ,K,(CBC(I,K),I=1,3)                        C1321300
    2 CONTINUE                                                          C1321400
 5036 CONTINUE                                                          C1321500
      IF(LEQU)PRINT 202                                                 C1321600
C                                                                       C1321700
C       FREE COORDINATE VECTORS                                         C1321800
      IF(NSVQ.EQ.0) GO TO 5037                                          C1321900
      DO 3  J=1,NSVQ                                                    C1322000
      M = SVQ(J)                                                        C1322100
      IF(FCCN(M).GE.0) GO TO 4                                          C1322200
      CALL VECTRN (QF(1,M+1),XMC(1,1,FCCN(M+1)),QFC(1,M+1))             C1322300
      CALL VECROS (QFC(1,M+1),QFC(1,M-1),QFC(1,M))                      C1322400
      CALL VECNRM (QFC(1,M))                                            C1322500
      IF(LEQU)M1 = M-1                                                  C1322600
      IF(LEQU)M2 = M+1                                                  C1322700
      IF(LEQU)PRINT 208, M,M2,M1,(QFC(I,M),I=1,3)                       C1322800
      GO TO 3                                                           C1322900
    4 CALL VECTRN (QF(1,M),XMC(1,1,FCCN(M)),QFC(1,M))                   C1323000
      IF(LEQU)PRINT 207, M,FCCN(M),M,(QFC(I,M),I=1,3)                   C1323100
    3 CONTINUE                                                          C1323200
```

```
 5037 CONTINUE                                                          C1323300
      IF(LEQU)PRINT 202                                                 C1323400
C                                                                       C1323500
C       LOCKED COORDINATE VECTORS                                       C1323600
      IF(NSVP.EQ.0) GO TO 5038                                          C1323700
      DO 5  J=1,NSVP                                                    C1323800
      L = SVP(J)                                                        C1323900
      IF(LCON(L).GE.0) GO TO 6                                          C1324000
      M = -LCON(L)                                                      C1324100
      CALL VECROS (QFC(1,M),QFC(1,M+1),QLC(1,L))                        C1324200
      CALL VECNRM (QLC(1,L))                                            C1324300
      IF(LEQU)M1 = M+1                                                  C1324400
      IF(LEQU)PRINT 209, L,M,M1,(QLC(I,L),I=1,3)                        C1324500
      GO TO 5                                                           C1324600
    6 CALL VECTRN (QL(1,L),XMC(1,1,LCON(L)),QLC(1,L))                   C1324700
      IF(LEQU)PRINT 210, L,LCON(L),L,(QLC(I,L),I=1,3)                   C1324800
    5 CONTINUE                                                          C1324900
 5038 CONTINUE                                                          C1325000
      IF(LEQU)PRINT 202                                                 C1325100
C                                                                       C1325200
C       POINT MASS POSITION                                             C1325300
      CALL UNPAC(SET,NSET,SL)                                           C1325400
      IF(NSET.EQ.0) GO TO 5039                                          C1325500
      DO 9  KK=1,NSET                                                   C1325600
      K = SET(KK)                                                       C1325700
      DO 11  I=1,3                                                      C1325800
   11 TEM(I) = CA(I,K)                                                  C1325900
      IN1=SOF(K)                                                        C1326000
      IT1=IN1+2-PCON(K)                                                 C1326100
      DO 12 M=IN1,IT1                                                   C1326200
      IF(LEQU)PRINT 212, K,K,M,M                                        C1326300
      DO 12  I=1,3                                                      C1326400
   12 TEM(I) = TEM(I) + THA(M)*QF(I,M)                                  C1326500
      CALL VECTRN (TEM,XMC(1,1,K),CAC(1,K))                             C1326600
      IF(LEQU)PRINT 213, K,K,K,(CAC(I,K),I=1,3)                         C1326700
    9 CONTINUE                                                          C1326800
 5039 CONTINUE                                                          C1326900
C                                                                       C1327000
C       ANGULAR MOMENTUM WHEEL                                          C1327100
      CALL UNPAC(SET,NSET,SVM)                                          C1327200
      IF(NSET.EQ.0) GO TO 5040                                          C1327300
      DO 7  JJ=1,NSET                                                   C1327400
      J = SET(JJ)                                                       C1327500
      CALL VECTRN (HM(1,J),XMC(1,1,MO(J)),HMC(1,J))                     C1327600
    7 CONTINUE                                                          C1327700
 5040 CONTINUE                                                          C1327800
      IF(.NOT. LEQU) RETURN                                             C1327900
      IF(NSET.EQ.0) GO TO 5041                                          C1328000
      DO 8  JJ=1,NSET                                                   C1328100
      J = SET(JJ)                                                       C1328200
      PRINT 211, J,MO(J),J,(HMC(I,J),I=1,3)                             C1328300
    8 CONTINUE                                                          C1328400
 5041 CONTINUE                                                          C1328500
      PRINT 202                                                         C1328600
C                                                                       C1328700
C                                                                       C1328800
  200 FORMAT ('1  SUBROUTINE TRANVD ENTERED ',2(/))                     C1328900
  201 FORMAT ('  CAC(',I2,') = XMC(',I2,') * CA(',I2,') = ',3D17.8)     C1329000
  202 FORMAT(3(/))                                                      C1329100
  203 FORMAT (44X,3D17.8)                                               C1329200
```

```
 204 FORMAT ('  XIC(',I2,') = XMC(',I2,') * XI(',I2,') * XMC(',I2,')**T C1329300
    * = ',3D17.8)                                                      C1329400
 205 FORMAT (' ')                                                      C1329500
 206 FORMAT ('  CBC(',I2,') = XMC(',I2,') * CB(',I2,') = ',3D17.8)     C1329600
 207 FORMAT ('  QFC(',I2,') = XMC(',I2,') * QF(',I2,')  = ',3D17.8)    C1329700
 208 FORMAT ('  QFC(',I2,') = NORM(QFC(',I2,') X QFC(',I2,')) = ',3D17.C1329800
    *E)                                                                C1329900
 209 FORMAT ('  QLC(',I2,') = NORM(QFC(',I2,') X QFC(',I2,')) = ',3D17.C1330000
    *E)                                                                C1330010
 210 FORMAT ('  QLC(',I2,') = XMC(',I2,') * QL(',I2,')  = ',3D17.8)    C1330200
 211 FORMAT ('  HMC(',I2,') = XMC(',I2,') * HM(',I2,') = ',12X,3D17.E) C1330300
 212 FORMAT ('  CA(',I2,') = CA(',I2,') + THA(',I2,')*QF(',I2,') ')    C1330400
 213 FORMAT (35X,' CAC(',I2,') = XMC(',I2,')*CA(',I2,') = ',3D17.E,/)   C1330500
 214 FORMAT ('  CB( 1) = ',3D17.8)                                      C1330600
 215 FORMAT ('  CB( 1) = CB( 1) + THA(',I2,')*QF(',I2,') ')             C1330700
 216 FORMAT (30X,'  CA(',I2,') = CA(',I2,') + THA(',I2,')*FLA(',I2,')')C1330800
 217 FORMAT (30X,'  XI(',I2,') = XI(',I2,') + THA(',I2,')*FLE(',I2,')')C1330900
 218 FORMAT      (' FLAC(',I2,') = XMC(',I2,')*FLA(',I2,') = ',3D12.5)  C1331000
 219 FORMAT ('  FLQC(',I2,') = XMC(',I2,')*FLQ(',I2,') = ',3D12.5)     C1331100
 220 FORMAT (30X,'  CA(',I2,') = CAJ(',I2,') ')                        C1331200
 221 FORMAT (30X,'  XI(',I2,') = XIJ(',I2,') ')                        C1331300
 223 FORMAT (30X,'  CA(',I2,') =',3D12.5)                              C1331400
 224 FORMAT (40X,3D12.5)                                               C1331500
 225 FORMAT (30X,'  XI(',I2,') =',3D12.5)                              C1331600
 226 FORMAT (//,25X,'  FLEXIELE BODY PARAMETERS FOR BODY',I3)          C1331700
     RETURN                                                            C1331800
     END                                                               C1331900
```

```
C                                                                      C1400000
      SUBROUTINE RATE                                                  C1400100
C     USED TO COMPUTE ALL LINER AND ANGULAR VELOCITY VECTORS REQUIRED  C1400200
C                                                                      C1400300
C       RCMC(I,K) = FOR BODY K RIGID- ANGULAR VELOCITY OF BODY K FIXED C1400400
C                   AXES RELATIVE TO BODY JCON(K) FIXED AXES;          C1400500
C                   FOR BODY K POINT MASS- LINEAR VELOCITY OF POINT MASC1400600
C                   REALATIVE TO AXES FIXED AT HINGE POINT K-1         C1400700
C       CCMC(I,K) = FOR BODY K RIGID- ANGULAR VELOCITY OF BODY K FIXED C1400800
C                   AXES RELATIVE TO FRAME OF COMPUTATION              C1400900
C                   FOR BODY K POINT MASS- EQUALS RCMC(I,K)            C1401000
C       FCMC(I,K) = FOR BODY K RIGID- ANGULAR VELOCITY OF BODY K FIXED C1401100
C                   AXES RELATIVE TO INERTIAL FRAME                    C1401200
C                   FOR BODY K POINT MASS- EQUALS RCMC(I,K)            C1401300
C       DCMC(I,K) = SUM OF INERTIAL DERIVATIVES(FIRST) OF FREE VECTORS C1401400
C                   AT HINGE POINT K-1                                 C1401500
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C1401600
C                                                                      C1401700
C                                                                      C1401800
      LOGICAL FG1, FG2, FG3, FG4, FG5, INERF, RBLD, LEQU, LINIT(1)     C1401900
     *,LOGICAL      ERUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,      C1402000
     *          (EE) ,LTRANV , LRATE  , LXDY   , LETA  , LTORQU ,      C1402100
     *               LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C1402200
C                                                                      C1402300
C                                                                      C1402400
      INTEGER                                                          C1402500
     * A*CRK , CT1   , CT2   , CT3   , CT4   , CT5   , FCCN  , PCCN  , C1402600
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    , C1402700
```

```
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,  C1402800
     * SMA    , SMCDUM, SMC    , SMV    , SOK   , SPIDUM, SPI    , SQF    ,  C1402900
     * SGL    , SR     , SSCN   , SSIX   , SVA   , SVB    , SVD    , SVI    ,  C1403000
     * SVM    , SVP    , SVQ    , SXM    , SXT   , TORQ   , SMAL   , SEU    ,  C1403100
     * SC     , SCG    , NFLXB  , SFLX   , SFXM  , NMODS  , SFCC   , SCC    ,  C1403200
     * IINIT(1)        , IZINIT(1)     , SD                                    C1403300
C                                                                              C1403400
C                                                                              C1403500
C                                                                              C1403600
      REAL*8                                                                   C1403700
     * ANGC  (33)     , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,       C1403800
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                       C1403900
     * THACD (33)     , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)             C1404000
C                                                                              C1404100
C                                                                              C1404200
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQDIV , LTRAN  ,            C1404300
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,            C1404400
     *                LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ               C1404500
C                                                                              C1404600
C                                                                              C1404700
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERR, RBLO(10)                  C1404800
C                                                                              C1404900
C                                                                              C1405000
      COMMON /INTG/ AWORK(200) ,                                               C1405100
     * CT1            , CT2            , CT3            , CT4            ,     C1405200
     * CT5            , FCON  (33)     , JCON  (10)     , LCON  (22)     ,     C1405300
     * MC     (10)    , NBI            , NBUD           , NCTC           ,     C1405400
     * NFER           , NFKC           , NFRC           , NLOR           ,     C1405500
     * NMV            , NMQ            , NMJA           , NSVP           ,     C1405600
     * NSVG           , FCON  (11)     , SD             , SFR   (33)     ,     C1405700
     * SG             , SI    (55)     , SIG            , SL             ,     C1405800
     * SLK    (33)    , SMA   (10)     , SOK   (11)     , SQF   (11)     ,     C1405900
     * SGL    (11)    , SMV            , SR             , SSCN           ,     C1406000
     * SSIX           , SVA            , SVB            , SVD            ,     C1406100
     * SVI            , SVM            , SVP   (22)     , SVQ   (33)     ,     C1406200
     * SXM    (3,10)  , SXT            , TORQ  (97)     , SMAL           ,     C1406300
     * SEU            , NTQ            , SC    (33)     , SCG            ,     C1406400
     * NFLXB          , SFLX           , SFXM  (10)     , NMODS          ,     C1406500
     * SFCC           , SCC   (10)                                             C1406600
C                                                                              C1406700
C                                                                              C1406800
      COMMON /INTGZ/                                                           C1406900
     * SCNDUM         , SCN   (9)      , SCRDUM         , SCR   (9)      ,     C1407000
     * SFKDUM         , SFK   (9)      , SIXDUM         , SIX   (9)      ,     C1407100
     * SKDUM          , SK    (9)      , SPIDUM         , SPI   (9)      ,     C1407200
     * SMCDUM         , SMC   (9)                                              C1407300
C                                                                              C1407400
C                                                                              C1407500
      COMMON /REAL/                                                            C1407600
     * CA     (3,10)  , CAC   (3,10)  , CLM   (10)     , COMC  (3,11)  ,      C1407700
     * DCMC   (3,11)  , ETC   (3,11)  , ETM   (33)     , FLMC  (3,11)  ,      C1407800
     * GAM    (3,66)  , H              , HM    (3,10)   , HMC   (3,10)  ,      C1407900
     * HMCM   (10)    , PHI   (3,11)  , PLM   (10)     , QF    (3,33)  ,      C1408000
     * QFC    (3,33)  , QL    (3,22)  , QLC   (3,22)   , ROMC  (3,11)  ,      C1408100
     * RTD            , THA   (33)    , THAD  (33)     , THADW (10)    ,      C1408200
     * THAW   (10)    , XDIC  (3,3,66), XI    (3,3,10) , XIC   (3,3,10),      C1408300
     * XMAS   (10)    , XMN   (33,33) , XMT   (3,3,10) , TLG   (33)    ,      C1408400
     * FLA    (3,20)  , FLB   (3,20)  , FLC   (3,20)   , FLD   (3,3,20),      C1408500
     * FLJ    (3,3,20), CAD   (3,10)  , XIO   (3,3,10) , FLIRC (3,10)  ,      C1408600
     * FLCRC  (3,10)  , FLAC  (3,20)  , FLQC  (3,20)   ,                       C1408700
```

```
     * FLCM  (20)     , ZETA  (20)     , FCF   (3,3,40), FCK   (3,40)  ,  C1408800
     * TIMEND                                                             C1408900
C                                                                        C1409000
C                                                                        C1409100
      COMMON /REALZ/                                                     C1409200
     * CBDUM (1,3)    , CB    (3,10)   , CBCDUM(1,3)   , CBC   (3,10)  ,  C1409300
     * XMCDUM(1,1,9) , XMC   (3,3,10), CBN(3)                           C1409400
C                                                                        C1409500
C                                                                        C1409600
      EQUIVALENCE (ETM(1),THACC(1))          ,(XMN(1,1),ANGD(1))      ,  C1409700
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,  C1409800
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,  C1409900
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),   C1410000
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                           C1410100
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,  C1410200
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,  C1410300
     *            (SCNDUM,IZINIT(1))                                     C1410400
      DIMENSION X1(3),X2(3)                                              C1410500
      DIMENSION TEM1(33),TEM2(33)                                        C1410600
      REAL*8 EFCMC(33), ECCMC(33), ERCMC(33), EDCMC(33)                  C1410700
      INTEGER  SET(10)                                                   C1410800
      LOGICAL CTAIN                                                      C1410900
C                                                                        C1411000
C     USE SINGLE DIMENSION ARRAYS WHEN ADVANTAGEOUS TO INCREASE          C1411100
C      COMPUTATION SPEED                                                 C1411200
      EQUIVALENCE(FCMC(1,1),TEM1(1)), (CCMC(1,1),TEM2(1))                C1411300
      EQUIVALENCE (EFCMC(1),FCMC(1,1)), (ECCMC(1),CCMC(1,1)),            C1411400
     *            (ERCMC(1),RCMC(1,1)), (EDCMC(1),DCMC(1,1))             C1411500
C                                                                        C1411600
      EQUIVALENCE (LRATE,LEQU)                                           C1411700
C                                                                        C1411800
C                                                                        C1411900
      IF(LEQU)PRINT 200                                                  C1412000
      IF(.NOT.LEQU) GO TO 1000                                           C1412100
      IF(INERF) GO TO 33                                                 C1412200
      PRINT 201                                                          C1412300
      GO TO 34                                                           C1412400
   33 PRINT 202                                                          C1412500
   34 CONTINUE                                                           C1412600
C     EVALUTE RELATIVE VELOCITY VECTORS                                  C1412700
 1000 CONTINUE                                                           C1412800
      M = 1                                                              C1412900
      DO 1   K= 1,NB1                                                    C1413000
      IF(K.EQ.1) GO TO 2                                                 C1413100
      M = M+3-PCON(K-1)                                                  C1413200
    2 CONTINUE                                                           C1413300
      IK1 = 3*K-3                                                        C1413400
      DO 31  I=1,3                                                       C1413500
      IK = IK1+I                                                         C1413600
   31 ERCMC(IK) = 0.D0                                                   C1413700
      IF(LEQU)PRINT 221, K                                               C1413800
      IK = IK1 + 1                                                       C1413900
      ITERM = M+2-PCON(K)                                                C1414000
      IF(M.GT.ITERM)GO TO 1                                              C1414100
      DO 3 MM=M,ITERM                                                    C1414200
      CALL SCLV(THAD(MM),QFC(1,MM),X1)                                   C1414300
      CALL VECADD(X1,ERCMC(IK),ERCMC(IK))                                C1414400
      IF(LEQU)PRINT 205, K,K,MM,MM                                       C1414500
    3 CONTINUE                                                           C1414600
    1 CONTINUE                                                           C1414700
```

```
      IF(.NOT.LEQU) GO TO 1001                                          C1414800
      PRINT 214                                                         C1414900
      DO 25 K=1,NB1                                                     C1415000
   25 PRINT 215, K,(ROMC(J,K),J=1,3)                                    C1415100
      PRINT 214                                                         C1415200
 1001 CONTINUE                                                          C1415300
C                                                                       C1415400
C       EVALUATE ANGULAR VELOCITY BODY K TO INERTIAL AND COMPUTING FRAME C1415500
C                                                                       C1415600
C     TAKE CARE OF RIGID BODY ANGULAR RATES                             C1415700
      CALL UNPAC(SET,NSET,SR)                                           C1415800
      DO 7  I=1,3                                                       C1415900
    7 ROMC(I,1) = ROMC(I,1)                                             C1416000
      IF(LEQU)PRINT 219                                                 C1416100
      NSETM1 = NSET - 1                                                 C1416200
      IF(NSETM1.EQ.0) GO TO 5042                                        C1416300
      DO 8 JJ= 1,NSETM1                                                 C1416400
      J = NSETM1 - (JJ-1)                                               C1416500
      K = SET(J)                                                        C1416600
      CALL VECADD(ROMC(1,JCON(K)),ROMC(1,K),ROMC(1,K))                  C1416700
      IF(LEQU)JK = JCON(K)                                              C1416800
      IF(LEQU)PRINT 209, K,JK,K                                         C1416900
    8 CONTINUE                                                          C1417000
 5042 CONTINUE                                                          C1417100
      IF(INERF) GO TO 11                                                C1417200
      DO 9  I=1,3                                                       C1417300
    9 ECOMC(I) = 0.DO                                                   C1417400
      IF(LEQU)PRINT 214                                                 C1417500
      IF(LEQU)PRINT 222                                                 C1417600
      NSETM1 = NSET - 1                                                 C1417700
      IF(NSETM1.EQ.0) GO TO 5043                                        C1417800
      DO 10 JJ = 1 , NSETM1                                             C1417900
      J = NSETM1 - ( JJ - 1 )                                           C1418000
      K = SET(J)                                                        C1418100
      CALL VECADD(COMC(1,JCON(K)),ROMC(1,K),COMC(1,K))                  C1418200
      IF(LEQU)JK = JCON(K)                                              C1418300
      IF(LEQU)PRINT 210, K,JK,K                                         C1418400
   10 CONTINUE                                                          C1418500
 5043 CONTINUE                                                          C1418600
      IF(LEQU)PRINT 214                                                 C1418700
      GO TO 12                                                          C1418800
C                                                                       C1418900
C     COMC = ROMC  VIA EQUIVALENCE FOR INERTIAL COMPUTING FRAME         C1419000
   11 ITERM = 3 * NBOD                                                  C1419100
      DO 13 I = 1, ITERM                                                C1419200
   13 TEM2(I) = TEM1(I)                                                 C1419300
      IF(.NOT.LEQU) GO TO 1002                                          C1419400
      PRINT 214                                                         C1419500
      DO 26  K=1,NBOD                                                   C1419600
   26 PRINT 220, K,K                                                    C1419700
      PRINT 214                                                         C1419800
 1002 CONTINUE                                                          C1419900
   12 CONTINUE                                                          C1420000
C                                                                       C1420100
C     TAKE CARE OF LINEAR OSCILLATORS AND CENTER OF MASS MOTION         C1420200
      CALL UNPAC(SET,NSET,SL)                                           C1420300
      IF(NSET.EQ.0) GO TO 5000                                          C1420400
      DO 23 J=1,NSET                                                    C1420500
      K = SET(J)                                                        C1420600
      IF(LEQU) PRINT 223, K,JCON(K)                                     C1420700
```

```
      IF(LEQU) PRINT 224, K,JCON(K)                                     C1420800
      DO 6  I=1,3                                                       C1420900
      FOMC(I,K) = FOMC(I,JCON(K))                                       C1421000
    6 COMC(I,K) = COMC(I,JCON(K))                                       C1421100
   23 CONTINUE                                                          C1421200
 5000 CONTINUE                                                          C1421300
      FOMC(I,NB1) = ROMC(I,NB1)                                         C1421400
   15 COMC(I,NB1) = ROMC(I,NB1)                                         C1421500
      IF(LEQU) PRINT 206, NB1,NB1,NB1                                   C1421600
C                                                                       C1421700
      IF(.NOT.LEQU) GO TO 1003                                          C1421800
      DO 28  K=1,NB1                                                    C1421900
   28 PRINT 217, K,(COMC(J,K),J=1,3),K,(FOMC(J,K),J=1,3)                C1422000
      PRINT 214                                                         C1422100
 1003 CONTINUE                                                          C1422200
C                                                                       C1422300
C     EVALUTE COMPONENTS OF ACCELERATION ASSOCIATED WITH COORDINATE     C1422400
C     FRAME ROTATION                                                    C1422500
C                                                                       C1422600
      IF(CT4.NE.1) GO TO 15                                             C1422700
      DO 15  K=1,NB1                                                    C1422800
      IK = 3*K-3                                                        C1422900
      DO 14  I=1,3                                                      C1423000
      IK = IK+1                                                         C1423100
   14 EDOMC(IK) = 0.D0                                                  C1423200
      IF(LEQU)PRINT 212, K                                              C1423300
   15 CONTINUE                                                          C1423400
C                                                                       C1423500
C     CYCLE THROUGH ALL BODIES                                          C1423600
      M = 1                                                             C1423700
      DO 16  K=1,NBOD                                                   C1423800
      IK1 = 3*K-3                                                       C1423900
      IF(LEQU)PRINT 203                                                 C1424000
      IF(K.EQ.1) GO TO 17                                               C1424100
      M = M+3-PCON(K-1)                                                 C1424200
   17 IF(.NOT.RBLO(K)) GO TO 16                                         C1424300
      DO 21  I=1,3                                                      C1424400
      IK = IK1+1                                                        C1424500
   21 EDOMC(IK) = 0.D0                                                  C1424600
      IF(LEQU)PRINT 212, K                                              C1424700
C     TAKE CARE OF COMPONENT DUE TO ROTATION OF FRAME K RELATIVE TO INER C1424800
C     CYCLE THROUGH FREE COORDINATE VECTORS AT HINGE POINT K-1           C1424900
      IK = IK1 + 1                                                      C1425000
      MMM = M + 2 - PCON(K)                                             C1425100
      IF(M.GT.MMM) GO TO 5044                                           C1425200
      DO 16 MM= M,MMM                                                   C1425300
      MF = PCON(MM)                                                     C1425400
      IF(MF.LT.0) GO TO 18                                              C1425500
      IF(MF.EQ.0) GO TO 16                                              C1425600
      CALL VECROS (FOMC(1,MF),GFC(1,MM),X1)                             C1425700
      CALL SCLV(THAD(MM),X1,X1)                                         C1425800
      CALL VECADD(EDOMC(IK),X1,EDOMC(IK))                               C1425900
      IF(LEQU)PRINT 213, K,K,MM,MF,MM                                   C1426000
      GO TO 16                                                          C1426100
   18 MF = PCON(MM-1)                                                   C1426200
      M1 = MM-1                                                         C1426300
      CALL SCLV(THAD(M1),GFC(1,M1),X1)                                  C1426400
      IF(MF.EQ.0) GO TO 4                                               C1426500
      CALL VECADD(FOMC(1,MF),X1,X1)                                     C1426600
    4 CONTINUE                                                          C1426700
```

```
      CALL VECROS (X1,QFC(1,MM),X2)                                     C1426800
      CALL SCLV(THAD(MM),X2,X2)                                         C1426900
      CALL VECADD(EDCMC(IK),X2,EDCMC(IK))                               C1427000
      IF(MF.EQ.0) GO TO 5                                               C1427100
      IF(LEQU)PRINT 216, K,K,MM,MF,M1,M1,MM                             C1427200
      GO TO 16                                                          C1427300
    5 CONTINUE                                                          C1427400
      IF(LEQU)PRINT 204, K,K,MM,M1,M1,MM                                C1427500
 5044 CONTINUE                                                          C1427600
   16 CONTINUE                                                          C1427700
      IF(.NOT.LEQU) RETURN                                              C1427800
      PRINT 214                                                         C1427900
      DO 25  M=1,NB1                                                    C1428000
   25 PRINT 227, M,(DOMC(J,M),J=1,3)                                    C1428100
C                                                                       C1428200
C                                                                       C1428300
  200 FORMAT ('   SUBROUTINE RATE ENTERED ')                            C1428400
  201 FORMAT (/,'  COMPUTING FRAME IS BODY I ',/)                       C1428500
  202 FORMAT (/,'  COMPUTING FRAME IS INERTIALLY FIXED ',/)              C1428600
  203 FORMAT ('  ')                                                     C1428700
  204 FORMAT ('  DOMC(',I2,') = DCMC(',I2,') + THAD(',I2,')*THAD(',I2,')C1428800
     **QFC(',I2,') X QFC(',I2,') ')                                     C1428900
  205 FORMAT ('  RCMC(',I2,') = RCMC(',I2,') + THAD(',I2,')*QFC(',I2,') C1429000
     *')                                                                C1429100
  208 FORMAT ('  FCMC(',I2,') = CCMC(',I2,') = RCMC(',I2,') ')         C1429200
  209 FORMAT ('  FCMC(',I2,') = FCMC(',I2,') + RCMC(',I2,') ')         C1429300
  210 FORMAT ('  CCMC(',I2,') = CCMC(',I2,') + RCMC(',I2,')  ')        C1429400
  212 FORMAT ('  DOMC(',I2,') = 0  ')                                   C1429500
  213 FORMAT ('  DOMC(',I2,') = DCMC(',I2,') + THAD(',I2,')*(FCMC(',I2,'C1429600
     *) X QFC(',I2,'))')                                                C1429700
  214 FORMAT(2(/))                                                      C1429800
  215 FORMAT ('  RCMC(',I2,') = ',3D17.8)                               C1429900
  216 FORMAT ('  DCMC(',I2,') = DCMC(',I2,') + THAD(',I2,')*((FCMC(',I2,C1430000
     *') + THAD(',I2,')*QFC(',I2,')) X QFC(',I2,'))')                   C1430100
  217 FORMAT ('  CCMC(',I2,') = ',3D17.8,'  FOMC(',I2,') = ',3D17.8)    C1430200
  219 FORMAT ('  FCMC( 1) = RCMC( 1) ')                                 C1430300
  220 FORMAT ('  CCMC(',I2,') = FCMC(',I2,') ')                         C1430400
  221 FORMAT ('  RCMC(',I2,') = 0 ')                                    C1430500
  222 FORMAT ('  CCMC( 1) = C ')                                        C1430600
  223 FORMAT ('  FOMC(',I2,') = FCMC(',I2,') ')                         C1430700
  224 FORMAT ('  CCMC(',I2,') = CCMC(',I2,') ')                         C1430800
  227 FORMAT ('  DOMC(',I2,') = ',3D17.8)                               C1430900
      RETURN                                                            C1431000
      END                                                               C1431100
```

```
C                                                                       C1500000
      SUBROUTINE XDY                                                    C1500100
C       COMPUTES VALUES OF VECTORS AND DYADS USED TO DEFINE SYSTEM INERTC1500200
C         TENSOR MATRIX                                                 C1500300
C                                                                       C1500400
C     LET:                                                              C1500500
C          GAM(I,KL) = COMPONENTS OF VECTOR FROM HINGE POINT K-1 TO THE C1500600
C                      CENTER OF MASS OF BODY LAMBA WHERE               C1500700
C                              KL = KTO(NBOD+1,K-1,LAMBA)               C1500800
C        XDIC(I,J,KI) = COMPONENTS OF TENSOR OF RANK TWO IN ROW K, COL I C1500900
C                      OF THE SYSTEM MATRIX OF INERTIA TENSORS          C1501000
```

```
C                       WHERE                                                   C1501100
C                                  KI = KT1(NBOD+1,K,I)                         C1501200
C     NOTE:                                                                     C1501300
C          THE SYSTEM MATRIX OF INERTIA TENSORS IS SYMMETRIC THUS ONLY          C1501400
C          LOWER TRIANGULAR PORTION COMPUTED                                    C1501500
      IMPLICIT REAL*8(A-H,O-Z,$)                                                C1501600
C                                                                               C1501700
C                                                                               C1501800
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLD, LEQU, LINIT(1)             C1501900
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,             C1502000
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,             C1502100
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ                C1502200
C                                                                               C1502300
C                                                                               C1502400
      INTEGER                                                                   C1502500
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCCN   , PCCN   ,   C1502600
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,   C1502700
     * SI     , SIG    , SIXDUM, SIX    , SKDUM  , SK     , SL     , SLK    ,   C1502800
     * SMA    , SMCDUM, SMC    , SMV    , SOK    , SPIDUM, SPI    , SQF    ,   C1502900
     * SGL    , SR     , SSCN   , SSIX   , SVA    , SVB    , SVD    , SVI    ,   C1503000
     * SVM    , SVP    , SVQ    , SXM    , SXT    , TORQ   , SMAL   , SEU    ,   C1503100
     * SC     , SCG    , NFLXB  , SFLX   , SFXM   , NMODS  , SFCC   , SCC    ,   C1503200
     * IINIT(1)        , IZINIT(1)       , SD                                   C1503300
C                                                                               C1503400
C                                                                               C1503500
      REAL*8                                                                    C1503600
     * ANGC   (33)     , CNF    (3,10)   , ETIC   (3,10)   , ETMC   (3,10)  ,   C1503700
     * FLQ    (3,20)   , FLE    (3,3,20), FLH    (3,3,20),                     C1503800
     * THACD  (33)     , YMCD   (3,2,11), RINIT  (1)      , RZINIT(1)          C1503900
C                                                                               C1504000
C                                                                               C1504100
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,             C1504200
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,             C1504300
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ                C1504400
C                                                                               C1504500
C                                                                               C1504600
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RELO(10)                   C1504700
C                                                                               C1504800
C                                                                               C1504900
      COMMON /INTG/ AWORK(200) ,                                                C1505000
     * CT1             , CT2             , CT3             , CT4             ,  C1505100
     * CT5             , FCON   (33)     , JCCN   (10)     , LCCN   (22)     ,  C1505200
     * MC     (10)     , NB1             , NBOD            , NCTC            ,  C1505300
     * NFER            , NFKC            , NFRC            , NLOR            ,  C1505400
     * NMV             , NMO             , NMOA            , NSVP            ,  C1505500
     * NSVC            , FCON   (11)     , SD              , SFR    (33)     ,  C1505600
     * SG              , SI     (55)     , SIG             , SL              ,  C1505700
     * SLK    (33)     , SMA    (10)     , SOK    (11)     , SQF    (11)     ,  C1505800
     * SGL    (11)     , SMV             , SR              , SSCN            ,  C1505900
     * SSIX            , SVA             , SVB             , SVD             ,  C1506000
     * SVI             , SVM             , SVP    (22)     , SVQ    (33)     ,  C1506100
     * SXM    (3,10)   , SXT             , TORQ   (97)     , SMAL            ,  C1506200
     * SEU             , NTQ             , SC     (33)     , SCG             ,  C1506300
     * NFLXB           , SFLX            , SFXM   (10)     , NMODS           ,  C1506400
     * SFCC            , SCC    (10)     , SVD             , XIRTAM AITREMI  ,  C1506500
C                                                                               C1506600
C                                                                               C1506700
      COMMON /INTGZ/                                                            C1506800
     * SCNDUM          , SCN    (9)      , SCRDUM          , SCR    (9)      ,  C1506900
     * SFKDUM          , SFK    (9)      , SIXDUM          , SIX    (9)      ,  C1507000
```

```
     * SKDUM          , SK    (9)      , SPIDUM         , SPI   (9)      ,  C1507100
     * SMCDUM         , SMC   (9)                                          C1507200
C                                                                          C1507300
C                                                                          C1507400
      COMMON /REAL/                                                        C1507500
     * CA    (3,10)   , CAC   (3,10)   , CLM   (10)     , CCMC  (3,11)   ,  C1507600
     * DCMC  (3,11)   , ETC   (3,11)   , ETM   (33)     , FCMC  (3,11)   ,  C1507700
     * GAM   (3,66)   , H              , HM    (3,10)   , HMC   (3,10)   ,  C1507800
     * HMDM  (10)     , PHI   (3,11)   , PLM   (10)     , QF    (3,33)   ,  C1507900
     * QFC   (3,33)   , QL    (3,22)   , QLC   (3,22)   , RCMC  (3,11)   ,  C1508000
     * T              ,                  THA   (33)     , THAD  (33)     ,  C1508100
     * THADW (10)     , THAW  (10)     , XDIC  (3,3,66), XI    (3,3,10),  C1508200
     * XIC   (3,3,10), XMAS  (10)     , XMN   (33,33)  , XMT   (3,3,10),  C1508300
     * TLG   (33)     , FLA   (3,20)   , FLB   (3,20)   , FLC   (3,20)   ,  C1508400
     * FLD   (3,3,20), FLJ   (3,3,20), CAU   (3,10)   , XIU   (3,3,10),  C1508500
     * FLIRC (3,10)   , FLCRC (3,10)   , FLAC  (3,20)   , FLQC  (3,20)   ,  C1508600
     * FLCM  (20)     , ZETA  (20)     , FCF   (3,3,40), FCK   (3,40)   ,  C1508700
     * TIMEND                                                              C1508800
C                                                                          C1508900
C                                                                          C1509000
      COMMON /REALZ/                                                       C1509100
     * CEDUM (1,3)    , CB    (3,10)   , CBCDUM(1,3)    , CBC   (3,10)   ,  C1509200
     * XMCDUM(1,1,9)  , XMC   (3,3,10), CBN(3)                             C1509300
C                                                                          C1509400
C                                                                          C1509500
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      ,    C1509600
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,   C1509700
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,   C1509800
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),    C1509900
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                            C1510000
     *            (FG1,LINIT(1))             ,(GA(1,1),RINIT(1))      ,   C1510100
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,   C1510200
     *            (SCNDUM,IZINIT(1))                                       C1510300
C                                                                          C1510400
      INTEGER ST1(10),ST2(10),ST3(10),ST4(10)                              C1510500
      DIMENSION TEM(3),TEM1(3,3),TEM2(3,3),TEM3(3),TEM4(3,3)               C1510600
      EQUIVALENCE (LXDY,LEQU)                                              C1510700
C                                                                          C1510800
C                                                                          C1510900
C                                                                          C1511000
C                                                                          C1511100
C                                                                          C1511200
      IF(LEQU)PRINT 228                                                    C1511300
      IF(CT4.NE.1) GO TO 20                                                C1511400
C     FIRST PASS THROUGH PUT FULL SETS IN DECREASING                       C1511500
C      ORDER OF MAGNITUDE                                                  C1511600
      DO 21  K=1,NBOD                                                      C1511700
      ST3(K) = NB1-K                                                       C1511800
   21 ST4(K) = NB1-K                                                       C1511900
      NST3 = NBOD                                                          C1512000
      NST4 = NBOD                                                          C1512100
      GO TO 22                                                             C1512200
C                                                                          C1512300
C     SXT - THOSE ROWS OF INERTIA MATRIX, OVER TO DIAGONAL, WHICH          C1512400
C            ARE TIME VARYING AND MUST BE RECALCULATED                     C1512500
C                                                                          C1512600
C     SVD - THOSE BODIES IN WHICH BODY FIXED VECTORS AND DYADS ARE         C1512700
C            TIME VARYING IN COMPUTING FRAME                               C1512800
C                                                                          C1512900
   20 CALL UNPAC(ST3,NST3,SXT)                                             C1513000
```

```
      IF(LEQU)PRINT 202, SXT,(ST3(I),I=1,NST3)                          C1513100
      CALL UNPAC(ST4,NST4,SVD)                                          C1513200
      IF(LEQU)PRINT 203, SVD,(ST4(I),I=1,NST4)                          C1513300
C                                                                       C1513400
C     COMPUTE VECTORS FROM HINGE POINT K-1 TO BODY LAMEA CENTER OF MASS  C1513500
C           STORE UPPER TRIANGULAR                                      C1513600
C                                                                       C1513700
C     AFTER FIRST PASS RECOMPUTE ONLY THOSE GAM VECTORS WHICH ARE TIME V C1513800
C                                                                       C1513900
   22 CONTINUE                                                          C1514000
      IF(NST4.EQ.0) GO TO 5045                                          C1514100
      DO 3  LP=1,NST4                                                   C1514200
      L = ST4(LP)                                                       C1514300
      KLL = KTC(NBI,L-1,L)                                              C1514400
      DO 4  I=1,3                                                       C1514500
    4 GAM(I,KLL) = CAC(I,L)                                             C1514600
      IF(LEQU)PRINT 229                                                 C1514700
      IF(LEQU)L1 = L-1                                                  C1514800
      IF(LEQU)PRINT 204, KLL,L1,L,L                                     C1514900
      LL = L                                                            C1515000
    5 K = JCON(LL)                                                      C1515100
      IF(K.EQ.0) GO TO 3                                                C1515200
      KL = KTC(NBI,K-1,L)                                               C1515300
      CALL VECADD(CBC(1,LL),GAM(1,KLL),GAM(1,KL))                       C1515400
      IF(LEQU)K1 = K-1                                                  C1515500
      IF(LEQU)PRINT 205, KL,K1,L,KLL,LL                                 C1515600
      KLL = KL                                                          C1515700
      LL = K                                                            C1515800
      GO TO 5                                                           C1515900
    3 CONTINUE                                                          C1516000
 5045 CONTINUE                                                          C1516100
      IF(.NOT. LEQU) GO TO 1000                                         C1516200
      PRINT 230                                                         C1516300
      DO 23  K=1,NBOD                                                   C1516400
      PRINT 229                                                         C1516500
      DO 23  L=K,NBOD                                                   C1516600
      KL = KTC(NBI,K-1,L)                                               C1516700
      K1 = K-1                                                          C1516800
   23 PRINT 231, K1,L,KL,(GAM(I,KL),I=1,3)                              C1516900
      PRINT 230                                                         C1517000
 1000 CONTINUE                                                          C1517100
C                                                                       C1517200
C     COMPUTE ELEMENTS OF SYSTEM INERTIA TENSOR MATRIX                  C1517300
C                                                                       C1517400
C     AFTER FIRST PASS RECOMPUTE ONLY TIME VARYING ELEMENTS OF XDIC     C1517500
C       1) IF K CONTAINED IN SXT, ELEMENTS OF ROW K FROM COLUMN 1 TO K  C1517600
C          ARE EITHER ZERO OR ASSUMED TIME VARYING, INTERNAL LOGIC, SUM C1517700
C          ON SET SPI(K-1), SKIPS ZERO ELEMENTS                         C1517800
C       2) IF K CONTAINED IN SXT, ELEMENT IN ROW NBOD+1 COLUMN K IS     C1517900
C          ASSUMED TIME VARYING                                         C1518000
C       3) IF K NOT CONTAINED IN SXT ABOVE ELEMENTS ASSUMED CONSTANT AN C1518100
C          EQUAL TO VALUE DEFINED ON FIRST PASS THROUGH                 C1518200
C       4) IF HINGE K-1 IS A RIGID HINGE INERTIA TENSORS IN ROW K NOT,  C1518300
C          NEEDED FOR DYNAMICS HOWEVER THEY ARE NECESSARY FOR CONSTRAIN C1518400
C          TORQUES, NO LOGIC TO AVOID THIS SINCE IT WILL BE HARDLY EVER C1518500
C     SXT STORED IN ST3 ARRAY                                           C1518600
      IF(NST3.EQ.0) GO TO 5046                                          C1518700
      DO 6  KP=1,NST3                                                   C1518800
      IF(LEQU)PRINT 230                                                 C1518900
      K = ST3(KP)                                                       C1519000
```

```
      IN = KTC(NBOD,0,K-1)                                              C1519100
C     SET SI(IN) DEFINES BODY LABELS ON PATH FROM HINGE POINT ZERO TO   C1519200
C       CENTER OF MASS OF BODY K                                        C1519300
      CALL UNPAC(ST1,NST1,SI(IN))                                       C1519400
      IF(RELO(K)) GO TO 7                                               C1519500
C                                                                       C1519600
C     BODY K IS A LINEAR OSCILLATOR                                     C1519700
      IF(LEQU)PRINT 206, K                                              C1519800
      IF(LEQU)PRINT 232, K                                              C1519900
      IF(LEQU)PRINT 207,K,(ST1(JJ),JJ=1,NST1)                           C1520000
      IF(LEQU)PRINT 229                                                 C1520100
C     FILL UP ROW K TO DIAGONAL                                         C1520200
      IF(NST1.EQ.0) GO TO 6                                             C1520300
      DO 8  II=1,NST1                                                   C1520400
      I = ST1(II)                                                       C1520500
      KI = KT1(NB1,K,I)                                                 C1520600
      IF(I.EQ.K) GO TO 5                                                C1520700
      IK = KTC(NB1,I-1,K)                                               C1520800
      CALL SCLV(XMAS(K),GAM(1,IK),TEM)                                  C1520900
      CALL DYCP(TEM,XDIC(1,1,KI))                                       C1521000
      IF(LEQU)II = I-1                                                  C1521100
      IF(LEQU)PRINT 208, KI,K,I,K,IK,K,II,K                             C1521200
      GO TO 8                                                           C1521300
    5 IF(CT4.NE.1) GO TO 8                                              C1521400
      NK = KT1(NB1,NB1,K)                                               C1521500
      DO 10  M=1,3                                                      C1521600
      XDIC(M,M,KI) = XMAS(K)                                            C1521700
   10 XDIC(M,M,NK) = XMAS(K)                                            C1521800
      IF(LEQU)PRINT 210, KI,K,I,K                                       C1521900
      IF(LEQU)PRINT 210, NK,NB1,K,K                                     C1522000
    8 CONTINUE                                                          C1522100
      GO TO 6                                                           C1522200
C                                                                       C1522300
    7 CONTINUE                                                          C1522400
C     BODY K IS A RIGID BODY                                            C1522500
      IF(LEQU)PRINT 211, K                                              C1522600
      IF(LEQU)PRINT 232, K                                              C1522700
      IF(LEQU)PRINT 229                                                 C1522800
      IF(LEQU)PRINT 207,K,(ST1(JJ),JJ=1,NST1)                           C1522900
      DO 11 I=1,3                                                       C1523000
      TEM3(I) = 0                                                       C1523100
      DO 11 J=1,3                                                       C1523200
   11 TEM1(I,J) = 0                                                     C1523300
C                                                                       C1523400
C       SPI(K-1) = SET OF BODIES IN NEST K-1 CONTRIBUTING TO PSUEDO INER C1523500
C                  TENSORS OF NEST K-1                                  C1523600
      CALL UNPAC(ST2,NST2,SPI(K-1))                                     C1523700
      IF(LEQU)PRINT 212, K,K,(ST2(M),M=1,NST2)                          C1523800
      IF(LEQU)PRINT 229                                                 C1523900
      IF(LEQU)PRINT 217                                                 C1524000
      IF(NST2.EQ.0) GO TO 5047                                          C1524100
      DO 12  LL=1,NST2                                                  C1524200
      L = ST2(LL)                                                       C1524300
      KL = KTO(NB1,K-1,L)                                               C1524400
      CALL SCLV(XMAS(L),GAM(1,KL),TEM)                                  C1524500
      CALL VECADD(TEM3,TEM,TEM3)                                        C1524600
      IF(LEQU)K1 = K-1                                                  C1524700
      IF(LEQU)PRINT 213, L,KL,L,K1,L                                    C1524800
      IF(.NOT.RELO(L)) GO TO 12                                         C1524900
      CALL DYADD(TEM1,XIC(1,1,L),TEM1)                                  C1525000
```

```
      IF(LEQU)PRINT 215, L                                              C1525100
   12 CONTINUE                                                          C1525200
 5047 CONTINUE                                                          C1525300
      NK = KT1(NB1,NB1,K)                                               C1525400
      CALL CYCP(TEM3,XDIC(1,1,NK))                                      C1525500
      IF(LEQU)PRINT 218, NK,NB1,K                                       C1525600
C     FILL OUT ROW K OVER TO DIAGONAL                                   C1525700
      IF(NST1.EQ.0) GO TO 6                                             C1525800
      DO 13  II=1,NST1                                                  C1525900
      I = ST1(II)                                                       C1526000
      KI = KT1(NB1,K,I)                                                 C1526100
      DO 14 M=1,3                                                       C1526200
      DO 14 N=1,3                                                       C1526300
   14 TEM2(M,N) = 0                                                     C1526400
      IF(LEQU)PRINT 220                                                 C1526500
      IF(NST2.EQ.0) GO TO 5048                                          C1526600
      DO 15 LL=1,NST2                                                   C1526700
      L = ST2(LL)                                                       C1526800
      KL = KTC(NB1,K-1,L)                                               C1526900
      IL = KTC(NB1,I-1,L)                                               C1527000
      CALL SUBCP(GAM(1,KL),GAM(1,IL),XMAS(L),TEM4)                      C1527100
      CALL CYADD(TEM2,TEM4,TEM2)                                        C1527200
      IF(LEQU)K1 = K-1                                                  C1527300
      IF(LEQU)I1 = I-1                                                  C1527400
      IF(LEQU)PRINT 221, L,K1,L,I1,L,I1,L,K1,L,KL,IL,L                  C1527500
      IF(LEQU)PRINT 222                                                 C1527600
   15 CONTINUE                                                          C1527700
 5048 CONTINUE                                                          C1527800
      CALL CYADD(TEM1,TEM2,XDIC(1,1,KI))                                C1527900
      IF(LEQU)PRINT 224, KI,K,I                                         C1528000
   13 CONTINUE                                                          C1528100
    6 CONTINUE                                                          C1528200
 5046 CONTINUE                                                          C1528300
      IF(CT4.NE.1) GO TO 17                                             C1528400
      NN = KT1(NB1,NB1,NE1)                                             C1528500
      DO 16  K=1,NBOD                                                   C1528600
      DO 16  I=1,3                                                      C1528700
   16 XDIC(I,I,NN) = XDIC(I,I,NN) + XMAS(K)                             C1528800
      IF(LEQU)PRINT 226, NN,NE1,NE1                                     C1528900
   17 CONTINUE                                                          C1529000
      IF(.NOT. LEQU) GO TO 1001                                         C1529100
      PRINT 230                                                         C1529200
      DO 24 K=1,NB1                                                     C1529300
      DO 24 I=K,NB1                                                     C1529400
      IK = KT1(NB1,I,K)                                                 C1529500
      PRINT 229                                                         C1529600
      PRINT 233, (XDIC(1,L,IK),L=1,3)                                   C1529700
      PRINT 234, I,K,IK,(XDIC(2,L,IK),L=1,3)                            C1529800
      PRINT 233, (XDIC(3,L,IK),L=1,3)                                   C1529900
   24 CONTINUE                                                          C1530000
 1001 CONTINUE                                                          C1530100
C                                                                       C1530200
  202 FORMAT (' SXT = ',Z8,' ELEMENTS OF SXT = ',1015 )                 C1530300
  203 FORMAT (' SVD = ',Z8,' ELEMENTS OF SVD = ',1015 )                 C1530400
  204 FORMAT (' GAM(',I2,') = GAM(',I2,',',I2,') = CAC(',I2,')    ')    C1530500
  205 FORMAT (' GAM(',I2,') = GAM(',I2,',',I2,') = GAM(',I2,') + CBC(',IC1530600
     *2,') ')                                                           C1530700
  206 FORMAT (' BODY ',I2,' IS A LINER OSCILLATOR ')                    C1530800
  207 FORMAT (' NON-ZERO COLUMNS OF ROW ',I2,' OVER TO DIAGONAL ARE ',1CC1530900
     *I5,/)                                                             C1531000
```

```
 206 FORMAT (' XDIC(',I2,') = XDIC(',I2,',',I2,') = XMAS(',I2,')*DYDP(GC1531100
    *AM(',I2,')) = XMAS(',I2,')*DYDP(GAM(',I2,',',I2,')) ')            C1531200
 210 FORMAT  (' XDIC(',I2,') = XDIC(',I2,',',I2,') = XMAS(',I2,')*1 ') C1531300
 211 FORMAT (' BODY ',I2,' IS A RIGID BODY ')                          C1531400
 212 FORMAT (' TO GET INERTIA TENSORS IN ROW ',I2,' COLUMNS 1 THROUGH C1531500
    *',I2,' SUM OVER BODIES ',10I5)                                    C1531600
 213 FORMAT (5X,' TEM3 = TEM3 + XMAS(',I2,')*GAM(',I2,') = TEM3 + XMAS(C1531700
    *',I2,')*GAM(',I2,',',I2,') ')                                     C1531800
 215 FORMAT (5X,' TEM1 = TEM1 + XIC(',I2,') ')                         C1531900
 217 FORMAT (5X,' TEM1 = 0 ,   TEM3 = 0 ')                             C1532000
 218 FORMAT (' XDIC(',I2,') = XDIC(',I2,',',I2,') = SKEW(TEM3) ')      C1532100
 220 FORMAT (5X,' TEM2 = 0 ')                                          C1532200
 221 FORMAT (5X,' TEM4 = XMAS(',I2,')*( GAM(',I2,',',I2,').GAM(',I2,',',C1532300
    *,I2,')*1 - GAM(',I2,',',I2,')GAM(',I2,',',I2,') ) = SUEOP(GAM(',I2C1532400
    *,'),GAM(',I2,'),XMAS(',I2,')) ')                                  C1532500
 222 FORMAT (5X,' TEM2 = TEM2 + TEM4 ')                                C1532600
 224 FORMAT (' XDIC(',I2,') = XDIC(',I2,',',I2,') = TEM1 + TEM2 ')     C1532700
 226 FORMAT  (' XDIC(',I2,') = XDIC(',I2,',',I2,') = (TOTAL MASS)*1 ') C1532800
 228 FORMAT ('1  SUBROUTINE XDY ENTERED ')                             C1532900
 229 FORMAT (' ')                                                      C1533000
 230 FORMAT (3(/))                                                     C1533100
 231 FORMAT (' GAM(',I2,',',I2,') = GAM(',I2,') = ',3D17.8)            C1533200
 232 FORMAT (/,'         COMPUTE ELEMENTS IN ROW ',I2,' OF INERTIA MATRIC1533300
    *X ',/)                                                            C1533400
 233 FORMAT (27X,3D17.8)                                               C1533500
 234 FORMAT (' XDIC(',I2,',',I2,') = XDIC(',I2,') = ',3D17.8)          C1533600
     RETURN                                                            C1533700
     END                                                               C1533800
```

```
C                                                                      C1600000
      SUBROUTINE ETA                                                   C1600100
C      USE TO COMPUTE GYROSCOPIC CROSS COUPLING TERMS DUE TO           C1600200
C        1) INERTIA CROSS COUPLING                                     C1600300
C        2) CENTRIPITAL CROSS COUPLING                                 C1600400
C        3) CORIOLIS CROSS COUPLING                                    C1600500
C                                                                      C1600600
C     LET:                                                             C1600700
C        ETC(I,K) = COMPONENTS OF GYROSCOPIC CROSS COUPLING TORQUE     C1600800
C                   ON NEST K-1                                        C1600900
C                                                                      C1601000
C                                                                      C1601100
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C1601200
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLC, LEQU, LINIT(1)    C1601300
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,    C1601400
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTOFQU ,    C1601500
     *                 LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ       C1601600
C                                                                      C1601700
C                                                                      C1601800
C                                                                      C1601900
C                                                                      C1602000
C                                                                      C1602100
C                                                                      C1602200
      INTEGER                                                          C1602300
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  , C1602400
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    , C1602500
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   , C1602600
```

```
     * SMA    , SMCDUM, SMC    , SMV    , SGK    , SPIDUM, SPI    , SQF    ,  C1602700
     * SGL    , SR    , SSCN   , SSIX   , SVA    , SVB   , SVD    , SVI    ,  C1602800
     * SVM    , SVP   , SVQ    , SXM    , SXT    , TORQ  , SMAL   , SEU    ,  C1602900
     * SC     , SCG   , NFLXB  , SFLX   , SFXM   , NMODS , SFCC   , SCC    ,  C1603000
     * IINIT(1)       , IZINIT(1)      , SD                                   C1603100
C                                                                             C1603200
C                                                                             C1603300
      REAL*8                                                                  C1603400
     * ANGD  (33)     , CNF   (3,10)   , ETIC  (3,10)   , ETMC  (3,10)   ,   C1603500
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                      C1603600
     * THACD (33)     , YMCD  (3,2,11), RINIT (1)      , RZINIT(1)          C1603700
C                                                                             C1603800
C                                                                             C1603900
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,           C1604000
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,           C1604100
     *                LCFDOT , LDCT   , LANGLE , LSETUP , LSIMQ              C1604200
C                                                                             C1604300
C                                                                             C1604400
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RELO(10)                 C1604500
C                                                                             C1604600
C                                                                             C1604700
      COMMON /INTG/ AWORK(200),                                               C1604800
     * CT1            , CT2            , CT3            , CT4            ,   C1604900
     * CTS            , FCON  (33)     , JCCN  (10)     , LCCN  (22)     ,   C1605000
     * MC     (10)    , NB1            , NBCD           , NCTC           ,   C1605100
     * NFER           , NFKC           , NFRC           , NLOR           ,   C1605200
     * NMV            , NMO            , NMOA           , NSVP           ,   C1605300
     * NSVC           , PCCN  (11)     , SD             , SFR   (33)     ,   C1605400
     * SG             , SI    (55)     , SIG            , SL             ,   C1605500
     * SLK    (33)    , SMA   (10)     , SGK   (11)     , SQF   (11)     ,   C1605600
     * SGL    (11)    , SMV            , SR             , SSCN           ,   C1605700
     * SSIX           , SVA            , SVB            , SVD            ,   C1605800
     * SVI            , SVM            , SVP   (22)     , SVQ   (33)     ,   C1605900
     * SXM    (3,10)  , SXT            , TORQ  (97)     , SMAL           ,   C1606000
     * SEU            , NTQ            , SC    (33)     , SCG            ,   C1606100
     * NFLXB          , SFLX           , SFXM  (10)     , NMODS          ,   C1606200
     * SFCC           , SCC   (10)                                           C1606300
C                                                                             C1606400
C                                                                             C1606500
      COMMON /INTG2/                                                          C1606600
     * SCNDUM         , SCN   (9)      , SCRDUM         , SCR   (9)      ,   C1606700
     * SFKDUM         , SFK   (9)      , SIXDUM         , SIX   (9)      ,   C1606800
     * SKDUM          , SK    (9)      , SPIDUM         , SPI   (9)      ,   C1606900
     * SMCDUM         , SMC   (9)                                            C1607000
C                                                                             C1607100
      COMMON /REAL/                                                           C1607200
     * CA     (3,10)  , CAC   (3,10)   , CLM   (10)     , CCMC  (3,11)   ,   C1607300
     * DCMC   (3,11)  , ETC   (3,11)   , ETM   (33)     , FDMC  (3,11)   ,   C1607400
     * GAM    (3,66)  , H              , HM    (3,10)   , HMC   (3,10)   ,   C1607500
     * HMCM   (10)    , PHI   (3,11)   , PLM   (10)     , QF    (3,33)   ,   C1607600
     * QFC    (3,33)  , GL    (3,22)   , QLC   (3,22)   , RDMC  (3,11)   ,   C1607700
     * T              ,                  THA   (33)     , THAD  (33)     ,   C1607800
     * THADM  (10)    , THAW  (10)     , XDIC  (3,3,66), XI    (3,3,10),   C1607900
     * XIC    (3,3,10), XMAS  (10)     , XMN   (33,33)  , XMT   (3,3,10),   C1608000
     * TLG    (33)    , FLA   (3,20)   , FLB   (3,20)   , FLC   (3,20)   ,   C1608100
     * FLD    (3,3,20), FLJ   (3,3,20), CAO   (3,10)   , XIO   (3,3,10),   C1608200
     * FLIRC  (3,10)  , FLCRC (3,10)   , FLAC  (3,20)   , FLQC  (3,20)   ,   C1608300
     * FLCM   (20)    , ZETA  (20)     , FCF   (3,3,40), FCK   (3,40)   ,   C1608400
     * TIMEND                                                                 C1608500
C                                                                             C1608600
```

```
C                                                                       C1608700
      COMMON /REALZ/                                                    C1608800
     * CEDUM (1,3)    , CE     (3,10)  , CBCDUM(1,3)    , CBC    (3,10)  ,  C1608900
     * XMCDUM(1,1,9) , XMC    (3,3,10), CBN(3)                          C1609000
C                                                                       C1609100
C                                                                       C1609200
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))     ,    C1609300
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNF(1,1))    ,    C1609400
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))  ,    C1609500
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1)),    C1609600
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                            C1609700
     *            (FGI,LINIT(1))            ,(CA(1,1),RINIT(1))     ,    C1609800
     *            (CBDUM(1,1),RZINIT(1))    ,(AWORK(1),IINIT(1))    ,    C1609900
     *            (SCNDUM,IZINIT(1))                                       C1610000
C                                                                       C1610100
      INTEGER ST1(10),ST2(10),ST3(10),ST4(10),SET(10),SFXMN             C1610200
      DIMENSION TEM(3),TEM1(3),TEM2(3),TEM3(3),TEM4(3),TEM5(3),TEM6(3)  C1610300
      DIMENSION TEM7(3,3),TEM8(3,3),TEM9(3),TEM10(3),FLEC(3,3)          C1610400
      EQUIVALENCE (LETA,LEQU)                                           C1610500
C                                                                       C1610600
      IF(LEQU)PRINT 240                                                 C1610700
      DO 1  I=1,3                                                       C1610800
      TEM3(I) = 0                                                       C1610900
      TEM6(I) = 0                                                       C1611000
    1 ETC(I,NB1) = 0                                                    C1611100
C                                                                       C1611200
C     USE SETS SSIX AND SSCN TO REDUCE REDUNDANCY IN CROSS COUPLING COMPC1611300
C     COMPUTE INTERTIAL CROSS COUPLING CONTRIBUTIONS FOR EACH BODY      C1611400
      IF(LEQU)PRINT 233                                                 C1611500
      CALL UNPAC(ST1,NST1,SSIX)                                         C1611600
      IF(NST1.EQ.0) GO TO 5049                                          C1611700
      DO 30  II=1,NST1                                                  C1611800
      I = ST1(II)                                                       C1611900
      CALL VXDYUV(FOMC(1,I),XIC(1,1,I),ETIC(1,I))                       C1612000
      IF(LEQU)PRINT 239, I,I,I,I,(ETIC(J,I),J=1,3)                      C1612100
   30 CONTINUE                                                          C1612200
 5049 CONTINUE                                                          C1612300
C                                                                       C1612400
C                                                                       C1612500
C     COMPUTE FORCE ASSOCIATED WITH CENTRIPITAL ACCELERATION OF EACH    C1612600
C     BODY IN THE SET SSCN                                              C1612700
      CALL UNPAC(ST2,NST2,SSCN)                                         C1612800
      IF(LEQU)PRINT 234                                                 C1612900
      IF(NST2.EQ.0) GO TO 5050                                          C1613000
      DO 31 III=1,NST2                                                  C1613100
      II=NST2-(III-1)                                                   C1613200
      I = ST2(II)                                                       C1613300
      IF(I.EQ.1) GO TO 31                                               C1613400
      CALL TRIPVP(FOMC(1,JCON(I)),CBC(1,I),CNF(1,I))                    C1613500
      IF(JCON(I).EQ.1) GO TO 32                                         C1613600
      CALL VECADD(CNF(1,JCON(I)),CNF(1,I),CNF(1,I))                     C1613700
      IF(LEQU)PRINT 235, I,JCON(I),JCON(I),JCON(I),I                    C1613800
      GO TO 31                                                          C1613900
   32 CONTINUE                                                          C1614000
      IF(LEQU)PRINT 236, I,JCON(I),JCON(I),I                            C1614100
   31 CONTINUE                                                          C1614200
 5050 CONTINUE                                                          C1614300
      IF(LEQU)PRINT 229                                                 C1614400
      IF(NST2.EQ.0) GO TO 5051                                          C1614500
      DO 33 III=1,NST2                                                  C1614600
```

```
      II=NST2-(III-1)                                                   C1614700
      I = ST2(II)                                                       C1614800
      IF(I.EQ.1) GO TO 33                                               C1614900
      CALL TRIPVP(FOMC(1,I),CAC(1,I),TEM)                               C1615000
      IF(LEQU)PRINT 237, I,I,I,I,I,I                                    C1615100
   35 CALL VECADD(CNF(1,I),TEM,CNF(1,I))                                C1615200
      CALL SCLV(XMAS(I),CNF(1,I),CNF(1,I))                              C1615300
   33 CONTINUE                                                          C1615400
 5051 CONTINUE                                                          C1615500
      IF(.NOT. LEQU) GO TO 1000                                         C1615600
      PRINT 229                                                         C1615700
      IF(NST2.EQ.0) GO TO 1000                                          C1615800
      DO 36  II=1,NST2                                                  C1615900
      I = ST2(II)                                                       C1616000
      IF(I.EQ.1) GO TO 36                                               C1616100
      PRINT 238, I,(CNF(J,I),J=1,3)                                     C1616200
   36 CONTINUE                                                          C1616300
 1000 CONTINUE                                                          C1616400
C                                                                       C1616500
C                                                                       C1616600
C                                                                       C1616700
C     DEFORMATION CONTRIBUTIONS TO CROSS COUPLING                       C1616800
      IF(NFLXE.EQ.0) GO TO 38                                           C1616900
      MN = 0                                                            C1617000
      CALL UNPAC(SET,NSET,SFLX)                                         C1617100
      IF(LEQU) PRINT 252, (SET(I),I=1,NSET)                             C1617200
      DO 40 NN=1,NSET                                                   C1617300
      N = SET(NSET+1-NN)                                                C1617400
      IF(LEQU) PRINT 253, N                                             C1617500
      DO 37 I=1,3                                                       C1617600
      TEM9(I) = 0.                                                      C1617700
      DO 37 J=1,3                                                       C1617800
   37 TEM8(I,J) = 0.                                                    C1617900
      IF(LEQU) PRINT 231                                                C1618000
      SFXMN = SFXM(N)                                                   C1618100
      DO 39 M=1,SFXMN                                                   C1618200
      MN = MN + 1                                                       C1618300
      CALL DYOF(FLQC(1,MN),TEM7)                                        C1618400
      CALL TENTRN(FLE(1,1,MN),XMC(1,1,N),FLEC)                          C1618500
      CALL DYADD(FLEC,TEM7,TEM7)                                        C1618600
      CALL SCLD(THAD(NFER+MN),TEM7,TEM7)                                C1618700
      CALL DYADD(TEM8,TEM7,TEM8)                                        C1618800
      CALL SCLV(THAD(NFER+MN),FLAC(1,MN),TEM10)                         C1618900
      CALL VECADD(TEM9,TEM10,TEM9)                                      C1619000
      IF(.NOT.LEQU) GO TO 5001                                          C1619100
      PRINT 251, MN,N,MN,N                                              C1619200
      NFQX = NFER+MN                                                    C1619300
      PRINT 232, NFQX,MN,MN                                             C1619400
      PRINT 243, NFQX,MN                                                C1619500
 5001 CONTINUE                                                          C1619600
   39 CONTINUE                                                          C1619700
      CALL DYDCTV(TEM8,FCMC(1,N),FLIRC(1,N))                            C1619800
      CALL SCLV(2.000,TEM9 ,TEM9 )                                      C1619900
      CALL SCLV(XMAS(N),TEM9 ,TEM9 )                                    C1620000
      CALL VECROS (FCMC(1,N),TEM9 ,FLCRC(1,N))                          C1620100
      CALL VECADD(ETIC(1,N),FLIRC(1,N),ETIC(1,N))                       C1620200
      CALL VECADD(CNF(1,N),FLCRC(1,N),CNF(1,N))                         C1620300
      IF(.NOT.LEQU) GO TO 5000                                          C1620400
      PRINT 244, N,N,(FLIRC(I,N),I=1,3)                                 C1620500
      PRINT 246, N,N,N,(ETIC(I,N),I=1,3)                                C1620600
```

```
      PRINT 245, N,N,N,(FLCRC(I,N),I=1,3)                               C1620700
      PRINT 247, N,N,N,(CNF(I,N),I=1,3)                                 C1620800
 500C CONTINUE                                                          C1620900
   4C CONTINUE                                                          C1621000
   3E CONTINUE                                                          C1621100
C                                                                       C1621200
C                                                                       C1621300
C                                                                       C1621400
C     ELIMINATE REDUNDANT COMPUTATION IN MOMENTUM WHEEL CROSS           C1621500
C      COUPLING COMPUTATION BY ETMC                                     C1621600
C                                                                       C1621700
      IF(NMC.EQ.0) GO TO 5CE2                                           01621800
      IF(LEQU)PRINT 208                                                 C1621900
      DO 1C  M=1,NMO                                                    C1622000
      CALL VECROS (FCMC(1,MO(M)),FMC(1,M),ETMC(1,M))                    C1622100
      CALL SCLV(HMOM(M),ETMC(1,M),ETMC(1,M))                            C1622200
      IF(LEQU)PRINT 207, M,M,MO(M),M,(ETMC(I,M),I=1,3)                  C1622300
   1C CONTINUE                                                          C1622400
 5052 CONTINUE                                                          C1622500
C                                                                       C1622600
      DO 2  K=1,NBOD                                                    C1622700
      DO 3  I=1,3                                                       C1622800
    3 ETC(I,K) = 0                                                      C1622900
      K1 = K-1                                                          C1623000
      IF(LEQU)PRINT 225, K1                                             C1623100
C     SET UP SUMMATION SETS FOR NEST K-1                                C1623200
C        INERTIA CROSS COUPLING                                         C1623300
      CALL UNPAC(ST1,NST1,SIX(K1))                                      C1623400
      IF(LEQU)PRINT 200, K1,SIX(K1),(ST1(I),I=1,NST1)                   C1623500
C        CENTRIPITAL CROSS COUPLING                                     C1623600
      CALL UNPAC(ST2,NST2,SCN(K1))                                      C1623700
      IF(LEQU)PRINT 201, K1,SCN(K1),(ST2(I),I=1,NST2)                   C1623800
C        CORIOLLIS CROSS COUPLING                                       C1623900
      CALL UNPAC(ST3,NST3,SCR(K1))                                      C1624000
      IF(LEQU)PRINT 202, K1,SCR(K1),(ST3(I),I=1,NST3)                   C1624100
C        MOMENTUM WHEEL CROSS COUPLING                                  C1624200
      CALL UNPAC(ST4,NST4,SMC(K1))                                      C1624300
      IF(.NOT. LEQU) GO TO 1001                                         C1624400
      PRINT 209, K1,SMC(K1),(ST4(I),I=1,NST4)                           C1624500
C                                                                       C1624600
C                                                                       C1624700
      PRINT 229                                                         C1624800
      PRINT 203, K                                                      C1624900
      IF(K.NE.1) GO TO 5                                                C1625000
      PRINT 203, NB1                                                    C1625100
      PRINT 230                                                         C1625200
    5 CONTINUE                                                          C1625300
C                    INERTIA CROSS COUPLING                             C1625400
      PRINT 241                                                         C1625500
      PRINT 226                                                         C1625600
 1001 CONTINUE                                                          C1625700
      IF(.NOT.RBLO(K)) GO TO 4                                          C1625800
      IF(LEQU)PRINT 204, K,K                                            C1625900
C       SUM OVER ONLY THOSE BODIES OF NEST K-1 WHICH SIGNIFICANTLY      C1626000
C         CONTRIBUTE TO INERTIAL CROSS COUPLING IN EQUATIONS OF         C1626100
C         MOTION OF NEST K-1                                            C1626200
      IF(NST1.EQ.0) GO TO 4                                             C1626300
      DO 6  LL=1,NST1                                                   C1626400
      L = ST1(LL)                                                       C1626500
      IF(LEQU)PRINT 205, K,K,L                                          C1626600
```

```
    6 CALL VECSUB(ETC(1,K),ETIC(1,L),ETC(1,K))                          C1626700
C     NOTE REDUNDANT VECTOR ADDITIONS LESS COSTLY THAN LOGIC NEEDED TO  C1626800
C      AVOID THEM                                                       C1626900
    4 CONTINUE                                                          C1627000
      IF(LEQU)PRINT 216, K,(ETC(I,K),I=1,3)                             C1627100
C                                                                       C1627200
C                        CENTRIPITAL CROSS COUPLING                     C1627300
      IF(LEQU)PRINT 241                                                 C1627400
      IF(LEQU)PRINT 227                                                 C1627500
      IF(NST2.EQ.0) GO TO 7                                             C1627600
      DO 8  I=1,3                                                       C1627700
      TEM4(I) = 0                                                       C1627800
    8 TEM2(I) = 0                                                       C1627900
C       SUM OVER ONLY THOSE BODIES OF NEST K-1 WHICH SIGNIFICANTLY      C1628000
C         CONTRIBUTE TO CENTRIFITAL CROSS COUPLING IN EQUATIONS OF      C1628100
C         MOTION OF NEST K-1                                            C1628200
      IF(LEQU)PRINT 206                                                 C1628300
      IF(NST2.EQ.0) GO TO 5053                                          C1628400
      DO 9  LL=1,NST2                                                   C1628500
      L = ST2(LL)                                                       C1628600
      LC = L                                                            C1628700
      IF(LC.EQ.1) GO TO 9                                               C1628800
      IF(RELO(K)) GO TO 22                                              C1628900
      CALL VECSUB(TEM4,CNF(1,L),TEM4)                                   C1629000
      IF(LEQU)PRINT 211, L                                              C1629100
      GO TO 9                                                           C1629200
   22 IF(K.NE.1) GO TO 14                                               C1629300
      CALL VECSUB(TEM3,CNF(1,L),TEM3)                                   C1629400
      IF(LEQU)PRINT 212, L                                              C1629500
   14 KL = KTC(NB1,K-1,L)                                               C1629600
      CALL VECROS (GAM(1,KL),CNF(1,L),TEM)                              C1629700
      CALL VECSUB(TEM4,TEM,TEM4)                                        C1629800
      IF(LEQU)K1 = K-1                                                  C1629900
      IF(LEQU)PRINT 213, K1,L,L                                         C1630000
      IF(LEQU)PRINT 229                                                 C1630100
    9 CONTINUE                                                          C1630200
 5053 CONTINUE                                                          C1630300
      CALL VECADD(ETC(1,K),TEM4,ETC(1,K))                               C1630400
      IF(LEQU)PRINT 214, K, K, (ETC(I,K),I=1,3)                         C1630500
      IF(K.NE.1) GO TO 7                                                C1630600
      CALL VECADD(ETC(1,NB1),TEM3,ETC(1,NB1))                           C1630700
      IF(LEQU)PRINT 215, NB1,NE1, (ETC(I,NB1),I=1,3)                    C1630800
    7 CONTINUE                                                          C1630900
C                                                                       C1631000
C                        CORIOLIS CROSS COUPLING                        C1631100
      IF(NST3.EQ.0) GO TO 15                                            C1631200
      IF(LEQU)PRINT 241                                                 C1631300
      IF(LEQU)PRINT 228                                                 C1631400
      DO 16 I=1,3                                                       C1631500
   16 TEM5(I) = 0                                                       C1631600
C       SUM OVER ONLY THOSE BODIES OF NEST K-1 WHICH SIGNIFICANTLY      C1631700
C         CONTRIBUTE TO CORIOLIS CROSS COUPLING IN EQUATIONS OF         C1631800
C         MOTION OF NEST K-1                                            C1631900
      IF(LEQU)PRINT 217                                                 C1632000
      IF(NST3.EQ.0) GO TO 5054                                          C1632100
      DO 17  LL=1,NST3                                                  C1632200
      L = ST3(LL)                                                       C1632300
      CALL VECROS (FCMC(1,L),RCMC(1,L),TEM)                             C1632400
      DO 18  I=1,3                                                      C1632500
   18 TEM2(I) =  2.D0*XMAS(L)*TEM(I)                                    C1632600
```

```
      IF(LEQU)PRINT 219, L,L,L                                          C1632700
      IF(RBLJ(K)) GO TO 20                                              01632800
      DO 21 I=1,3                                                       C1632900
   21 TEM5(I) = -TEM2(I)                                                C1633000
      IF(LEQU)PRINT 221                                                 C1633100
      GO TO 17                                                          C1633200
   20 KL = KTC(NB1,K-1,L)                                               C1633300
      CALL VECROS (GAM(1,KL),TEM2,TEM)                                  C1633400
      CALL VECSUB(TEM5,TEM,TEM5)                                        C1633500
      IF(LEQU)K1 = K- 1                                                 C1633600
      IF(LEQU)PRINT 222, K1,L                                           C1633700
      IF(K.NE.1) GO TO 15                                               C1633800
      CALL VECSUB(TEM6,TEM2,TEM6)                                       C1633900
      IF(LEQU)PRINT 220                                                 C1634000
   15 CONTINUE                                                          C1634100
      IF(LEQU)PRINT 229                                                 C1634200
   17 CONTINUE                                                          C1634300
 5054 CONTINUE                                                          C1634400
      CALL VECADD(ETC(1,K),TEM5,ETC(1,K))                               C1634500
      IF(LEQU)PRINT 223, K,K,(ETC(I,K),I=1,3)                           C1634600
      IF(K.NE.1) GO TO 15                                               C1634700
      CALL VECADD(ETC(1,NB1),TEM6,ETC(1,NB1))                           01634800
      IF(LEQU)PRINT 224, NB1,NE1,(ETC(I,NB1),I=1,3)                     C1634900
   15 CONTINUE                                                          C1635000
C                                                                       C1635100
C                                                                       C1635200
C                    MOMENTUM WHEEL CROSS COUPLING                      C1635300
      IF(NST4.EQ.0) GO TO 5055                                          C1635400
      IF(LEQU)PRINT 241                                                 C1635500
      IF(LEQU)PRINT 210                                                 C1635600
C                                                                       C1635700
      DO 11  MM=1,NST4                                                  C1635800
      M = ST4(MM)                                                       C1635900
      IF(LEQU)PRINT 218, K,K,M                                          C1636000
      CALL VECSUB(ETC(1,K),ETMC(1,M),ETC(1,K))                          C1636100
   11 CONTINUE                                                          C1636200
 5055 CONTINUE                                                          C1636300
      IF(LEQU)PRINT 216, K,(ETC(I,K),I=1,3)                             C1636400
C                                                                       C1636500
C                                                                       C1636600
C                                                                       C1636700
    2 CONTINUE                                                          C1636800
C                                                                       01636900
  200 FORMAT (5X,'  SIX(',I2,') = ',28,' CONTRIBUTERS TO INERTIAL CROSS C1637000
     *COUPLING ARE BODIES    ',1015)                                    C1637100
  201 FORMAT (5X,'  SCN(',I2,') = ',28,' CONTRIBUTERS TO CENTRIPITAL CRC1637200
     *SS COUPLING ARE BODIES ',1015)                                    C1637300
  202 FORMAT (5X,'  SCR(',I2,') = ',28,' CONTRIBUTERS TO CORIOLIS CROSS C1637400
     *COUPLING ARE BODIES    ',1015)                                    C1637500
  203 FORMAT (' CROSS COUPLING TORQUE ETC(',I2,') = 0 ')                C1637600
  204 FORMAT ('   ETC(',I2,') = ETC(',I2,') + ETIC(',I2,') ')           C1637700
  205 FORMAT ('   ETC(',I2,') = ETC(',I2,') - ETIC(',I2,') ')           C1637800
  206 FORMAT (20X,'  TEM2 =0, TEM4 = 0 ')                              C1637900
  207 FORMAT (5X,'ETMC(',I2,') = HMCM(',I2,') * (FCMC(',I2,') X HMC(',I2C1638000
     *,')) = ',3D17.8)                                                  C1638100
  208 FORMAT (///,5X,' INERTIAL CROSS COUPLING TERM FOR EACH MOMENTUM WHEC1638200
     *EL ',/)                                                           C1638300
  209 FORMAT (5X,'  SMC(',I2,') = ',28,' CONTRIBUTERS TO MOMENTUM WHEEL C1638400
     *CROSS COUPLING ARE WHEELS' ,1015)                                 C1638500
  210 FORMAT ('      MOMENTUM WHEEL CROSS COUPLING EFFECTS ')           C1638600
```

```
211 FORMAT (20X,'  TEM4 = TEM4 - CNF(',I2,') ')                                C1638700
212 FORMAT (20X,'  TEM3 = TEM3 - CNF(',I2,') ')                                C1638800
213 FORMAT (20X,'  TEM4 = TEM4 - GAM(',I2,',',I2,') X CNF(',I2,') ')          C1638900
214 FORMAT ('  ETC(',I2,') = ETC(',I2,') + TEM4  = ',3D17.8)                   C1639000
215 FORMAT ('  ETC(',I2,') = ETC(',I2,') + TEM3  = ',3D17.8)                   C1639100
216 FORMAT ('  ETC(',I2,') = ',3D17.8)                                         C1639200
217 FORMAT (40X,'  TEM5 = C           ')                                       C1639300
218 FORMAT ('  ETC(',I2,') = ETC(',I2,') - ETMC(',I2,') ')                     C1639400
219 FORMAT (40X,'  TEM2 =  2*XMAS(',I2,') * FOMC(',I2,') X RCMC(',I2,'C1639500
   *) ')                                                                       C1639600
220 FORMAT (40X,'  TEM6 = TEM6 - TEM2 ')                                      C1639700
221 FORMAT (40X,'  TEM5 = -TEM2 ')                                            C1639800
222 FORMAT (40X,'  TEM5 = TEM5 - GAM(',I2,',',I2,') X TEM2 ')                 C1639900
223 FORMAT ('  ETC(',I2,') = ETC(',I2,') + TEM5  = ',3D17.8)                   C1640000
224 FORMAT ('  ETC(',I2,') = ETC(',I2,') + TEM6  = ',3D17.8)                   C1640100
225 FORMAT(7(/),                                                               C1640200
   *'  BODY LABELS OF THOSE BODIES WHICH SIGNIFICANTLY CONTC1640300
   *RIBUTE TO GYROSCOPIC CROSS COUPLING TORQUES ON NEST ',I2,/)                 C1640400
226 FORMAT ( '       INERTIA CROSS COUPLING EFFECTS ')                         C1640500
227 FORMAT (  20X,'      CENTRIPITAL CROSS COUPLING EFFECTS ')                  C1640600
228 FORMAT (  40X,'      CORICLIS CROSS COUPLING EFFECTS ')                     C1640700
229 FORMAT ( '    ')                                                           C1640800
230 FORMAT ('   TEM3 = 0,  TEM6 = 0 ')                                         C1640900
231 FORMAT (10X,'  TEM8 = 0',45X,'TEM9 = 0 ',/)                               C1641000
232 FORMAT (10X,'  TEM8 = TEM8 + THAD(',I2,')*(FLEC(',I2,') + SKEW(FLOC1641100
   *(',I2,')) ')                                                               C1641200
233 FORMAT (//,5X,' INERTIAL CROSS COUPLING CONTRIBUTIONS OF BODIES INC1641300
   *SET SSIX ',/)                                                              C1641400
234 FORMAT (//,5X,' INERTIAL FORCE ASSOCIATED WITH CENTRIPITAL ACCELERC1641500
   *ATION OF CENTER OF MASS OF EACH BODY IN SSCN ',/)                          C1641600
235 FORMAT (8X,'CNF(',I2,') = CNF(',I2,') + FOMC(',I2,') X (FOMC(',I2,C1641700
   *') X CBC(',I2,',I2,')) ')                                                  C1641800
236 FORMAT (8X,'CNF(',I2,') = ',10X,       'FOMC(',I2,') X (FOMC(',I2,C1641900
   *') X CBC(',I2,',I2,')) ')                                                  C1642000
237 FORMAT (8X,'CNF(',I2,') = XMAS(',I2,') * (CNF(',I2,') + FOMC(',I2,C1642100
   *') X (FOMC(',I2,') X CAC(',I2,',I2,')) ')                                  C1642200
238 FORMAT (8X,'CNF(',I2,') = ',3D17.8)                                        C1642300
239 FORMAT (8X,'ETIC(',I2,') = FOMC(',I2,') X (XIC(',I2,') . FOMC(',I2C1642400
   *,',I2,')) = ',3D17.8)                                                      C1642500
240 FORMAT ('1  SUBROUTINE ETA ENTERED ',2(/))                                 C1642600
241 FORMAT (3(/))                                                              C1642700
243 FORMAT (65X,'TEM9 = TEM9 + THAD(',I2,')*FLAC(',I2,',I2,')')                C1642800
244 FORMAT (/,'  FLIRC(',I2,') = TEM8.FOMC(',I2,') =',6X,3D12.5)               C1642900
246 FORMAT ('  ETIC(',I2,') = ETIC(',I2,') + FLIRC(',I2,') =',3D12.5)         C1643000
245 FORMAT (/,55X,'FLCRC(',I2,') = 2*XMAS(',I2,')*FOMC(',I2,') X TEM9 C1643100
   *=',3D12.5)                                                                 C1643200
247 FORMAT (53X,'  CNF(',I2,') = CNF(',I2,') + FLCRC(',I2,') =',9X,3D1C1643300
   *2.5)                                                                       C1643400
251 FORMAT (10X,'  FLEC(',I2,') = XMC(',I2,')*FLE(',I2,')*XMC(',I2,')*C1643500
   **T ')                                                                      C1643600
252 FORMAT (//,5X,' ELASTIC CROSS COUPLING CONTRIBUTIONS DUE TO FLEXIBC1643700
   *ILITY OF BODIES',1CI5,//)                                                  C1643800
253 FORMAT (///,35X,'  ELASTIC DEFORMATION EFFECTS DUE TO BODY',I5,/) C1643900
      RETURN                                                                   C1644000
      END                                                                      C1644100
```

```
C                                                                       C1700000
      SUBROUTINE TORQUE(Y,YD,NEQ)                                       C1700100
C                                                                       C1700200
C                                                                       C1700300
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1700400
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C1700500
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1700600
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1700700
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1700800
C                                                                       C1700900
C                                                                       C1701000
      INTEGER                                                           C1701100
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  ,  C1701200
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C1701300
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C1701400
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,  C1701500
     * SQL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,  C1701600
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  C1701700
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  C1701800
     * IINIT(1)      , IZINIT(1)     , SD                              C1701900
C                                                                       C1702000
C                                                                       C1702100
      REAL*8                                                            C1702200
     * ANGD  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C1702300
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                 C1702400
     * THADD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C1702500
C                                                                       C1702600
C                                                                       C1702700
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1702800
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1702900
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1703000
C                                                                       C1703100
C                                                                       C1703200
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C1703300
C                                                                       C1703400
C                                                                       C1703500
      COMMON /INTG/    AWORK (200)    ,                                C1703600
     * CT1             , CT2           , CT3           , CT4           ,  C1703700
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C1703800
     * MC    (10)      , NBI           , NBOD          , NCTC          ,  C1703900
     * NFER            , NFRC          , NFRC          , NLGR          ,  C1704000
     * NMV             , NMO           , NMOA          , NSVP          ,  C1704100
     * NSVC            , PCON  (11)    , SD            , SFR   (33)    ,  C1704200
     * SG              , SI    (55)    , SIG           , SL            ,  C1704300
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)    ,  C1704400
     * SQL   (11)      , SMV           , SR            , SSCN          ,  C1704500
     * SSIX            , SVA           , SVB           , SVD           ,  C1704600
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)    ,  C1704700
     * SXM   (3,10)    , SXT           , TORQ  (97)    , SMAL          ,  C1704800
     * SEU             , NTQ           , SC    (33)    , SCG           ,  C1704900
     * NFLXB           , SFLX          , SFXM  (10)    , NMODS         ,  C1705000
     * SFCC            , SCC   (10)                                     C1705100
C                                                                       C1705200
C                                                                       C1705300
      COMMON /INTGZ/                                                    C1705400
     * SCNDUM          , SCN   (9)     , SCRDUM        , SCR   (9)     ,  C1705500
     * SFKDUM          , SFK   (9)     , SIXDUM        , SIX   (9)     ,  C1705600
     * SKDUM           , SK    (9)     , SPIDUM        , SPI   (9)     ,  C1705700
     * SMCDUM          , SMC   (9)                                      C1705800
C                                                                       C1705900
```

```
C                                                                       C1706000
      COMMON /REAL/                                                     C1706100
     * CA    (3,10)   , CAC   (3,10)   , CLM   (10)     , CCMC  (3,11)  , C1706200
     * DCMC  (3,11)   , ETC   (3,11)   , ETM   (33)     , FCMC  (3,11)  , C1706300
     * GAM   (3,66)   , H              , HM    (3,10)   , HMC   (3,10)  , C1706400
     * HMCM  (10)     , PHI   (3,11)   , PLM   (10)     , QF    (3,33)  , C1706500
     * QFC   (3,33)   , QL    (3,22)   , QLC   (3,22)   , RCMC  (3,11)  , C1706600
     * T              ,                  THA   (33)     , THAD  (33)    , C1706700
     * THADW (10)     , THAW  (10)     , XDIC  (3,3,66), XI    (3,3,10), C1706800
     * XIC   (3,3,10), XMAS  (10)     , XMN   (33,33)  , XMT   (3,3,10), C1706900
     * TUG   (33)     , FLA   (3,20)   , FLB   (3,20)   , FLC   (3,20)  , C1707000
     * FLD   (3,3,20), FLJ   (3,3,20), CAD   (3,10)   , XIU   (3,3,10), C1707100
     * FLIRC (3,10)   , FLCRC (3,10)   , FLAC  (3,20)   , FLQC  (3,20)  , C1707200
     * FLCM  (20)     , ZETA  (20)     , FCF   (3,3,40), FCK   (3,40)  , C1707300
     * TIMEND                                                           C1707400
C                                                                       C1707500
C                                                                       C1707600
      COMMON /REAL2/                                                    C1707700
     * CEDUM (1,3)    , CB    (3,10)   , CBCDUM(1,3)    , CBC   (3,10)  , C1707800
     * XMCDUM(1,1,9)  , XMC   (3,3,10), CBN(3)                          C1707900
C                                                                       C1708000
C                                                                       C1708100
      COMMON /SATELL/                                                   C1708200
     * DUMMY(1000)                                                      C1708300
C                                                                       C1708400
C                                                                       C1708500
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      , C1708600
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     , C1708700
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   , C1708800
     *            (FLB(1,1),FLQ(1,1))        ,(FLE(1,1,1),FLD(1,1,1)), C1708900
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                         C1709000
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))      , C1709100
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     , C1709200
     *            (SCNDUM,IZINIT(1))                                   C1709300
C                                                                       C1709400
      EQUIVALENCE (LTORQU,LEQL)                                         C1709500
C                                                                       C1709600
      DIMENSION Y(NEQ),YD(NEQ)                                          C1709700
C                                                                       C1709800
C                                                                       C1709900
C                                                                       C1710000
C                                                                       C1710100
C                    SYMBOL LISTING OF PARAMETERS USED FROM COMMON      C1710200
C                                                                       C1710300
C          NBOD = TOTAL NUMBER OF RIGID BODIES AND POINT MASSES         C1710400
C           NBI = NBOD + 1                                              C1710500
C             M = GIMBAL AXIS LABEL                                     C1710600
C           K-1 = HINGE POINT AT WHICH GIMBAL AXIS M IS LOCATED         C1710700
C      JCON(K)  = LABEL OF BODY INBOARD OF HINGE POINT K-1              C1710800
C             K = LABEL OF BODY OUTBOARD OF HINGE POINT K-1             C1710900
C      RELC(K)  = TRUE IF BODY K IS A RIGID BODY, FALSE OTHERWISE       C1711000
C      XMAS(K)  = MASS OF BODY K, (M)                                   C1711100
C   XIC(I,J,K)  = INERTIA TENSOR OF BODY K ABOUT ITS CENTER OF MASS     C1711200
C                 RELATIVE TO FRAME OF COMPUTATION, (M*L**2)            C1711300
C    QFC(I,M)   = COMPONENTS RELATIVE TO COMPUTING FRAME OF UNIT        C1711400
C                 VECTOR ALONG GIMBAL AXIS M                            C1711500
C       THA(M)  = DISPLACEMENT ABOUT OR ALONG GIMBAL AXIS M (R OR L)    C1711600
C      THAD(M)  = RATE ABOUT OR ALONG GIMBAL AXIS M (R/T OR L/T)        C1711700
C   PHI(I,NBI)  = RESULTANT EXTERNAL FORCE ACTING ON COMPOSITE SYSTEM (M.C1711800
C     PHI(I,K)  = RESULTANT EXTERNAL TORQUE ON NEST K-1                 C1711900
```

```
C          MW = MOMENTUM WHEEL LABEL                                        C1712000
C       NO(MW) = BODY IN WHICH MOMENTUM WHEEL MW IS EMBEDDED                C1712100
C     EMC(I,MW) = COMPONENTS OF UNIT VECTOR ALONG SPIN AXIS OF WHEEL MW     C1712200
C                 (RELATIVE TO COMPUTING FRAME)                           C1712300
C    GAM(I,KL) = COMPONENTS OF VECTOR FROM HINGE POINT K-1 TO CENTER OF     C1712400
C                MASS OF BODY L, WHERE KL = KTO(NBI,K-1,L)                 C1712500
C      SK(K-1) = CODED WORD, ALL BODIES IN NEST K-1                        C1712600
C     SMC(K-1) = CODED WORD, ALL MOMENTUM WHEELS IN NEST K-1               C1712700
C   XMC(I,J,L) = TRANSFORMATION MATRIX, BODY L TO COMPUTING FRAME          C1712800
C  XMC(I,J,0) = TRANSFORMATION MATRIX, INERTIAL TO COMPUTING FRAME         C1712900
C           Y = SOLUTION ARRAY, CONTAINS INTEGRATED PARAMETERS             C1713000
C          YD = EQUATION ARRAY, SENT TO RUNGE FOR INTEGRATION              C1713100
C         NEQ = NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS DEFINED       C1713200
C               OUTSIDE OF SUBROUTINE TORQUE                               C1713300
C                                                                          C1713400
C                                                                          C1713500
C                                                                          C1713600
C                  INPUT OF USER REQUIRED DATA FOR SUBROUTINE TORQUE       C1713700
C                                                                          C1713800
C           THE USER MAY APPLY ONE OF THREE OPTIONS                        C1713900
C             1) PREFERABLE, DEFINE ALL USER REQUIRED DATA ON              C1714000
C                   'DATA' CARDS WITHIN SUBROUTINE TORQUE                  C1714100
C             2) WRITE SUBROUTINE INTOR AND PASS ALL USER REQUIRED DATA    C1714200
C                   THROUGH COMMON IN /SATELL/                             C1714300
C             3) READ INPUT DATA ON FIRST PASS THROUGH TORQUE              C1714400
C                   CT4 = 1 ON FIRST PASS THROUGH, STORE DATA              C1714500
C                         IN /SATELL/                                      C1714600
C                                                                          C1714700
C                                                                          C1714800
C                                                                          C1714900
C                                                                          C1715000
C                                                                          C1715100
C                  REACTION TORQUE ACTING ACROSS OR ALONG GIMBAL AXIS M    C1715200
C                    AT HINGE POINT K-1 DUE TO :                           C1715300
C                    LINEAR SPRINGS                                        C1715400
C                    LINEAR VISCOUS DAMPERS                                C1715500
C                    MOTORS                                                C1715600
C     LET:                                                                 C1715700
C       SPR(M) = SPRING CONSTANT ABOUT OR ALONG GIMBAL AXIS M              C1715800
C                (M*L**2/T**2 OR M/T**2)                                   C1715900
C       DPC(M) = DAMPING CONSTANT ABOUT OR ALONG GIMBAL AXIS M             C1716000
C                (M*L**2/T OR M/T)                                         C1716100
C       CLT(M) = CONTROL TORQUE APPLIED BY MOTOR ABOUT OR ALONG GIMBAL     C1716200
C                AXIS M  M*L**2/T**2 OR M*L/T**2                           C1716300
C                                                                          C1716400
                 DIMENSION TEM(3)                                          C1716500
C                SPRING TORQUE                                             C1716600
C                SPR(M) = USER INPUT                                       C1716700
                 A = SPR(M)*THA(M)                                         C1716800
                 CALL SCLV(A,QFC(1,M),TEM)                                 C1716900
                 CALL VECSUB(PHI(1,K),TEM,PHI(1,K))                        C1717000
C                DAMPER TORQUE                                             C1717100
C                DPC(M) = USER INPUT                                       C1717200
                 A = DPC(M)*THAD(M)                                        C1717300
                 CALL SCLV(A,QFC(1,M),TEM)                                 C1717400
                 CALL VECSUB(PHI(1,K),TEM,PHI(1,K))                        C1717500
C                MOTOR TORQUE                                              C1717600
C                CLT(M) = FUNCTION OF STATE VARIABLES, USER DEF.           C1717700
                 CALL SCLV(CLT(M),QFC(1,M),TEM)                            C1717800
                 CALL VECADD(PHI(1,K),TEM,PHI(1,K))                        C1717900
```

```
C                                                                       C1718000
C                                                                       C1718100
C                                                                       C1718200
C                                                                       C1718300
C                                                                       C1718400
C                    REACTION TORQUES ON SYSTEM DUE TO A CONTROL TORQUE  C1718500
C                     APPLIED TO MOMENTUM WHEEL MW                      C1718600
C      LET:                                                             C1718700
C       CLM(MW) = CONTROL TORQUE APPLIED TO WHEEL MW ABOUT ITS SPIN AXIS C1718800
C                 USER DEFINED FUNCTION OF STATE VARIABLES (M*L**2/T**2) C1718900
C                                                                       C1719000
C                  C      CLM(MW) = USER DEFINED FUNCTION OF STATE VARS. C1719100
C                                                                       C1719200
C                                                                       C1719300
C                                                                       C1719400
C                                                                       C1719500
C                                                                       C1719600
C                                                                       C1719700
C                    REACTION TORQUES ON SYSTEM DUE TO A LOCALLY        C1719800
C                     APPLIED EXTERNAL FORCE (I.E. A GAS JET)           C1719900
C      LET:                                                             C1720000
C             J = INTEGER LABEL ASSIGNED TO GAS JET                     C1720100
C             L = BODY TO WHICH EXTERNAL FORCE DIRECTLY APPLIED         C1720200
C        RJ(I) = RADIUS VECTOR FROM CENTER OF MASS OF BODY L TO         C1720300
C                GAS JET J, (COMPONENTS RELATIVE TO BODY L COORDINATES) C1720400
C        FJ(I) = COMPONENTS OF APPLIED FORCE DUE TO GAS JET J, (RELATVE C1720500
C                TO BODY L COORDINATES) USER DEFINED FUNCTION OF STATE  C1720600
C                VARIABLES (M*L/T**2)                                   C1720700
C                                                                       C1720800
C               DIMENSION RJC(3),FJC(3),TEM(3),TEM1(3),RJ(3),FJ(3)      C1720900
C               INTEGER S1(10),NS1                                      C1721000
C               LOGICAL CTAIN                                           C1721100
C         C     RJ(I) = USER INPUT                                      C1721200
C               CALL VECTRN(RJ,XMC(1,1,L),RJC)                          C1721300
C         C     FJ(I) = USER DEFINED FUNCTION OF STATE VARIABLES        C1721400
C               CALL VECTRN(FJ,XMC(1,1,L),FJC)                          C1721500
C               CALL VECADD(PHI(1,NB1),FJC,PHI(1,NE1))                  C1721600
C               DO 3  K=1,NBOD                                          C1721700
C               CALL UNPAC(S1,NS1,SK(K-1))                              C1721800
C               IF(.NOT.CTAIN(L,S1,NS1)) GO TO 3                        C1721900
C               IF(RBLO(K)) GO TO 4                                     C1722000
C               CALL VECADD(PHI(1,K),FJC,PHI(1,K))                      C1722100
C               GO TO 3                                                 C1722200
C             4 KL = KTO(NB1,K-1,L)                                     C1722300
C               CALL VECADD(GAM(1,KL),RJC,TEM)                          C1722400
C               CALL VECROS (TEM,FJC,TEM1)                              C1722500
C               CALL VECADD(PHI(1,K),TEM1,PHI(1,K))                     C1722600
C             3 CONTINUE                                                C1722700
C                                                                       C1722800
C                                                                       C1722900
C                                                                       C1723000
C                                                                       C1723100
C                                                                       C1723200
C                                                                       C1723300
C                    REACTION TORQUES ON SYSTEM DUE TO GRAVITATIONAL    C1723400
C                     EFFECTS ON AN EARTH BASED SYSTEM                  C1723500
C      LET:                                                             C1723600
C          GRAV = ACCELERATION OF GRAVITY (L/T**2)                      C1723700
C         BF(I) = COMPONENTS OF UNIT VECTOR FROM INERTIAL ORIGIN TO COMP.C1723800
C                 SYSTEM CENTER OF MASS, (RELATIVE TO INERTIAL FRAME)   C1723900
```

```
C                THAT IS, PARALLEL TO DIRECTION OF GRAVITY FORCE          C1724000
C                                                                         C1724100
C               INTEGER S1(10),NS1                                        C1724200
C               DIMENSION TEM(3),BFC(3),BH(3)                             C1724300
C         C     BH(I) = USER INPUT                                        C1724400
C               CALL VECTRN(BF,XMC(1,1,0),BHC)                            C1724500
C               DO 4  K=1,NBOD                                            C1724600
C         C     GRAV = USER INPUT                                         C1724700
C               A = XMAS(K)*GRAV                                          C1724800
C               CALL SCLV(A,BFC,TEM)                                      C1724900
C               CALL VECSUB(PHI(1,NB1),TEM,PHI(1,NB1))                    C1725000
C               IF(RBLO(K)) GO TO 5                                       C1725100
C               CALL VECSUB(PHI(1,K),TEM,PHI(1,K))                        C1725200
C               GO TO 4                                                   C1725300
C             5 CALL UNPAC(S1,NS1,SK(K-1))                                C1725400
C               DO 4  LL=1,NS1                                            C1725500
C               L = S1(LL)                                                C1725600
C               KL = KTO(NB1,K-1,L)                                       C1725700
C               A = XMAS(L)*GRAV                                          C1725800
C               CALL SCLV(A,BFC,TEM)                                      C1725900
C               CALL VECROS (GAM(1,KL),TEM,TEM1)                          C1726000
C               CALL VECSUB(PHI(1,K),TEM1,PHI(1,K))                       C1726100
C             4 CONTINUE                                                  C1726200
C                                                                         C1726300
C                                                                         C1726400
C                                                                         C1726500
C                                                                         C1726600
C                                                                         01726700
C                     KEPLERIAN ORBIT                                     C1726800
C     LET:                                                                C1726900
C      CE(I,0) = COMPONENTS OF VECTOR FROM EARTH'S CENTER TO COMPOSITE     C1727000
C                SYSTEM CENTER OF MASS, RELATIVE TO INERTIAL REFERENCE    C1727100
C                FRAME, ORBIT ASSUMED TO BE IN INERTIAL 2-3 PLANE         C1727200
C           ASM = SEMI-MAJOR AXIS OF ELLIPTIC ORBIT INERTIAL 2 DIRECTION  C1727300
C           ECC = ORBIT ECCENTRICITY                                      C1727400
C           TPP = TIME OF PERIHELION PASSAGE, (T)                         C1727500
C           GEV = EARTH'S GRAVITATIONAL CONSTANT, (L**3/T**2)             C1727600
C           ETE = MEAN MOTION                                             C1727700
C           AME = MEAN ANOMALY                                            C1727800
C           ECE = ECCENTRIC ANOMALY                                       C1727900
C           TVE = TRUE ANOMALY                                            C1728000
C           BTO = MAGNITUDE OF CE(I,0), (L)                               C1728100
C                                                                         C1728200
C                                                                         C1728300
C         C     ASM = USER INPUT                                          C1728400
C         C     GEV = USER INPUT                                          C1728500
C               ETE = 1./SQRT(ASM**3/GEV)                                 C1728600
C         C     TPP = USER INPUT                                          C1728700
C               AME = ETE*(T-TPP)                                         C1728800
C               SM1 = SIN(AME)     ; SM4 = SIN(4*AME)                     C1728900
C               SM2 = SIN(2*AME)                                          C1729000
C               SM3 = SIN(3*AME)                                          C1729100
C         C     ECC = USER INPUT                                          C1729200
C               ECE = AME + ECC*SM1 + ECC**2*SM2/2                        C1729300
C              *      +ECC**3*(9*SM3 - 3*SM1)/(24)                        C1729400
C              *      +ECC**4*(64*SM4 - 32*SM2)/192                       C1729500
C               CE = COS(ECE)                                             C1729600
C               SE = SIN(ECE)                                             C1729700
C               BTO = ASM*(1 - ECC*CE)                                    C1729800
C               CB(1,0) = 0                                               C1729900
```

```
C               CB(2,0) = BTO*((CE- ECC)/(1 - ECC*CE))                          C1730000
C               CB(3,0) = BTO*(SQRT(1-ECC**2)*SE/(1-ECC*CE))                    C1730100
C               CALL VECTRN(CE(1,0),XMC(1,1,0),CBC(1,0))                        C1730200
C                                                                               C1730300
C                                                                               C1730400
C                                                                               C1730500
C                                                                               C1730600
C                                                                               C1730700
C                     REACTION TORQUES ON ORBITING SYSTEM DUE TO                C1730800
C                          GRAVITY GRADIENT EFFECTS                             C1730900
C     LET:                                                                      C1731000
C    .CBC(I,1) = COMPONENTS OF VECTOR FROM COMPOSITE SYSTEM CENTER OF           C1731100
C                MASS TO CENTER OF MASS OF BODY 1, RELATIVE TO                  C1731200
C                COMPUTING FRAME                                                C1731300
C     * NOTE FOR GRAVITY GRADIENT OPTION                                        C1731400
C     * CE(I,1) AND ITS INERTIAL DERIVATIVE                                     C1731500
C     * ARE REDEFINED TO CIRCUMVENT DIFFERENCE                                  C1731600
C     * OF LARGE NUMBER PROBLEMS, THAT IS                                       C1731700
C     * THEY ARE MEASURED FROM COMPOSITE CM TO                                  C1731800
C     * CM OF BODY 1 RATHER THAN FROM INERTIAL                                  C1731900
C     * ORIGIN TO CM OF BODY 1                                                  C1732000
C       BH(I) = UNIT VECTOR FROM EARTH'S CENTER TO SYSTEM COMPOSITE             C1732100
C               CENTER OF MASS, COMPONENTS RELATIVE TO INERTIAL FRAME           C1732200
C     CEL(I,K) = COMPONENTS OF VECTOR FROM COMPOSITE SYSTEM CENTER OF           C1732300
C                MASS TO CENTER OF MASS OF BODY K                               C1732400
C     DFG(I,K) = COMPONENTS OF GRAVITY GRADIENT FORCE ACTING ON BODY K          C1732500
C      SGG(I) = COMPACTED INTEGER WORD, THOSE BODIES IN THE NEST I              C1732600
C               WHICH SIGNIFICANTLY CONTRIBUTE TO GRAVITY GRADIENT EFF.         C1732700
C          BTO = DISTANCE FROM EARTH'S CENTER TO COMPOSITE SYSTEM CM             C1732800
C                                                                               C1732900
C               DIMENSION DEL(3,10),DFG(3,10),BHC(3),BH(3),TEM(3),TEM1(3)       C1733000
C               INTEGER SGG(0,9), SI(10)                                        C1733100
C       C       KEPLERIAN ORBIT MUST BE USED WITH GRAVITY GRADIENT OPTION       C1733200
C               DO 10  I=1,3                                                    C1733300
C            10 BHC(I) = CBC(I,0)/BTO                                           C1733400
C               DO 7  K=1,NBOD                                                  C1733500
C               KL = KTO(NB1,0,K)                                               C1733600
C               CALL VECADD(CBC(1,1),GAM(1,KL),DEL(1,K))                        C1733700
C               CALL VECDOT(BHC,DEL(1,K),A)                                     C1733800
C               A = 3*A                                                         C1733900
C               CALL SCLV(A,BHC,TEM)                                            C1734000
C               CALL VECSUB(DEL(1,K),TEM,TEM)                                   C1734100
C               A = -GEV*XMAS(K)/BTO**3                                         C1734200
C               CALL SCLV(A,TEM,DFG(1,K))                                       C1734300
C             7 CONTINUE                                                        C1734400
C               DO 8  K=1,NBOD                                                  C1734500
C               IF(RBLO(K)) GO TO 9                                             C1734600
C               CALL VECADD(PHI(1,K),DFG(1,K),PHI(1,K))                         C1734700
C               GO TO 8                                                         C1734800
C       C       SGG(I) = SK(I) IF ALL BODIES CONTRIBUTE TO GRAVITY GRAD.        C1734900
C       C                      EFFECTS. IF NOT USE COMPAC TO CONSTRUCT          C1735000
C       C                      SGG(I) FORM USER INPUT OR DEFINE DIRECTLY        C1735100
C             9 CALL UNPAC(SI,NS1,SGG(K-1))                                     C1735200
C               DO 8  LL=1,NS1                                                  C1735300
C               L = SI(LL)                                                      C1735400
C               KL = KTO(NB1,K-1,L)                                             C1735500
C               CALL VECRCS (GAM(1,KL),DFG(1,L),TEM)                            C1735600
C               CALL VXDYCV(BHC,XIC(1,1,L),TEM1)                                C1735700
C               A = 3*GEV/BTO**3                                                C1735800
C               CALL SCLV(A,TEM1,TEM1)                                          C1735900
```

```
C                CALL VECACC(TEM,TEM1,TEM)                                C1736000
C                CALL VECADD(PHI(1,K),TEM,PHI(1,K))                       C1736100
C              8 CONTINUE                                                 C1736200
C                                                                         C1736300
C                                                                         C1736400
C                                                                         C1736500
C                                                                         C1736600
C                                                                         C1736700
C                                                                         C1736800
C                    PARAMETERS DEFINED BY FIRST ORDER                    C1736900
C                      DIFFERENTIAL EQUATIONS                             C1737000
C     LET:                                                                C1737100
C          NTQ = TOTAL NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS TO    C1737200
C                BE SOLVED FOR USE IN SUBROUTINE TORQUE                   C1737300
C         TQ(N) = MAGNITUDE OF PARAMETER NUMBER N DEFINED WITHIN SUB.     C1737400
C                TORQUE AT TIME T                                         C1737500
C        TQD(N) = TIME DERIVATIVE OF PARAMETER TQ(N). A USER DEFINED      C1737600
C                FUNCTION OF THE SYSTEM'S STATE VARIABLES                 C1737700
C                                                                         C1737800
C                      DIMENSION TQ(20),TQD(20)                           C1737900
C                C     FOR THE PARAMETER N                                C1738000
C                      IF(CT4.NE.1) GO TO 11                              C1738100
C                C     Y(NEQ+N) = TQ(N) = INITIAL VALUE FOR TQ(N)         C1738200
C                   11 TQ(N) = Y(NEQ+N)                                   C1738300
C                C     TQD(N) = USER DEFINED FUNCTION OF STATE VARB.      C1738400
C                C                                                        C1738500
C                C     AFTER DEFINITION OF LAST DIFFERENTIAL EQUATION     C1738600
C                C     NTQ = TOTAL NUMBER OF FIRST ORDER DIFFERENTIAL     C1738700
C                C           EQUATIONS TO BE SOLVED FOR USE IN TORQUE     C1738800
C                      DO 12  N=1,NTQ                                     C1738900
C                   12 YD(NEQ+N) = TQD(N)                                 C1739000
C                                                                         C1739100
C                                                                         C1739200
C                                                                         C1739300
C                                                                         C1739400
C                                                                         C1739500
C                    THERMALLY INDUCED MOTION ABOUT GIMBAL AXIS M         C1739600
C                      AT HINGE POINT K-1                                 C1739700
C     ASSUME:                                                             C1739800
C          ALL THERMALLY INDUCED DEFLECTION IS SMALL ANGLE                C1739900
C             RELATIVE TO THE SYSTEM'S NOMINAL ZERO STRESS STATE          C1740000
C          THERMALLY INDUCED DEFLECTION IS MODELLED AS A MOVEMENT         C1740100
C             OF THE ZERO STRESS STATE                                    C1740200
C          ACROSS ALL HINGE POINTS SUBJECT TO THERMAL DEFORMATION         C1740300
C             SPRINGS AND DAMPERS ACT                                     C1740400
C          A REASONABLE MODEL OF THE THERMAL INPUT CAN BE DEFINED         C1740500
C             IN TERMS OF THE SYSTEM'S STATE VARIABLES                    C1740600
C          THERMAL EQUILIBRIUM POSITION ABOUT ANY GIMBAL AXIS IS DEF.     C1740700
C             BY SOLUTION OF THE HEAT CONDUCTION EQUATION                 C1740800
C     LET:                                                                C1740900
C          SPR(M) = SPRING CONSTANT ACROSS GIMBAL AXIS M                  C1741000
C          DPG(M) = DAMPING CONSTANT ACROSS GIMBAL AXIS M                 C1741100
C          TAU(M) = THERMAL TIME CONSTANT FOR DEFORMATION ABOUT GIMBAL    C1741200
C                   AXIS M, (T)                                           C1741300
C          TQ(N) = THERMAL EQUILIBRIUM POSITION FOR THERMAL DEFORMATION   C1741400
C                  ABOUT GIMBAL AXIS M, (RAD)                             C1741500
C         TQD(N) = RATE OF CHANGE OF THERMAL EQUILIBRIUM POSITION ABOUT   C1741600
C                  GIMBAL AXIS M, FIRST ORDER DIFF. EQ., (RAD/T)          C1741700
C          TINP = THERMAL INPUT USER DEFINED FUNCTION OF STATE VARIABLES. C1741800
C                 (RAD/T)                                                 C1741900
```

```
C                                                                       C1742000
C           DIMENSION TEM(3)                                            C1742100
C       C   N = USER DEFINED LABEL, DEPENDS UPON EQUATION NUMBERING     C1742200
C       C       SEQUENCE DEFINED WITHIN SUBROUTINE TORQUE               C1742300
C           IF(CT4.NE.1) GO TO 13                                       C1742400
C       C   Y(NEQ+N) = TQ(N) = INITIAL VALUE FOR THERMAL DEFORMATION    C1742500
C       C                      ABOUT GIMBAL AXIS M, USER INPUT          C1742600
C        13 TQ(N) = Y(NEQ+N)                                            C1742700
C       C   TINP = USER DEFINED THERMAL INPUT FOR THERMAL DEFORMATION   C1742800
C       C          ABOUT GIMBAL AXIS M                                  C1742900
C       C   TAU(M) = USER INPUT                                         C1743000
C           TQD(N) = -TQ(N)/TAU(M) + TINP                               C1743100
C           A = SPR(M)*(THA(M) - TQ(N))                                 C1743200
C           CALL SCLV(A,QFC(1,M),TEM)                                   C1743300
C           CALL VECSUB(PHI(1,K),TEM,PHI(1,K))                          C1743400
C           A = DPC(M)*THAD(M)                                          C1743500
C           CALL SCLV(A,QFC(1,M),TEM)                                   C1743600
C           CALL VECSUB(PHI(1,K),TEM,PHI(1,K))                          C1743700
C                                                                       C1743800
C                                                                       C1743900
C                                                                       C1744000
C                                                                       C1744100
C                                                                       C1744200
C                                                                       C1744300
C                                                                       C1744400
C     ZERO ALL ELEMENTS OF EXTERNAL TORQUE MATRIX                       C1744500
      DO 1  K=1,NB1                                                     C1744600
      DO 1  I=1,3                                                       C1744700
    1 PHI(I,K) = 0.D0                                                   C1744800
      DO 2 M=1,NMO                                                      C1744900
    2 CLM(M) = 0.D0                                                     C1745000
C                                                                       C1745100
      RETURN                                                            C1745200
      END                                                               C1745300
```

```
C                                                                       C1800000
      SUBROUTINE QFDOT                                                  C1800100
C     USED TO REDUCE THE SET OF NBOD+1+NMV VECTOR DYADIC EQUATIONS OF   C1800200
C     MOTION TO NFER+NMV SCALAR EQUATIONS                               C1800300
C                                                                       C1800400
C     LET                                                               C1800500
C        SGF(K) = LOWEST MAGNITUDE FREE COORDINATE INDICE AT HINGE      C1800600
C                 POINT K-1, EQUALS ZERO IF THREE CONSTRAINTED AXES     C1800700
C        SGL(K) = LOWEST MAGNITUDE LOCKED COORDINATE INDICE AT HINGE    C1800800
C                 POINT K-1, EQUALS ZERO IF THREE FREE AXES             C1800900
C        SGK(K) = BODY LABELS ON PATH FROM HINGE POINT ZERO TO C.M.     C1801000
C                 OF BODY K, FOR K=NB1 IT IS SET OF ALL BODY LABELS     C1801100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1801200
C                                                                       C1801300
C                                                                       C1801400
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C1801500
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1801600
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1801700
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1801800
C                                                                       C1801900
C                                                                       C1802000
```

```
      INTEGER                                                           C1802100
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , PCON   , C1802200
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     , C1802300
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    , C1802400
     * SMA    , SMCDUM, SMC    , SMV    , SOK   , SPIDUM, SPI    , SQF    , C1802500
     * SGL    , SR     , SSCN   , SSIX   , SVA   , SVB   , SVD    , SVI    , C1802600
     * SVM    , SVP    , SVQ    , SXM    , SXT   , TORQ   , SMAL   , SEU    , C1802700
     * SC     , SCG    , NFLXB  , SFLX   , SFXM  , NMODS  , SFCC   , SCC    , C1802800
     * IINIT(1)        , IZINIT(1)       , SD    , SCXC(20)                  C1802900
C                                                                       C1803000
C                                                                       C1803100
      REAL*8                                                            C1803200
     * ANGE   (33)     , CNF    (3,10)  , ETIC   (3,10)  , ETMC   (3,10) , C1803300
     * FLQ    (3,20)   , FLE    (3,3,20), FLH    (3,3,20),               C1803400
     * THACD  (33)     , YMCD   (3,2,11), RINIT  (1)     , RZINIT(1)     C1803500
C                                                                       C1803600
C                                                                       C1803700
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,      C1803800
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,      C1803900
     *                LGFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C1804000
C                                                                       C1804100
C                                                                       C1804200
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLJ(10)           C1804300
C                                                                       C1804400
C                                                                       C1804500
      COMMON /INTG/  AWORK(200) ,                                       C1804600
     * CT1            , CT2            , CT3            , CT4          , C1804700
     * CT5            , FCON   (33)    , JCON   (10)    , LCON   (22)  , C1804800
     * MC     (10)    , NEI            , NBOD           , NCTC         , C1804900
     * NFER           , NFKC           , NFRC           , NLOR         , C1805000
     * NMV            , NMC            , NMOA           , NSVP         , C1805100
     * NSVC           , PCON   (11)    , SD             , SFR    (33)  , C1805200
     * SG             , SI     (55)    , SIG            , SL           , C1805300
     * SLK    (33)    , SMA    (10)    , SOK    (11)    , SQF    (11)  , C1805400
     * SGL    (11)    , SMV            , SR             , SSCN         , C1805500
     * SSIX           , SVA            , SVB            , SVD          , C1805600
     * SVI            , SVM            , SVP    (22)    , SVQ    (33)  , C1805700
     * SXM    (3,10)  , SXT            , TORQ   (97)    , SMAL         , C1805800
     * SEU            , NTQ            , SC     (33)    , SCG          , C1805900
     * NFLXB          , SFLX           , SFXM   (10)    , NMODS        , C1806000
     * SFCC           , SCC    (10)                                      C1806100
C                                                                       C1806200
C                                                                       C1806300
      COMMON /INTGZ/                                                    C1806400
     * SCNDUM         , SCN    (9)     , SCRDUM         , SCR    (9)   , C1806500
     * SFKDUM         , SFK    (9)     , SIXDUM         , SIX    (9)   , C1806600
     * SKDUM          , SK     (9)     , SPIDUM         , SPI    (9)   , C1806700
     * SMCDUM         , SMC    (9)                                      C1806800
C                                                                       C1806900
C                                                                       C1807000
      COMMON /REAL/                                                     C1807100
     * CA     (3,10)  , CAC    (3,10)  , CLM    (10)    , CLMC   (3,11) , C1807200
     * DCMC   (3,11)  , ETC    (3,11)  , ETM    (33)    , FDMC   (3,11) , C1807300
     * GAM    (3,66)  , H              , HM     (3,10)  , HMC    (3,10) , C1807400
     * HNOM   (10)    , PHI    (3,11)  , PLM    (10)    , QF     (3,33) , C1807500
     * QFC    (3,33)  , QL     (3,22)  , QLC    (3,22)  , RDMC   (3,11) , C1807600
     * T              ,                 THA    (33)    , THAD   (33)    , C1807700
     * THADW  (10)    , THAW   (10)    , XDIC   (3,3,66), XI     (3,3,10), C1807800
     * XIC    (3,3,10), XMAS   (10)    , XMN    (33,33) , XMT    (3,3,10), C1807900
     * TLG    (33)    , FLA    (3,20)  , FLB    (3,20)  , FLC    (3,20)  , C1808000
```

```
     * FLD   (3,3,20), FLJ   (3,3,20), CAC   (3,10)  , XIO   (3,3,10),   C1808100
     * FLIRC (3,10)  , FLCRC (3,10)  , FLAC  (3,20)  , FLQC  (3,20)  ,   C1808200
     * FLCN  (20)    , ZETA  (20)    , FCF   (3,3,40), FCK   (3,40)  ,   C1808300
     * TIMEND                                                            C1808400
C                                                                        C1808500
C                                                                        C1808600
      COMMCN /REALZ/                                                     C1808700
     * CEDUM (1,3)   , CE    (3,10)  , CBCDUM(1,3)   , CBC   (3,10)  ,   C1808800
     * XMCCUM(1,1,9) , XMC   (3,3,10), CBN(3)                           C1808900
C                                                                        C1809000
C                                                                        C1809100
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))     ,   C1809200
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNF(1,1))    ,   C1809300
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))  ,   C1809400
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1)),   C1809500
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                           C1809600
     *            (FG1,LINIT(1))            ,(CA(1,1),RINIT(1))     ,   C1809700
     *            (CBDUM(1,1),RZINIT(1))    ,(AWORK(1),IINIT(1))    ,   C1809800
     *            (SCNDUM,IZINIT(1))        ,(TORG(73),SCXC(1))         C1809900
C                                                                        C1810000
C                                                                        C1810100
C                                                                        C1810200
C                                                                        C1810300
      DIMENSION TEM1(3),TEM(3,11),XQD(3,11)                              C1810400
      DIMENSION TEM2(3),TEM3(3),FTEM(3),FLHC(3,3)                        C1810500
      DIMENSION EQFC(99), EXDIC(554)                                     C1810600
      REAL*8    FCUP(3,3,20),FCUP1(3,3),KCUP(3,20),KCUP1(3)              C1810700
      REAL*8    TMF(3,3), TMK(3)                                         C1810800
      INTEGER ST1(10),ST2(10),ST3(11)                                    C1810900
      INTEGER ST4(10)                                                    C1811000
      INTEGER SFXMN                                                      C1811100
      LOGICAL CTAIN                                                      C1811200
      EQUIVALENCE (DCMC(1,1),TEM(1,1)), (EQFC(1),QFC(1,1)),             C1811300
     *            (EXDIC(1),XDIC(1,1,1))                                 C1811400
      DATA IH/4HXMN(/                                                    C1811500
      EQUIVALENCE (LQFDCT,LEQU)                                          C1811600
C                                                                        C1811700
C                                                                        C1811800
      IF(LEQU)PRINT 226                                                  C1811900
C     ZERO ALL ELEMENTS IN XMN MATRIX                                    C1812000
      NTERM = NFER + NMV + NMODS                                         C1812100
      DO 18 N=1,NTERM                                                    C1812200
      DO 18 M=N,NTERM                                                    C1812300
      XMN(M,N) = 0                                                       C1812400
   18 CONTINUE                                                           C1812500
      IF(LEQU)PRINT 214                                                  C1812600
C                                                                        C1812700
C     CYCLE THROUGH ALL ROWS OF SYSTEM INERTIA MATRIX                    C1812800
C     COMPUTE (NFER)X(NFER) MATRIX OF SCALARS LEFT  HAND SIDE OF SYSTEM  C1812900
C        MATRIX EQUATION OF MOTION                                       C1813000
      DO 7  K=1,NB1                                                      C1813100
C     CHECK FOR 3 CONSTRAINED AXES                                       C1813200
      IF(PCON(K).NE.3) GO TO 21                                          C1813300
      IF(LEQU)PRINT 227                                                  C1813400
      IF(LEQU)PRINT 202, K                                               C1813500
      IF(LEQU)PRINT 227                                                  C1813600
      GO TO 7                                                            C1813700
C                                                                        C1813800
   21 CONTINUE                                                           C1813900
      CALL UNPAC(ST3,NST3,SGK(K))                                        C1814000
```

```
C      GET NON-ZERO ELEMENTS IN ROW K OF XDIC                          C1814100
       IF(LEQU)PRINT 227                                               C1814200
       IF(LEQU)PRINT 205, K,(ST3(I),I=1,NST3)                          C1814300
       IF(LEQU)PRINT 228                                               C1814400
       IF(NST3.EQ.0) GO TO 41                                          C1814500
       DO 40 II=1,NST3                                                 C1814600
       I = ST3(II)                                                     C1814700
       MBEG=SQF(K)                                                     C1814800
       NBEG=SQF(I)                                                     C1814900
       MTERM=SQF(K)+2-PCCN(K)                                          C1815000
       NTERM=SQF(I)+2-PCCN(I)                                          C1815100
       IF(MBEG.GT.MTERM) GO TO 43                                      C1815200
       DO 42 M=MBEG,MTERM                                              C1815300
       IF(NBEG.GT.NTERM) GO TO 45                                      C1815400
       DO 44 N=NBEG,NTERM                                              C1815500
       IF(N.GT.M) GO TO 44                                             C1815600
       KI = KT1(NB1,K,I)                                               C1815700
       ME = 3*(M-1)                                                    C1815800
       NE = 3*(N-1)                                                    C1815900
       KIE = 9*(KI-1)                                                  C1816000
C      CALL VUDYUV(QFC(1,M),XDIC(1,1,KI),QFC(1,N),XMN(M,N))            C1816100
C      REPLACE MULTI-SUBSCRIPT OPERATION WITH SINGLE SUBSCRIPT         C1816200
C       COMPUTATION OF VECTOR DOT DYAD DOT VECTOR                      C1816300
       DO 10  J=1,3                                                    C1816400
       TEM1(J) = EXDIC(KIE+J)*EQFC(NE+1) +                             C1816500
     *           EXDIC(KIE+3+J)*EQFC(NE+2) +                           C1816600
     *           EXDIC(KIE+6+J)*EQFC(NE+3)                             C1816700
    10 CONTINUE                                                        C1816800
       XMN(M,N) = EQFC(ME+1)*TEM1(1) +                                 C1816900
     *            EQFC(ME+2)*TEM1(2) +                                 C1817000
     *            EQFC(ME+3)*TEM1(3)                                   C1817100
       IF(LEQU)PRINT 206, M,N,M,KI,N,M,K,I,N,XMN(M,N)                  C1817200
    44 CONTINUE                                                        C1817300
    45 CONTINUE                                                        C1817400
    42 CONTINUE                                                        C1817500
    43 CONTINUE                                                        C1817600
    40 CONTINUE                                                        C1817700
    41 CONTINUE                                                        C1817800
     7 CONTINUE                                                        C1817900
C                                                                      C1818000
C                                                                      C1818100
C                                                                      C1818200
C      INCLUDE FLEXIBILITY TERMS                                       C1818300
       IF(NFLXB.EQ.0) GO TO 11                                         C1818400
C                                                                      C1818500
       CALL UNPAC(ST1,NST1,SFLX)                                       C1818600
       CALL UNPAC(ST4,NST4,SFCC)                                       C1818700
C                                                                      C1818800
C      ZERO OUT GYROSCOPIC TORQUE ARRAY ETM(M),M=NFER+1,...,NFER+NMODS C1818900
       NF1 = NFER + 1                                                  C1819000
       NF2 = NFER + NMODS                                              C1819100
       DO 17 M=NF1,NF2                                                 C1819200
       IF(LEQU) PRINT 252, M                                           C1819300
    17 ETM(M) = 0.                                                     C1819400
C                                                                      C1819500
C      CYCLE THRU ALL NESTS, FOR K=1 PICK UP DIAGONAL AND MOST OFF     C1819600
C      RIGHT HAND SIDE OF EQUATION                                     C1819700
       DO 3 K=1,NBOD                                                   C1819800
       CALL UNPAC(ST2,NST2,SK(K-1))                                    C1819900
       MN=0                                                            C1820000
```

```
      DO 3 NN=1,NST1                                                    C1E20100
      N = ST1(NST1+1-NN)                                                C1E20200
C     CHECK IS BODY N A FLEXIBILE BODY IN NEST K-1                      C1E20300
      IF(CTAIN(N,ST2,NST2)) GO TO 4                                     C1E20400
      MN = MN+SFXM(N)                                                   C1E20500
      GO TO 3                                                           C1E20600
    4 CONTINUE                                                          C1E20700
      IF(.NOT.LEQU) GO TO 5000                                          C1E20800
      KM1 = K-1                                                         C1E20900
      PRINT 224, N,KM1                                                  C1E21000
 5000 CONTINUE                                                          C1821100
C                                                                       C1821200
C     FOR K=1 GET: 1)DIAGONAL ELEMENTS OF XMN(M,N)                      C1821300
C                  2)CENTRIPITAL ACC. OF UNDEFORMED CM OF BODY N        C1821400
C                  3)MAJOR PORTION OF ETM(M)                            C1821500
C                     M=NFER+1,....,NFER+NMODS                          C1821600
C                                                                       C1821700
      IF(K.NE.1) GO TO 5                                                C1821800
      TEM2(1) = 0.                                                      C1821900
      TEM2(2) = 0.                                                      C1822000
      TEM2(3) = 0.                                                      C1822100
      IF(LEQU) PRINT 241                                                C1822200
C     CENTRIPITAL ACCELERATION UNDEFORMED POSITION OF BODY N CM.        C1822300
      NI = N                                                            C1822400
      IF(NI.EQ.1) GO TO 12                                              C1822500
      CALL VECTRN(CAD(1,NI),XMC(1,1,NI),TEM3)                           C1822600
      IF(LEQU) PRINT 211, NI,NI,NI                                      C1822700
      CALL TRIPVP(FOMC(1,NI),TEM3,TEM2)                                 C1822800
      IF(LEQU) PRINT 212, NI,NI,NI                                      C1822900
   32 JNI = JCON(NI)                                                    C1823000
      CALL TRIPVP(FOMC(1,JNI),CBC(1,NI),TEM3)                           C1823100
      CALL VECADD(TEM2,TEM3,TEM2)                                       C1823200
      IF(LEQU) PRINT 213, JNI,JNI,NI                                    C1823300
      NI = JNI                                                          C1823400
      IF(NI.NE.1) GO TO 32                                              C1823500
   12 CALL SCLV(XMAS(N),TEM2,TEM2)                                      C1823600
      IF(LEQU) PRINT 222, N,(TEM2(J),J=1,3)                             C1823700
    5 CONTINUE                                                          C1823800
C                                                                       C1823900
C                                                                       C1824000
C                                                                       C1824100
C     CYCLE THRU ALL FLEXIBLE BODY MODES ASSOCIATED WITH BODY N IN      C1824200
C      THE NEST K-1, FOR K=1 DO EXTRA COMPUTATIONS                      C1824300
C                                                                       C1824400
      SFXMN = SFXM(N)                                                   C1824500
      DO 3 II=1,SFXMN                                                   C1824600
      MN = MN+1                                                         C1824700
      M = NFER + MN                                                     C1824800
C     COMPUTE VECTOR ELEMENTS OF F MATRIX AS NEEDED DON'T STORE IN      C1824900
C     COMMON, STORE AS NEEDED IN FTEM TO SAVE STORAGE                   C1825000
      KN = KTC(NB1,K-1,N)                                               C1825100
      CALL SCLV(XMAS(N),GAM(1,KN),TEM3)                                 C1825200
      CALL VECROS(TEM3,FLAC(1,MN),FTEM)                                 C1825300
      CALL VECADD(FTEM,FLQC(1,MN),FTEM)                                 C1825400
      IF(.NOT.LEQU) GO TO 5001                                          C1825500
      KM1 = K-1                                                         C1825600
      PRINT 235, KM1,N,MN,N,KM1,N,MN,MN,(FTEM(J),J=1,3)                 C1825700
 5001 CONTINUE                                                          C1825800
      IF(K.NE.1)GO TO 19                                                C1825900
      CALL TENTRN(FLH(1,1,MN),XMC(1,1,N),FLHC)                          C1826000
```

```
      IF(.NOT.LEQU) GO TO 5002                                          C1826100
      PRINT 228                                                         C1826200
      PRINT 250,            (FLFC(1,I),I=1,3)                           C1826300
      PRINT 253, MN,N,MN,N,(FLFC(2,I),I=1,3)                           C1826400
      PRINT 250,            (FLFC(3,I),I=1,3)                           C1826500
 5002 CONTINUE                                                          C1826600
   19 CONTINUE                                                          C1826700
C                                                                       C1826800
C     COMPLTE ELEMENTS OF XMN ARRAY                                     C1826900
C                                                                       C1827000
      IF(LEQU) PRINT 228                                                C1827100
      IF(PCON(K).EQ.3) GO TO 46                                         C1827200
      N1 = SOF(K)                                                       C1827300
      N2 = SOF(K) + 2 - PCON(K)                                         C1827400
      DO 6 L=N1,N2                                                      C1827500
      CALL VECCOT(QFC(1,L),FTEM,XMN(M,L))                               C1827600
      IF(LEQU) PRINT 236, M,L,L,KM1,N,MN,XMN(M,L)                       C1827700
    6 CONTINUE                                                          C1827800
   46 CONTINUE                                                          C1827900
C                                                                       C1828000
C     THAT'S IT FOR XMN IF K.NE.1                                       C1828100
      IF(K.NE.1) GO TO 8                                                C1828200
      XMN(M,M) = XMAS(N)                                                C1828300
      IF(LEQU) PRINT 237, M,M,N,XMN(M,M)                                C1828400
C     GET TERMS ASSOCIATED WITH THE TRANSLATION EQUATION                C1828500
      CALL SCLV(XMAS(N),FLAC(1,MN),TEM3)                                C1828600
      IF(PCON(NB1).EQ.3) GO TO 8                                        C1828700
      N1 = SOF(NB1)                                                     C1828800
      N2 = SOF(NB1) + 2 - PCON(NB1)                                     C1828900
      DO 13 L=N1,N2                                                     C1829000
      CALL VECCOT(QFC(1,L),TEM3,XMN(M,L))                               C1829100
      IF(LEQU) PRINT 215, M,L,L,N,MN,XMN(M,L)                           C1829200
   13 CONTINUE                                                          C1829300
    8 CONTINUE                                                          C1829400
C                                                                       C1829500
C     OK CONCENTRATE ON ETM ARRAY NOW                                   C1829600
C      PUT IN FTEM,QFC(DOT) TERMS                                       C1829700
      CALL VECCOT(FTEM,QFMC(1,K),A)                                     C1829800
      ETM(M) = ETM(M) - A                                               C1829900
      IF(LEQU) PRINT 238, M,M,KM1,N,MN,K,ETM(M)                         C1830000
C     THAT'S IT IF K.NE.1                                               C1830100
      IF(K.NE.1) GO TO 3                                                C1830200
C                                                                       C1830300
C     PUT IN SPRING-DASHPOT EFFECTS MODE MM                             C1830400
      ETM(M) = ETM(M) - XMAS(N)*(2.*ZETA(MN)*FLUM(MN)*THAD(M)           C1830500
     *                           + FLUM(MN)**2*THA(M))                  C1830600
      IF(LEQU) PRINT 239, M,M,N,MM,MN,M,MN,M,ETM(M)                     C1830700
      CALL VECDOV(FLMC(1,N),FLFC,FOMC(1,N),A)                           C1830800
      CALL VECCOT(TEM2,FLAC(1,MN),B)                                    C1830900
      ETM(M) = ETM(M) + A - B                                           C1831000
      IF(LEQU) PRINT 240, M,M,N,MN,N,MN,ETM(M)                          C1831100
    3 CONTINUE                                                          C1831200
C                                                                       C1831300
C                                                                       C1831400
C     CHECK TO SEE IF COUPLING SIGNIFICANT                              C1831500
      IF(NST4.EQ.0) GO TO 11                                            C1831600
      KF = 0                                                            C1831700
      MN = 0                                                            C1831800
      DO 35 K=1,NBOD                                                    C1831900
C     CHECK TO SEE IF BODY K FLEXIBLE                                   C1832000
```

```
      IF(SFXM(K).EQ.0) GO TO 35                                          C1832100
C     CYCLE THROUGH ALL GENERALIZED COORDINATE EQUATIONS FOR BODY K      C1832200
      MMN = MN                                                           C1832300
      SFXMN = SFXM(K)                                                    C1832400
      DO 33 N=1,SFXMN                                                    C1832500
      MN = MN+1                                                          C1832600
      CALL UNPAC(ST1,NST1,SCXC(MN))                                      C1832700
      IF(NST1.EQ.0) GO TO 33                                             C1832800
      IF(LEQU) PRINT 256, N,K,MN,(ST1(I),I=1,NST1)                       C1832900
      MI = NFER+MN                                                       C1833000
      DO 37 I=1,3                                                        C1833100
      KCUP(I,MN) = 0.0                                                   C1833200
      DO 37 J=1,3                                                        C1833300
   37 FCUP(I,J,MN) = 0.0                                                 C1833400
      IF(LEQU) PRINT 245                                                 C1833500
      IF(LEQU) PRINT 242, MN,MMN                                         C1833600
      DO 34 I=1,NST1                                                     C1833700
      M = ST1(NST1+1-I)                                                  C1833800
      MJ = MMN + M + NFER                                                C1833900
      KF = KF+1                                                          C1834000
      CALL SCLD(THA(MJ),FCF(1,1,KF),TMF)                                 C1834100
      CALL DYADD(FCUP(1,1,MN),TMF,FCUP(1,1,MN))                          C1834200
      IF(.NOT.LEQU) GO TO 5004                                           C1834300
      PRINT 245                                                          C1834400
      PRINT 246,                (FCUP(1,L,MN),L=1,3)                     C1834500
      PRINT 247, MN,MN,MJ,KF,(FCUP(2,L,MN),L=1,3)                        C1834600
      PRINT 246,                (FCUP(3,L,MN),L=1,3)                     C1834700
 5004 CALL SCLV(THAD(MJ),FCK(1,KF),TMK)                                  C1834800
      CALL VECADD(KCUP(1,MN),TMK,KCUP(1,MN))                             C1834900
      IF(.NOT.LEQU) GO TO 5005                                           C1835000
      PRINT 245                                                          C1835100
      PRINT 246, MN,MN,MJ,KF,(KCUP(L,MN),L=1,3)                          C1835200
 5005 CONTINUE                                                           C1835300
   34 CONTINUE                                                           C1835400
C     TRANSFORM FCUP AND KCUP TO COMPUTING FRAME                         C1835500
      CALL TENTRN(FCUP(1,1,MN),XMC(1,1,K),FCUP1)                         C1835600
      CALL VECTRN(KCUP(1,MN),XMC(1,1,K),KCUP1)                           C1835700
      IF(.NOT.LEQU) GO TO 5003                                           C1835800
      PRINT 245                                                          C1835900
      PRINT 254,          (FCUP1(1,L),L=1,3)                             C1836000
      PRINT 257, K,MN,K,(FCUP1(2,L),L=1,3)                               C1836100
      PRINT 254,          (FCUP1(3,L),L=1,3)                             C1836200
      PRINT 245                                                          C1836300
      PRINT 255, K,MN,(KCUP1(L),L=1,3)                                   C1836400
 5003 CONTINUE                                                           C1836500
C                                                                        C1836600
      CALL VCDYDV(FOMC(1,K),FCUP1,FCMC(1,K),A)                           C1836700
      CALL VECDOT(FOMC(1,K),KCUP1,B)                                     C1836800
      ETM(MI) = ETM(MI) + A - 2.0*B                                      C1836900
      IF(.NOT.LEQU) GO TO 5007                                           C1837000
      PRINT 245                                                          C1837100
      PRINT 249,MI,MI,K,K,K,ETM(MI)                                      C1837200
 5007 CONTINUE                                                           C1837300
   33 CONTINUE                                                           C1837400
   35 CONTINUE                                                           C1837500
   11 CONTINUE                                                           C1837600
C     THAT'S IT FOR MODAL COUPLING                                       C1837700
C                                                                        C1837800
C                                                                        C1837900
C     COMPUTATION FOR VARIABLE SPEED MOMENTUM WHEELS                     C1838000
```

```
C      EXPAND DIMENSION OF MATRIX EQUATIONS AND COMPUTE SCALAR ELEMENTS C1838100
C      ASSOCIATED WITH PRESENCE OF MOMENTUM WHEELS                      C1838200
       CALL UNPAC(ST1,NST1,SMV)                                         C1838300
       IF(NST1.EQ.0) GO TO 5057                                         C1838400
       IF(LEQU)PRINT 201,(ST1(I),I=1,NST1)                              C1838500
       DO 2 MMMM=1,NST1                                                 C1838600
       NNN=NST1-(MMMM-1)                                                C1838700
       MM = ST1(MMMM)                                                   C1838800
       M = NFER+NST1+1-MMM+NMODS                                        C1838900
       XMN(M,M) = PLM(MM)                                               C1839000
       IF(LEQU)PRINT 203, M,M,MM,XMN(M,M)                               C1839100
       CALL UNPAC(ST2,NST2,SCR(MC(MM)))                                 C1839200
       ETM(M) = 0.DO                                                    C1839300
       IF(LEQU)PRINT 204, M,ETM(M)                                      C1839400
       IF(NST2.EQ.0) GO TO 5058                                         C1839500
       DO 1  II=1,NST2                                                  C1839600
       I = ST2(II)                                                      C1839700
       CALL VECDOT(HMC(1,MM),DFC(1,I),A)                                C1839800
       ETM(M) = ETM(M) - A*PLM(MM)                                      C1839900
       IF(LEQU)PRINT 207, M,M,MM,MM,I                                   C1840000
       KTERM=SQF(I)+2-PCCM(I)                                           C1840100
       KBEG=SQF(I)                                                      C1840200
       IF(KBEG.GT.KTERM) GO TO 48                                       C1840300
       DO 47 K=KBEG,KTERM                                               C1840400
       N = K                                                            C1840500
       CALL VECDOT(HMC(1,MM),QFC(1,K),A)                                C1840600
       XMN(M,N) = A*PLM(MM)                                             C1840700
       IF(LEQU)PRINT 209, M,N,MM,MM,K,XMN(M,N)                          C1840800
   47 CONTINUE                                                          C1840900
   48 CONTINUE                                                          C1841000
    1 CONTINUE                                                          C1841100
 5058 CONTINUE                                                          C1841200
       ETM(M) = ETM(M) + CLM(MM)                                        C1841300
       IF(LEQU)PRINT 210, M,M,MM,ETM(M)                                 C1841400
    2 CONTINUE                                                          C1841500
 5057 CONTINUE                                                          C1841600
C                                                                       C1841700
C                                                                       C1841800
C                                                                       C1841900
C      FILL IN UPPER TRIANGULAR PORTION                                 C1842000
       NFMV = NFER+NMODS+NMV                                            C1842100
       DO 9 M=1,NFMV                                                    C1842200
       DO 9 N=M,NFMV                                                    C1842300
    9  XMN(M,N) = XMN(N,M)                                              C1842400
       IF(.NOT.LEQU) GO TO 5006                                         C1842500
       PRINT 227                                                        C1842600
        I1 = NFMV/4                                                     C1842700
       I2 = NFMV - 4*I1                                                 C1842800
       I3 = I1 + 1                                                      C1842900
       DO 20 I=1,I3                                                     C1843000
       I4 = 3                                                           C1843100
       IF(I.NE.I3) GO TO 22                                             C1843200
       IF(I2.EQ.0) GO TO 20                                             C1843300
       I4 = I2 - 1                                                      C1843400
   22  I5 = 4*(I-1) + 1                                                 C1843500
       DO 23 N=1,NFMV                                                   C1843600
       I5PI4 = I5 + I4                                                  C1843700
       PRINT 200, (IH,M,N,XMN(M,N),M=I5,I5PI4)                          C1843800
   23 CONTINUE                                                          C1843900
       PRINT 228                                                        C1844000
```

```
   20 CONTINUE                                                          C1844100
      PRINT 227                                                         C1844200
C                                                                       C1844300
C     COMPUTE (NB1)X1 COLUMN MATRIX OF VECTORS RESULTING FROM DIFFENTIAT C1844400
C      VECTORS MOVING RELATIVE TO FRAME OF COMPUTATION                  C1844500
C     DONE BY EQUIVALENCE (DCMC,TEM)                                    C1844600
      PRINT 20E, (K,K,(TEM(I,K),I=1,3),K=1,NB1)                         C1844700
      PRINT 227                                                         C1844800
C                                                                       C1844900
C      MULTIPLY (NB1)X(NB1) INERTIA MATRIX XDIC WITH (NB1)X1 MATRIX TEM C1845000
C      NOTE THAT XDIC STORED IN TRIANGULAR FORM PUT RESULT IN TEM        C1845100
 5006 CONTINUE                                                          C1845200
 2000 CONTINUE                                                          C1845300
      DO 24  K=1,NB1                                                    C1845400
      DO 25  I=1,3                                                      C1845500
   25 XQD(I,K) = 0                                                      C1845600
      IF(LEQU)PRINT 215, K                                              C1845700
      IF(K.NE.NB1) GO TO 30                                             C1845800
      CALL UNPAC(ST1,NST1,SR)                                           C1845900
      NST2 = 0                                                          C1846000
      GO TO 31                                                          C1846100
   30 CALL UNPAC(ST1,NST1,SI(K))                                        C1846200
      CALL UNPAC(ST2,NST2,SK(K-1))                                      C1846300
      ST2(NST2) = NB1                                                   C1846400
   31 CONTINUE                                                          C1846500
C                                                                       C1846600
C     START MATRIX MULTIPLICATION ROW K TO RIGHT OF DIAGONAL            C1846700
      IF(LEQU)PRINT 217,K,(ST2(I),I=1,NST2)                             C1846800
      IF(NST2.EQ.0) GO TO 5059                                          C1846900
      DO 26  II=1,NST2                                                  C1847000
C     ELEMENT K=I DELETE IT WAS TAKEN CARE OF ABOVE                     C1847100
      I = ST2(II)                                                       C1847200
C     CHECK FOR MULTIPLICATION BY ZERO                                  C1847300
C     RECALL FROM RATE THAT DCMC(NB1) =TEM1(NB1) = 0                    C1847400
      IF(I.EQ.NB1) GO TO 26                                             C1847500
      IF(.NOT.RBLO(I)) GO TO 26                                         C1847600
      KI = KT1(NB1,K,I)                                                 C1847700
      IF(LEQU)PRINT 220, K,K,KI,I,K,K,I,I                               C1847800
C      NOTE THAT I,K TENSOR IS THE TRANSPOSE OF THE K,I TENSOR          C1847900
C        THIS HOWEVER DOES NOT INSURE THAT I,K IS SYMMETRIC, IN FACT    C1848000
C        IN GENERAL IT WILL NOT BE SO HENCE DYTDV IS USED               C1848100
      CALL DYTCV  (XDIC(1,1,KI),TEM(1,I),TEM1)                          C1848200
      CALL VECADD(XQD(1,K),TEM1,XQD(1,K))                               C1848300
   26 CONTINUE                                                          C1848400
 5059 CONTINUE                                                          C1848500
      IF(LEQU)PRINT 228                                                 C1848600
      IF(NST1.EQ.0)GO TO 5060                                           C1848700
      IF(LEQU)PRINT 216, K,(ST1(I),I=1,NST1)                            C1848800
      DO 25  II=1,NST1                                                  C1848900
      I = ST1(II)                                                       C1849000
      IF(.NOT.RBLO(I)) GO TO 25                                         C1849100
C      BODY I RIGID BODY                                                C1849200
      KI = KT1(NB1,K,I)                                                 C1849300
      IF(LEQU)PRINT 218, K,K,KI,I,K,K,I,I                               C1849400
      CALL DYDCTV(XDIC(1,1,KI),TEM(1,I),TEM1)                           C1849500
      CALL VECADD(XQD(1,K),TEM1,XQD(1,K))                               C1849600
   25 CONTINUE                                                          C1849700
 5060 CONTINUE                                                          C1849800
      IF(LEQU)PRINT 228                                                 C1849900
C     FINISHED MATRIX MULTIPLICATION ROW K OVER TO DIAGONAL             C1850000
```

```
C                                                                       C1E50100
C     ADD UP GYROSCOPIC, XDD AND EXTERNAL TORQUE ON NEST K-1            C1E50200
      IF(LEQU)PRINT 229, K,(ETC(I,K),I=1,3)                             C1E50300
      IF(LEQU)PRINT 230, K,(XDD(I,K),I=1,3)                             C1E50400
      IF(LEQU)PRINT 231, K,(PHI(I,K),I=1,3)                             C1E50500
      DO 27  I=1,3                                                      C1E50600
   27 ETC(I,K) = ETC(I,K) - XDD(I,K) + PHI(I,K)                         C1E50700
      IF(LEQU)PRINT 232, K,K,K,K,(ETC(I,K),I=1,3)                       C1E50800
   24 CONTINUE                                                          C1E50900
      IF(.NOT. LEQU) GO TO 3000                                         C1E51000
      PRINT 227                                                         C1E51100
      DO 15  K=1,NB1                                                    C1E51200
   15 PRINT 229, K,(ETC(I,K),I=1,3)                                     C1E51300
      PRINT 227                                                         C1E51400
 3000 CONTINUE                                                          C1E51500
C                                                                       C1E51600
C      COMPUTE (NFRE)X1 COLUMN MATRIX OF TORQUE COMPONENTS ALONG        C1E51700
C        FREE COORDINATE AXES                                           C1E51800
      DO 28  K=1,NB1                                                    C1E51900
      MBEG=SUF(K)                                                       C1E52000
      MTERM=MBEG+2-PCON(K)                                              C1E52100
      IF(MBEG.GT.MTERM) GO TO 5061                                      C1E52200
      DO 28 M=MBEG,MTERM                                                C1E52300
      CALL VECDOT(QFC(1,M),ETC(1,K),ETM(M))                             C1E52400
      IF(LEQU)PRINT 224, M,M,K                                          C1E52500
 5061 CONTINUE                                                          C1E52600
   28 CONTINUE                                                          C1E52700
      IF(.NOT. LEQU) RETURN                                             C1E52800
      PRINT 227                                                         C1E52900
      MTERM=NFER+NMV+ NMODS                                             C1E53000
      DO 16 M=1,MTERM                                                   C1E53100
   16 PRINT 223, M,ETM(M)                                               C1E53200
C                                                                       C1E53300
C     REDUCTION OF VECTOR DYADIC EQUATIONS TO SCALAR EQUATIONS COMPLETE C1E53400
C                                                                       C1E53500
C                          SUM OVER N=1,NFER+NMV+NMODS                  C1E53600
C                                                                       C1E53700
C                    XMN(M,N)*THDD(N) = ETM(M)                          C1E53800
C                                                                       C1E53900
  200 FORMAT (4(2X,A4,I2,',',I2,') =',D15.8))                           C1E54000
  201 FORMAT (//,'  LABELS OF VARIABLE SPEED MOMENTUM WHEELS ',10I5,/)   C1E54100
  202 FORMAT ('  BODY ',I2,' TIED TO SYSTEM AT RIGID HINGE ')           C1E54200
  203 FORMAT(/,'  XMN(',I2,',',I2,') = PLM(',I2,') = ',D17.8)           C1E54300
  204 FORMAT (60X,'  ETM(',I2,') = ',D17.8)                             C1E54400
  205 FORMAT (' NON-ZERO COLUMNS IN ROW ',I2,' OF XDIC OVER TO DIAGONAL C1E54500
     *ELEMENT ARE ',11I5)                                               C1E54600
  206 FORMAT (' XMN(',I2,',',I2,') = QFC(',I2,').(XDIC(',I2,').QFC(',I2,C1E54700
     *')) = QFC(',I2,').(XDIC(',I2,',',I2,').QFC(',I2,') = ',D17.8)     C1E54800
  207 FORMAT (60X,'  ETM(',I2,') = ETM(',I2,') - PLM(',I2,') * HMC(',I2,C1E54900
     *') . DCMC(',I2,') ')                                              C1E55000
  208 FORMAT ('  TEM(',I2,') = DCMC(',I2,') = ',3D17.8)                 C1E55100
  209 FORMAT ('  XMN(',I2,',',I2,') = PLM(',I2,') * HMC(',I2,') . QFC(',C1E55200
     *I2,') = ',D17.8)                                                  C1E55300
  210 FORMAT (60X,'  ETM(',I2,') = ETM(',I2,') + CLM(',I2,') = ',D17.8) C1E55400
  211 FORMAT (30X,'  CACC(',I2,') = XMC(',I2,')*CAC(',I2,') ')         C1E55500
  212 FORMAT (30X,'  TEM2 = DCMC(',I2,') X (DCMC(',I2,') X CACC(',I2,')'C1E55600
     *)                                                                 C1E55700
  213 FORMAT (30X,'  TEM2 = TEM2 + DCMC(',I2,') X (DCMC(',I2,') X CEC(',C1E55800
     *I2,') ')                                                          C1E55900
  214 FORMAT ('  XMN = 0.0 ',///)                                       C1E56000
```

```
 215 FORMAT ('  XDQ(',I2,') = 0 ')                                          C1856100
 216 FORMAT (' ROW ',I2,' OF XDIC LEFT OF DIAGONAL USE ELEMENTS IN CCLC1856200
    *MNS ',1015)                                                          C1856300
 217 FORMAT (' ROW ',I2,' OF XDIC RIGHT OF DIAGONAL USE ELEMENTS IN COL C1856400
    *UMNS ',1015)                                                         C1856500
 218 FORMAT (' XQD(',I2,') = XGD(',I2,') + XDIC(',I2,') . TEM(',I2,')  C1856600
    *  =',' XGD(',I2,') + XDIC(',I2,',',I2,') . TEM(',I2,') ')          C1856700
 219 FORMAT ('  XMN(',I2,',',I2,') = QFC(',I2,') . (XMAS(',I2,')*FLAC('C1856800
    *,I2,') =',D12.5)                                                     C1856900
 220 FORMAT (' XQD(',I2,') = XGD(',I2,') + XDIC(',I2,')**T . TEM(',I2,'C1857000
    *) = XDQ(',I2,') + XDIC(',I2,',',I2,') . TEM(',I2,') ')            C1857100
 222 FORMAT (30X,'  TEM2 = XMAS(',I2,')*TEM2 =',17X,3D12.5)             C1857200
 223 FORMAT (' ETM(',I2,') = ',3D17.8)                                    C1857300
 224 FORMAT (' ETM(',I2,') = QFC(',I2,').ETC(',I2,') ')                   C1857400
 226 FORMAT ('1  SUBROUTINE QFCOT ENTERED ',2(/))                         C1857500
 227 FORMAT (3(/))                                                        C1857600
 228 FORMAT ('  ')                                                        C1857700
 229 FORMAT ('  ETC(',I2,') = ',3D17.8)                                   C1857800
 230 FORMAT ('  XQD(',I2,') = ',3D17.8)                                   C1857900
 231 FORMAT ('  PHI(',I2,') = ',3D17.8)                                   C1858000
 232 FORMAT ('  ETC(',I2,') = ETC(',I2,') - XQD(',I2,') + PHI(',I2,') =C1858100
    *',3D17.8,//)                                                         C1858200
 233 FORMAT ('  ETM(',I2,') = ',D17.8)                                    C1858300
 234 FORMAT (///,40X,'  BODY ',I2,' IS A FLEXIBLE BODY IN NEST ',I2,/)    C1858400
 235 FORMAT (/,'  FTEM(',I2,',',I2,',',I2,') = XMAS(',I2,')*GAM(',I2,',C1858500
    *',I2,') X FLAC(',I2,') + FLQC(',I2,') =',3D12.5)                     C1858600
 236 FORMAT ('  XMN(',I2,',',I2,') = QFC(',I2,').FTEM(',I2,',',I2,',',ICI858700
    *2,') =',6X,D12.5)                                                    C1858800
 237 FORMAT ('  XMN(',I2,',',I2,') = XMAS(',I2,') =',20X,D12.5)           C1858900
 238 FORMAT (/,30X,'  ETM(',I2,') = ETM(',I2,') - FTEM(',I2,',',I2,',',C1859000
    *I2,').DCMC(',I2,') =',38X,D12.5)                                     C1859100
 239 FORMAT (30X,'  ETM(',I2,') = ETM(',I2,') - XMAS(',I2,')*(2.*ZETA('C1859200
    *,I2,')*FLOM(',I2,')*THAD(',I2,') + FLOM(',I2,')**2*THA(',I2,') =',C1859300
    *D12.5)                                                               C1859400
 240 FORMAT (30X,'  ETM(',I2,') = ETM(',I2,') + FCMC(',I2,').FLHC(',I2,C1859500
    *').FCMC(',I2,') - TEM2.FLAC(',I2,') =',19X,D12.5)                    C1859600
 241 FORMAT (30X,'  TEM2 = 0 ')                                           C1859700
 242 FORMAT (30X,'  FCUP(',I2,') = 0          KCUP(',I2,') = 0')         C1859800
 245 FORMAT ('  ')                                                        C1859900
 246 FORMAT (50X,3D12.5)                                                  C1860000
 247 FORMAT (10X,' FCUP(',I2,') = FCUP(',I2,') + THA(',I2,')*FCF(',I2,'C1860100
    *) =',3D12.5)                                                         C1860200
 248 FORMAT (10X,' KCUP(',I2,') = KCUP(',I2,') + THAD(',I2,')*FCK(',I2,C1860300
    *') =',3D12.5)                                                        C1860400
 249 FORMAT (30X,' ETM(',I2,') = ETM(',I2,') + FCMC(',I2,').FCUP1.FCMC(C1860500
    *',I2,') - 2.0*FOMC(',I2,').KCUP1 =',D12.5)                           C1860600
 250 FORMAT (71X,3D12.5)                                                  C1860700
 251 FORMAT (//,45X,'MODAL CROSS COUPLING SIGNIFICANT FOR BODY ',I2)      C1860800
 252 FORMAT (/,30X,'  ETM(',I2,') = 0 ')                                  C1860900
 253 FORMAT   (30X,'  FLHC(',I2,') = XMC(',I2,')*FLH(',I2,')*XMC(',I2,'C1861000
    *)**T =',3D12.5)                                                      C1861100
 254 FORMAT (48X,3D12.5)                                                  C1861200
 255 FORMAT (10X,' KCUP1 = XMC(',I2,')*KCUP(',I2,') =',3D12.5)           C1861300
 256 FORMAT (/,' EQUATION OF MOTION FOR MODE ',I2,' OF BODY ',I2,' (NOC1861400
    *E NUMBER ',I2,') HAS CROSS COUPLING FROM MODES ',1014)               C1861500
 257 FORMAT (10X,' FCUP1 = XMC(',I2,')*FCUP(',I2,')*XMC(',I2,')**T =',  C1861600
    *3D12.5)                                                              C1861700
C                                                                         C1861800
C                                                                         C1861900
      RETURN                                                              C1862000
```

```
      END                                                               C1E62100
C                                                                       C1900000
      SUBROUTINE DCT                                                    C1900100
C     USED TO DEFINE DIFFERENTIAL EQUATIONS WHICH MUST BE INTEGRATED    C1900200
C       TO DEFINE DIRECTION COSINE MATRICES                             C1900300
C                                                                       C1900400
C         SD  = ALL CONTIGUOUS PAIRS OF BODIES (K,JCON(K)) HAVING       C1900500
C               SIGNIFICANT RELATIVE MOTION. DELETION OF K FROM SD      C1900600
C               IMPLIES TRANSFORMATION MATRIX OF BODY K TO JCON(K)      C1900700
C               IS CONSTANT IN TIME, OR DEFINED BY SMALL ANGLE ASSUMPTIOC1900800
C               OR EULER ANGLE TECHNIQUES                               C1900900
C                                                                       C1901000
C                                                                       C1901100
C                                                                       C1901200
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C1901300
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C1901400
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1901500
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1901600
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1901700
C                                                                       C1901800
C                                                                       C1901900
      INTEGER                                                           C1902000
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  , C1902100
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    , C1902200
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   , C1902300
     * SMA   , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   , C1902400
     * SQL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   , C1902500
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   , C1902600
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   , C1902700
     * IINIT(1)      , IZINIT(1)     , SD                              C1902800
C                                                                       C1902900
C                                                                       C1903000
      REAL*8                                                            C1903100
     * ANGC  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  , C1903200
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                 C1903300
     * THACD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C1903400
C                                                                       C1903500
C                                                                       C1903600
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C1903700
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C1903800
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C1903900
C                                                                       C1904000
C                                                                       C1904100
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C1904200
C                                                                       C1904300
C                                                                       C1904400
      COMMON /INTG/ AWORK(200),                                         C1904500
     * CT1           , CT2           , CT3           , CT4           ,  C1904600
     * CT5           , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C1904700
     * MC    (10)    , NBI           , NBOD          , NCTC          ,  C1904800
     * NFER          , NFKC          , NFRC          , NLOR          ,  C1904900
     * NMV           , NMO           , NMCA          , NSVP          ,  C1905000
     * NSVC          , PCON  (11)    , SD            , SFR   (33)    ,  C1905100
     * SG            , SI    (55)    , SIG           , SL            ,  C1905200
     * SLK   (33)    , SMA   (10)    , SOK   (11)    , SQF   (11)    ,  C1905300
```

```
     * SGL    (11)     , SMV              , SR               , SSCN             .  C1905400
     * SSIX             , SVA              , SVB              , SVD              .  C1905500
     * SVI              , SVM              , SVP    (22)      , SVQ    (33)      .  C1905600
     * SXM    (3,10)    , SXT              , TORQ   (97)      , SMAL             .  C1905700
     * SEU              , NTQ              , SC     (33)      , SCG              .  C1905800
     * NFLXB            , SFLX             , SFXM   (10)      , NMODS            .  C1905900
     * SECC             , SCC    (10)                                             C1906000
C                                                                                 C1906100
C                                                                                 C1906200
      COMMON /INTGZ/                                                              C1906300
     * SCNDUM           , SCN    (9)       , SCRDUM           , SCR    (9)       .  C1906400
     * SFKDUM           , SFK    (9)       , SIXDUM           , SIX    (9)       .  C1906500
     * SKDUM            , SK     (9)       , SPIDUM           , SPI    (9)       .  C1906600
     * SMCDUM           , SMC    (9)                                              C1906700
C                                                                                 C1906800
C                                                                                 C1906900
      COMMON /REAL/                                                               C1907000
     * CA     (3,10)    , CAC    (3,10)    , CLM    (10)      , COMC   (3,11)    .  C1907100
     * DCMC   (3,11)    , ETC    (3,11)    , ETM    (33)      , FCMC   (3,11)    .  C1907200
     * GAM    (3,66)    , H                , HM     (3,10)    , HMC    (3,10)    .  C1907300
     * HMCM   (10)      , PHI    (3,11)    , PLM    (10)      , QF     (3,33)    .  C1907400
     * QFC    (3,33)    , QL     (3,22)    , QLC    (3,22)    , RCMC   (3,11)    .  C1907500
     * T                ,                  THA      (33)      , THAD   (33)      .  C1907600
     * THADM  (10)      , THAM   (10)      , XDIC   (3,3,66)  , XI     (3,3,10)  .  C1907700
     * XIC    (3,3,10)  , XMAS   (10)      , XMN    (33,33)   , XMT    (3,3,10)  .  C1907800
     * TUG    (33)      , FLA    (3,20)    , FLB    (3,20)    , FLC    (3,20)    .  C1907900
     * FLD    (3,3,20)  , FLJ    (3,3,20)  , CAD    (3,10)    , XID    (3,3,10)  .  C1908000
     * FLIRC  (3,10)    , FLCRC  (3,10)    , FLAC   (3,20)    , FLQC   (3,20)    .  C1908100
     * FLCM   (20)      , ZETA   (20)      , FCF    (3,3,40)  , FCK    (3,40)    .  C1908200
     * TIMEND                                                                     C1908300
C                                                                                 C1908400
C                                                                                 C1908500
      COMMON /REAL2/                                                              C1908600
     * CBDUM  (1,3)     , CB     (3,10)    , CBCDUM(1,3)      , CBC    (3,10)    .  C1908700
     * XMCDUM(1,1,9)    , XMC    (3,3,10)  , CBN(3)                               C1908800
C                                                                                 C1908900
C                                                                                 C1909000
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      .          C1909100
     *            (XMN(1,3),YMCD(1,1,1))    ,(XMN(1,6),CNF(1,1))     .          C1909200
     *            (XMN(1,8),ETIC(1,1))      ,(XMN(1,10),ETMC(1,1))   .          C1909300
     *            (FLB(1,1),FLQ(1,1))       ,(FLE(1,1,1),FLD(1,1,1)).          C1909400
     *            (FLH(1,1,1),FLJ(1,1,1))   ,                                    C1909500
     *            (FGI,LINIT(1))            ,(CA(1,1),RINIT(1))      .          C1909600
     *            (CBDUM(1,1),FZINIT(1))    ,(AWORK(1),IINIT(1))     .          C1909700
     *            (SCNDUM,IZINIT(1))                                             C1909800
C                                                                                 C1909900
      INTEGER SET(10)                                                             C1910000
      EQUIVALENCE (LDCT,LEQU)                                                     C1910100
C                                                                                 C1910200
C                                                                                 C1910300
C                                                                                 C1910400
C                                                                                 C1910500
      IF(.NOT. LEQU) GO TO 100C                                                   C1910600
006C00PRINT 1CC                                                                   C1910700
      JC= C                                                                       C1910800
      J1= 1                                                                       C1910900
      J2= 2                                                                       C1911000
      J3= 3                                                                       C1911100
 100C CONTINUE                                                                    C1911200
      N = C                                                                       C1911300
```

```
      IF(LEQU) PRINT 104,INERF                                          C1911400
      CALL UNPAC(SET,NSET,SD)                                           C1911500
      IF(NSET.EQ.0) GO TO 5062                                          C1911600
      IF(LEQU) PRINT 103, SD,(SET(I),I=1,NSET)                          C1911700
      IF((INERF.OR.SET(NSET).NE.1) GO TO 2                              C1911800
C     TRANS BODY 1 TO BODY 1 COMPUTING FRAME NOT NEEDED                 C1911900
      NSET = NSET - 1                                                   C1912000
C     ACCOUNT FOR INERTIAL TO BODY 1 TRANSFORMATION                     C1912100
C      ANGULAR VELOCITY INERTIAL TO BODY 1                              C1912200
C           = - ANGULAR VELOCITY BODY 1 TO INERTIAL                     C1912300
C           = - FOMC(I,1), I=1,2,3                                      C1912400
      N = N+1                                                           C1912500
      DO 4  I=1,2                                                       C1912600
C     GET MINUS SIGN IN BY REVERSEING VECTOR CROSS PRODUCT              C1912700
      CALL VECROS (XMC(1,I,0),FOMC(1,1),YMCD(1,I,N))                    C1912800
      IF(LEQU) PRINT 106, I,N,I,JC,(YMCD(J,I,N),J=1,3)                  C1912900
    4 CONTINUE                                                          C1913000
      IF(LEQU) PRINT 105                                                C1913100
    2 IF(NSET.EQ.0) GO TO 5062                                          C1913200
      DO 3 KK=1,NSET                                                    C1913300
      N = N+1                                                           C1913400
      K = SET(KK)                                                       C1913500
      DO 6  I=1,2                                                       C1913600
      CALL VECROS (CCMC(1,K),XMC(1,I,K),YMCD(1,I,N))                    C1913700
      IF(LEQU) PRINT 107, I,N,K,I,K,(YMCD(J,I,N),J=1,3)                 C1913800
    6 CONTINUE                                                          C1913900
      IF(LEQU) PRINT 105                                                C1914000
    3 CONTINUE                                                          C1914100
 5062 CONTINUE                                                          C1914200
  100 FORMAT ('1   SUBROUTINE DCT ENTERED ')                            C1914300
  103 FORMAT (' SD =',Z8,' SET ELEMENTS ',10I5)                         C1914400
  104 FORMAT ('  INERF = ',L10)                                         C1914500
  105 FORMAT (' ')                                                      C1914600
  106 FORMAT ('  YMCD(',I2,',',I2,') = XMC(',I2,',',I2,') X FOMC( 1) = ',C1914700
     *,3D17.8)                                                          C1914800
  107 FORMAT ('  YMCD(',I2,',',I2,') = CCMC(',I2,') X XMC(',I2,',',I2,')C1914900
     *,' = ',3D17.8)                                                    C1915000
      RETURN                                                            C1915100
      END                                                               C1915200
```

```
C                                                                       C2000000
      SUBROUTINE ANGLE                                                  C2000100
C       USED TO SET UP DIFFERENTIAL EQUATIONS WHICH DEFINE              C2000200
C       ANGULAR DISPLACEMENT ABOUT FREE COORDINATES                     C2000300
C     RECALL                                                            C2000400
C           SFR(I)  = FREE COORDINATE AXES ABOUT WHICH ANGLE TO         C2000500
C                     BE COMPUTED                                       C2000600
C                                                                       C2000700
C                                                                       C2000800
C                                                                       C2000900
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2001000
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C2001100
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2001200
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2001300
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMC        C2001400
C                                                                       C2001500
```

```
C                                                                       C2001600
      INTEGER                                                           C2001700
     * AWORK , CT1    , CT2    , CT3    , CT4    , CT5    , FCON   , PCON   ,  C2001800
     * SCNDUM, SCN    , SCRDUM, SCR    , SFKDUM, SFK    , SFR    , SG     ,  C2001900
     * SI     , SIG    , SIXDUM, SIX    , SKDUM , SK     , SL     , SLK    ,  C2002000
     * SMA    , SMCDUM, SMC    , SMV    , SOK   , SPIDUM, SPI    , SQF    ,  C2002100
     * SQL    , SR     , SSCN   , SSIX   , SVA   , SVE    , SVD    , SVI    ,  C2002200
     * SVM    , SVP    , SVQ    , SXM    , SXT   , TORQ   , SMAL   , SEU    ,  C2002300
     * SC     , SCG    , NFLXB  , SFLX   , SFXM  , NMODS  , SFCC   , SCC    ,  C2002400
     * IINIT(1)        , IZINIT(1)       , SD                             C2002500
C                                                                       C2002600
C                                                                       C2002700
      REAL*8                                                            C2002800
     * ANGE  (33)     , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C2002900
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                 C2003000
     * THADD (33)     , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)      C2003100
C                                                                       C2003200
C                                                                       C2003300
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2003400
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2003500
     *                LCFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C2003600
C                                                                       C2003700
C                                                                       C2003800
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C2003900
C                                                                       C2004000
C                                                                       C2004100
      COMMON /INTG/ AWORK(200) ,                                        C2004200
     * CT1             , CT2             , CT3             , CT4       ,  C2004300
     * CT5             , FCON   (33)     , JCON   (10)     , LCON (22) ,  C2004400
     * MC     (10)     , NEI             , NBOD            , NCTC      ,  C2004500
     * NFER            , NFKC            , NFRC            , NLOR      ,  C2004600
     * NMV             , NMO             , NMOA            , NSVP      ,  C2004700
     * NSVC            , PCON   (11)     , SD              , SFR  (33) ,  C2004800
     * SG              , SI     (55)     , SIG             , SL        ,  C2004900
     * SLK    (33)     , SMA    (10)     , SOK    (11)     , SQF  (11) ,  C2005000
     * SQL    (11)     , SMV             , SR              , SSCN      ,  C2005100
     * SSIX            , SVA             , SVB             , SVD       ,  C2005200
     * SVI             , SVM             , SVP    (22)     , SVQ  (33) ,  C2005300
     * SXM    (3,10)   , SXT             , TORQ   (97)     , SMAL      ,  C2005400
     * SEU             , NTQ             , SC     (33)     , SCG       ,  C2005500
     * NFLXB           , SFLX            , SFXM   (10)     , NMODS     ,  C2005600
     * SFCC            , SCC    (10)                                     C2005700
C                                                                       C2005800
C                                                                       C2005900
      COMMON /INTGZ/                                                    C2006000
     * SCNDUM          , SCN    (9)      , SCRDUM          , SCR  (9)  ,  C2006100
     * SFKDUM          , SFK    (9)      , SIXDUM          , SIX  (9)  ,  C2006200
     * SKDUM           , SK     (9)      , SPIDUM          , SPI  (9)  ,  C2006300
     * SMCDUM          , SMC    (9)                                      C2006400
C                                                                       C2006500
C                                                                       C2006600
      COMMON /REAL/                                                     C2006700
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , CCMC (3,11) ,  C2006800
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FCMC (3,11) ,  C2006900
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC  (3,10) ,  C2007000
     * HMCM   (10)     , PHI    (3,11)   , PLM    (10)     , QF   (3,33) ,  C2007100
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RCMC (3,11) ,  C2007200
     * T                ,                  THA    (33)     , THAD (33)    ,  C2007300
     * THADW  (10)     , THAW   (10)     , XDIC   (3,3,66), XI   (3,3,10),  C2007400
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT  (3,3,10),  C2007500
```

```
     * TUG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)   ,  C2007600
     * FLD    (3,3,20), FLJ    (3,3,20), CAO    (3,10)   , XIU    (3,3,10),  C2007700
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLGC   (3,20)   ,  C2007800
     * FLCN   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)   ;  C2007900
     * TIMEND                                                                   C2008000
C                                                                               C2008100
C                                                                               C2008200
      COMMON /REALZ/                                                            C2008300
     * CBDUM (1,3)    , CE     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)   ,  C2008400
     * XMCDUM(1,1,9) , XMC    (3,3,10), CBN(3)                                 C2008500
C                                                                               C2008600
C                                                                               C2008700
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))     ,          C2008800
     *            (XMN(1,3),YMCC(1,1,1))     ,(XMN(1,6),CNF(1,1))    ,          C2008900
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))  ,          C2009000
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),          C2009100
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                  C2009200
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))     ,          C2009300
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))    ;          C2009400
     *            (SCNDUM,IZINIT(1))                                            C2009500
C                                                                               C2009600
      EQUIVALENCE (LANGLE,LEGU)                                                 C2009700
      INTEGER S1(33),S2(33)                                                     C2009800
C                                                                               C2009900
C                                                                               C2010000
C                                                                               C2010100
C                                                                               C2010200
C                                                                               C2010300
      IF(LEQU) PRINT 100                                                        C2010400
C                                                                               C2010500
C     PROCEED SEQUENTIALLY NOW DEFINING ONLY THOSE EQUATIONS CALLED FOR         C2010600
C      TAKING SPECIAL NOTE OF 3-AXIS GIMBALS BETWEEN RIGID BODIES               C2010700
      IF(LEQU) PRINT 101, NFRC,(SFR(I),I=1,NFRC)                                C2010800
      K = 1                                                                     C2010900
      N = 1                                                                     C2011000
      DO 5  N=1,NB1                                                             C2011100
      IF(N.EQ.1) GO TO 9                                                        C2011200
      M = M+3-PCON(N-1)                                                         C2011300
    9 CONTINUE                                                                  C2011400
C     MT = HIGHEST MAGNITUDE INDICE AT HINGE POINT N-1                          C2011500
      MT = M+2-PCON(N)                                                          C2011600
    7 IF(SFR(K).GT.MT.OR.K.GT.NFRC) GO TO 5                                     C2011700
      IF(N.EQ.NB1) GO TO 4                                                      C2011800
      IF(RELO(N).AND.PCON(N).EQ.0) GO TO 6                                      C2011900
      IF(RELO(N).AND.PCON(N).EQ.1) GO TO 1                                      C2012000
    4 CONTINUE                                                                  C2012100
      IF(SFR(K).EQ.0) GO TO 5                                                   C2012200
      ANGD(K) = THAD(SFR(K))                                                    C2012300
      IF(LEQU) KK=SFR(K)                                                        C2012400
      IF(LEQU) PRINT 102, K,KK,ANGD(K)                                          C2012500
      K = K+1                                                                   C2012600
      GO TO 7                                                                   C2012700
C                                                                               C2012800
C     SPECIAL CASE CHECK FOR SKEWED TWO AXIS GIMBAL                             C2012900
    1 CALL VECDOT(QFC(1,M),QFC(1,M+1),C)                                        C2013000
      IF(C.EQ.C.D0) GO TO 4                                                     C2013100
      CALL VECDOT(QFC(1,M),RCMC(1,N),A1)                                        C2013200
      CALL VECDOT(QFC(1,M+1),RCMC(1,N),A3)                                      C2013300
      ANGD(K) = (A1 - C*A3)/(1.D0 - C**2)                                       C2013400
      ANGD(K+1) = (A3 - C*A1)/(1.D0 - C**2)                                     C2013500
```

```
      IF(.NOT.LEQU) GO TO 1001                                          C2013600
      M1 = M+1                                                          C2013700
      K1 = K+1                                                          C2013800
      PRINT 103, M,N,A1                                                 C2013900
      PRINT 105, M1,N,A3                                                C2014000
      PRINT 106, M,M1,C                                                 C2014100
      PRINT 107, K,ANGD(K)                                              C2014200
      PRINT 109, K1,ANGD(K1)                                            C2014300
 1001 CONTINUE                                                          C2014400
      K = K+2                                                           C2014500
      GO TO 5                                                           C2014600
C                                                                       C2014700
    6 CONTINUE                                                          C2014800
C     SPECIAL CASE THREE AXIS GIMBAL - NOTE LOGIC ASSUMES THAT          C2014900
C                                       SFR(K),SFR(K+1),SFR(K+2)        C2015000
C                                     ARE THE THREE GIMBAL AXES         C2015100
      CALL VECDOT(QFC(1,M),ROMC(1,N),A1)                                C2015200
      CALL VECDOT(QFC(1,M+1),ROMC(1,N),A2)                              C2015300
      CALL VECDOT(QFC(1,M+2),ROMC(1,N),A3)                              C2015400
      CALL VECDOT(QFC(1,M),QFC(1,M+2),C)                                C2015500
      ANGD(K) = (A1 - C*A3)/(1 - C**2)                                  C2015600
      ANGD(K+1) = A2                                                    C2015700
      ANGD(K+2) = (A3 - C*A1)/(1 - C**2)                                C2015800
      IF(.NOT. LEQU) GO TO 1000                                         C2015900
      M1 = M+1                                                          C2016000
      M2 = M+2                                                          C2016100
      K1 = K+1                                                          C2016200
      K2 = K+2                                                          C2016300
      PRINT 103, M,N,A1                                                 C2016400
      PRINT 104, M1,N,A2                                                C2016500
      PRINT 105, M2,N,A3                                                C2016600
      PRINT 106, M,M2,C                                                 C2016700
      PRINT 107, K,ANGD(K)                                              C2016800
      PRINT 108, K1,ANGD(K+1)                                           C2016900
      PRINT 109, K2,ANGD(K+2)                                           C2017000
 1000 CONTINUE                                                          C2017100
      K = K+3                                                           C2017200
    5 CONTINUE                                                          C2017300
C                                                                       C2017400
  100 FORMAT ('1  SUBROUTINE ANGLE ENTERED ')                           C2017500
  101 FORMAT (I10,' ELEMENTS IN ARRAY SFR, THEY ARE ',3X13)             C2017600
  102 FORMAT ('  ANGD(',I2,') = THAD(',I2,') = ',D17.8)                 C2017700
  103 FORMAT ('  A1 = QFC(',I2,') . ROMC(',I2,')  = ',D17.8)            C2017800
  104 FORMAT ('  A2 = QFC(',I2,') . ROMC(',I2,')  = ',D17.8)            C2017900
  105 FORMAT ('  A3 = QFC(',I2,') . ROMC(',I2,')  = ',D17.8)            C2018000
  106 FORMAT ('   C = QFC(',I2,') . QFC(',I2,')   = ',D17.8)            C2018100
  107 FORMAT ('  ANGD(',I2,') = (A1-C*A3)/(1-C**2) = ',D17.8)           C2018200
  108 FORMAT ('  ANGD(',I2,') = A2                 = ',D17.8)           C2018300
  109 FORMAT ('  ANGD(',I2,') = (A3-C*A1)/(1-C**2) = ',D17.8)           C2018400
  110 FORMAT (I10,' ELEMENTS IN ARRAY SMA, THEY ARE ',1013)             C2018500
      RETURN                                                            C2018600
      END                                                               C2018700
```

```
C                                                                       C2100000
      SUBROUTINE SETUP(Y,YD,NEG,SORT)                                   C2100100
C       IF   SORT = .TRUE.  SORT OUT INTEGRATED QUANTITIES              C2100200
```

```
C                = .FALSE. SET UP ONE DIMENSIONAL ARRAY OF EQUATIONS TOC2100300
C                      BE INTEGRATED                                    C2100400
C                                                                       C2100500
C                                                                       C2100600
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2100700
C                                                                       C2100800
C                                                                       C2100900
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C2101000
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2101100
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2101200
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C2101300
C                                                                       C2101400
C                                                                       C2101500
      INTEGER                                                           C2101600
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , PCON  ,  C2101700
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C2101800
     * SI    , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C2101900
     * SMA   , SMCDUM, SMC   , SMV   , SQK   , SPIDUM, SPI   , SQF   ,  C2102000
     * SGL   , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,  C2102100
     * SVM   , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  C2102200
     * SC    , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  C2102300
     * IINIT(1)      , IZINIT(1)     , SD                              C2102400
C                                                                       C2102500
C                                                                       C2102600
      REAL*8                                                            C2102700
     * ANGL  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C2102800
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),                 C2102900
     * THACD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)       C2103000
C                                                                       C2103100
C                                                                       C2103200
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2103300
     *                LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2103400
     *                LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ        C2103500
C                                                                       C2103600
C                                                                       C2103700
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)           C2103800
C                                                                       C2103900
C                                                                       C2104000
      COMMON /INTG/ AWORK(184),ST1,N,M,MM,NST1,J,I,                     C2104100
     * CT1           , CT2           , CT3           , CT4           ,  C2104200
     * CT5           , FCON  (33)    , JCON  (10)    , LCON  (22)    ,  C2104300
     * MC    (10)    , NB1           , NBOD          , NCTC          ,  C2104400
     * NFER          , NFKC          , NFRC          , NLOR          ,  C2104500
     * NMV           , NMO           , NMOA          , NSVP          ,  C2104600
     * NSVC          , PCON  (11)    , SD            , SFR   (33)    ,  C2104700
     * SG            , SI    (55)    , SIG           , SL            ,  C2104800
     * SLK   (33)    , SMA   (10)    , SQK   (11)    , SQF   (11)    ,  C2104900
     * SGL   (11)    , SMV           , SR            , SSCN          ,  C2105000
     * SSIX          , SVA           , SVB           , SVD           ,  C2105100
     * SVI           , SVM           , SVP   (22)    , SVQ   (33)    ,  C2105200
     * SXM   (3,10)  , SXT           , TORQ  (97)    , SMAL          ,  C2105300
     * SEU           , NTQ           , SC    (33)    , SCG           ,  C2105400
     * NFLXB         , SFLX          , SFXM  (10)    , NMODS         ,  C2105500
     * SFCC          , SCC   (10)                                      C2105600
C                                                                       C2105700
C                                                                       C2105800
      COMMON /INTGZ/                                                    C2105900
     * SCNDUM        , SCN   (9)     , SCRDUM        , SCR   (9)     ,  C2106000
     * SFKDUM        , SFK   (9)     , SIXDUM        , SIX   (9)     ,  C2106100
     * SKDUM         , SK    (9)     , SPIDUM        , SPI   (9)     ,  C2106200
```

```
     * SMCDUM          , SMC    (9)                                          C2106300
C                                                                            C2106400
C                                                                            C2106500
      COMMON /REAL/                                                          C2106600
     * CA     (3,10)   , CAC    (3,10)   , CLM    (10)     , CDMC   (3,11)  , C2106700
     * DCMC   (3,11)   , ETC    (3,11)   , ETM    (33)     , FCMC   (3,11)  , C2106800
     * GAM    (3,66)   , H                , HM     (3,10)   , HMC    (3,10)  , C2106900
     * HMCM   (10)     , PHI    (3,11)   , PLM    (10)     , QF     (3,33)  , C2107000
     * QFC    (3,33)   , QL     (3,22)   , QLC    (3,22)   , RCMC   (3,11)  , C2107100
     * T                ,                  THA    (33)     , THAD   (33)    , C2107200
     * THADM  (10)     , THAM   (10)     , XDIC   (3,3,66), XI     (3,3,10), C2107300
     * XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10), C2107400
     * TLG    (33)     , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)  , C2107500
     * FLD    (3,3,20), FLJ    (3,3,20), CAD    (3,10)   , XIO    (3,3,10), C2107600
     * FLIRC  (3,10)   , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)  , C2107700
     * FLCM   (20)     , ZETA   (20)     , FCF    (3,3,40), FCK    (3,40)  , C2107800
     * TIMEND                                                                 C2107900
C                                                                            C2108000
C                                                                            C2108100
      COMMON /REALZ/                                                         C2108200
     * CEDUM (1,3)     , CE     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)  , C2108300
     * XMCDUM(1,1,9)   , XMC    (3,3,10), CBN(3)                            C2108400
C                                                                            C2108500
C                                                                            C2108600
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))     ,       C2108700
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))    ,       C2108800
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))  ,       C2108900
     *            (FLB(1,1),FLG(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),       C2109000
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                               C2109100
     *            (FGI,LINIT(1))             ,(CA(1,1),RINIT(1))     ,       C2109200
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))    ,       C2109300
     *            (SCNDUM,IZINIT(1))                                         C2109400
C                                                                            C2109500
      INTEGER ST1(10)                                                        C2109600
      LOGICAL SORT                                                           C2109700
      DIMENSION Y(NEQ),YD(NEQ)                                               C2109800
      EQUIVALENCE (LSETUP,LEQU)                                              C2109900
C                                                                            C2110000
C                                                                            C2110100
      IF(LEQU) PRINT 100                                                     C2110200
C                                                                            C2110300
C     ANGULAR OR LINEAR RATE RELATIVE TO FREE COORDINATE AXES                C2110400
      DO 1  N=1,NFER                                                         C2110500
      IF(SORT) GO TO 10                                                      C2110600
      YD(N) = THADD(N)                                                       C2110700
      IF(LEQU) PRINT 102, N,N,YD(N)                                          C2110800
      GO TO 1                                                                C2110900
   10 THAD(N) = Y(N)                                                         C2111000
      IF(LEQU) PRINT 202,N,N,THAD(N)                                         C2111100
    1 CONTINUE                                                               C2111200
      NEQ = NFER                                                             C2111300
C                                                                            C2111400
C                                                                            C2111500
C     GENERALIZED ELASTIC COORDINATE RATE EQUATIONS                          C2111600
      IF(NMODS.EQ.0) GO TO 5063                                              C2111700
      DO 7 NN=1,NMODS                                                        C2111800
      N = NEQ+NN                                                             C2111900
      IF(SORT) GO TO 9                                                       C2112000
      YD(N) = THADD(N)                                                       C2112100
      IF(LEQU) PRINT 102, N,N,YD(N)                                          C2112200
```

```
      GO TO 7                                                          C2112300
    5 THAD(N) = Y(N)                                                   C2112400
      IF(LEQU) PRINT 202, N,N,THAD(N)                                  C2112500
    7 CONTINUE                                                         C2112600
 5063 CONTINUE                                                         C2112700
      NEQ = NEQ + NMODS                                                C2112800
C                                                                      C2112900
C                                                                      C2113000
C     RELATIVE ANGULAR MOMENTUM WHEEL                                  C2113100
      CALL UNPAC(ST1,NST1,SMV)                                         C2113200
      IF(NST1.EQ.0) GO TO 5064                                         C2113300
      DO 2 MMM=1,NST1                                                  C2113400
      MM=NST1-(MMM-1)                                                  C2113500
      M = ST1(MM)                                                      C2113600
      N = NEQ+NST1+1-MM                                                C2113700
      IF(SCRT) GO TO 11                                                C2113800
      YD(N) = THADD(N)                                                 C2113900
      IF(LEQU) PRINT 104, N,N,YD(N)                                    C2114000
      GO TO 2                                                          C2114100
   11 THADW(M) = Y(N)                                                  C2114200
      HMOM(M) = PLM(M)*THADW(M)                                        C2114300
      IF(LEQU) PRINT 205, M,N,THADW(M)                                 C2114400
      IF(LEQU) PRINT 204, M,M,M,HMOM(M)                                C2114500
    2 CONTINUE                                                         C2114600
 5064 CONTINUE                                                         C2114700
      NEQ = NEQ + NST1                                                 C2114800
C                                                                      C2114900
C     DISPLACEMENT ABOUT OR ALONG FREE COORDINATE AXES                 C2115000
      IF(NFRC.EQ.0) GO TO 5065                                         C2115100
      DO 3  MM=1,NFRC                                                  C2115200
      N = NEQ + MM                                                     C2115300
      IF(SCRT) GO TO 12                                                C2115400
      YD(N) = ANGD(MM)                                                 C2115500
      IF(LEQU) PRINT 106, N,MM,YD(N)                                   C2115600
      GO TO 3                                                          C2115700
   12 M = SFR(MM)                                                      C2115800
      THA(M) = Y(N)                                                    C2115900
      IF(LEQU) PRINT 206, M,N,THA(M)                                   C2116000
    3 CONTINUE                                                         C2116100
 5065 CONTINUE                                                         C2116200
      NEQ = NEQ + NFRC                                                 C2116300
C                                                                      C2116400
C                                                                      C2116500
C     GENERALIZED ELASTIC COORDINATE DISPLACEMENT EQUATIONS            C2116600
      IF(NMODS.EQ.0) GO TO 5062                                        C2116700
      DO 19 NN=1,NMODS                                                 C2116800
      N = NEQ+NN                                                       C2116900
      NST1 = NFER+NN                                                   C2117000
      IF(SCRT) GO TO 20                                                C2117100
      YD(N) = THAD(NST1)                                               C2117200
      IF(LEQU) PRINT 103, N,NST1,YD(N)                                 C2117300
      GO TO 19                                                         C2117400
   20 CONTINUE                                                         C2117500
      THA(NST1) = Y(N)                                                 C2117600
      IF(LEQU) PRINT 206, NST1,N,THA(NST1)                             C2117700
   19 CONTINUE                                                         C2117800
 5062 CONTINUE                                                         C2117900
      NEQ = NEQ + NMODS                                                C2118000
C                                                                      C2118100
C                                                                      C2118200
```

```
C       DISPLACEMENT ABOUT WHEEL SPIN AXIS                              C2118300
        IF(NMCA.EQ.0) GO TO 5066                                        C2118400
        DO 4  MM=1,NMOA                                                 C2118500
        M = SMA(MM)                                                     C2118600
        N = NEQ + MM                                                    C2118700
        IF(SCRT) GO TO 13                                               C2118800
        YD(N) = THADW(M)                                                C2118900
        IF(LEQU) PRINT 108, N,M,YD(N)                                   C2119000
        GO TO 4                                                         C2119100
   13   CONTINUE                                                        C2119200
        THAW(M) = Y(N)                                                  C2119300
        IF(LEQU) PRINT 208, M,N,THAW(M)                                 C2119400
    4   CONTINUE                                                        C2119500
 5066   CONTINUE                                                        C2119600
        NEQ = NEQ + NMOA                                                C2119700
C                                                                       C2119800
C       DIRECTION COSINES                                               C2119900
        CALL UNPAC(ST1,NST1,SD)                                         C2120000
        N = NEQ                                                         C2120100
        M = 0                                                           C2120200
        IF(NST1.EQ.0) GO TO 5067                                        C2120300
        IF(INERF.OR.ST1(NST1).NE.1) GO TO 5                             C2120400
        NST1 = NST1 - 1                                                 C2120500
        IF(SCRT) GO TO 14                                               C2120600
        N = N+1                                                         C2120700
   14   DO 8  J=1,2                                                     C2120800
        DO 8  I=1,3                                                     C2120900
        N = N+1                                                         C2121000
        IF(SCRT) GO TO 15                                               C2121100
        YD(N) = YMCD(I,J,M)                                             C2121200
        IF(LEQU) PRINT 111, N,I,J,M,YD(N)                               C2121300
        GO TO 8                                                         C2121400
   15   XMC(I,J,M) = Y(N)                                               C2121500
        IF(LEQU) PRINT 211, I,J,M,N,XMC(I,J,M)                          C2121600
    8   CONTINUE                                                        C2121700
    5   IF(NST1.EQ.0) GO TO 5067                                        C2121800
        DO 6 MM=1,NST1                                                  C2121900
        IF(SCRT) GO TO 16                                               C2122000
        M = M+1                                                         C2122100
        GO TO 17                                                        C2122200
   16   M = ST1(MM)                                                     C2122300
   17   DO 6  J=1,2                                                     C2122400
        DO 6  I=1,3                                                     C2122500
        N = N+1                                                         C2122600
        IF(SCRT) GO TO 18                                               C2122700
        YD(N) = YMCD(I,J,M)                                             C2122800
        IF(LEQU) PRINT 111, N,I,J,M,YD(N)                               C2122900
        GO TO 6                                                         C2123000
   18   XMC(I,J,M) = Y(N)                                               C2123100
        IF(LEQU) PRINT 211, I,J,M,N,XMC(I,J,M)                          C2123200
    6   CONTINUE                                                        C2123300
 5067   CONTINUE                                                        C2123400
        NEQ = N                                                         C2123500
        IF(LEQU) PRINT 112, NEQ                                         C2123600
  100   FORMAT ('1 SUBROUTINE SETUP ENTERED ')                          C2123700
  102   FORMAT ('  YD(',I2,') = THACD(',I2,') = ',D20.8)                C2123800
  103   FORMAT ('  YD(',I2,') = THAC(',I2,') = ',D20.8)                 C2123900
  202   FORMAT ('  THAD(',I2,') = Y(',I2,') = ',D20.8)                  C2124000
  104   FORMAT ('  YD(',I2,') = THADD(',I2,') = ',D20.8)                C2124100
  204   FORMAT ('  HMOM(',I2,') = FLM(',I2,') * THACW(',I2,') = ',D20.8) C2124200
```

```
 205 FORMAT (*  THADW(*,I2,*) = Y(*,I2,*) = *,D20.8)                    C2124300
 106 FORMAT (*  YD(*,I2,*) = ANGD(*,I2,*) = *,D20.8)                    C2124400
 206 FORMAT (*  THA(*,I2,*) = Y(*,I2,*) = *,D20.8)                     C2124500
 106 FORMAT (*  YD(*,I2,*) = THADW(*,I2,*) = *,D20.8)                  C2124600
 208 FORMAT (*  THA*(*,I2,*) = Y(*,I2,*) = *,D20.8)                    C2124700
 111 FORMAT (*  YD(*,I2,*) = YMCD(*,I2,*,*,I2,*,*,I2,*) = *,D20.8)     C2124800
 211 FORMAT (*  XMC(*,I2,*,*,I2,*,*,I2,*) = Y(*,I2,*) = *,D20.8)      C2124900
 112 FORMAT (*  TOTAL NUMBER OF EQUATIONS, NEQ = *,I5)                 C2125000
     RETURN                                                           C2125100
     END                                                              C2125200


C                                                                     C2200000
      SUBROUTINE OUTPUT                                               C2200100
C       GENERAL TYPE OUTPUT FOR NO PARTICULAR SATELLITE               C2200200
C                                                                     C2200300
C                                                                     C2200400
      IMPLICIT REAL*8(A-H,O-Z,$)                                      C2200500
      LOGICAL  FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)   C2200600
C                                                                     C2200700
C                                                                     C2200800
      INTEGER                                                         C2200900
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCON  , FCON  ,  C2201000
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C2201100
     * SI     , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C2201200
     * SMA    , SMCDUM, SMC   , SMV   , SOK   , SPIDUM, SPI   , SQF   ,  C2201300
     * SGL    , SR    , SSCN  , SSIX  , SVA   , SVE   , SVD   , SVI   ,  C2201400
     * SVM    , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  C2201500
     * SC     , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  C2201600
     * IINIT(1)       , IZINIT(1)     , SD                             C2201700
C                                                                     C2201800
C                                                                     C2201900
C                                                                     C2202000
      REAL*8                                                          C2202100
     * ANGD  (33)    , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C2202200
     * FLQ   (3,20)  , FLE   (3,3,20), FLH   (3,3,20),               C2202300
     * THADD (33)    , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)     C2202400
C                                                                     C2202500
C                                                                     C2202600
C                                                                     C2202700
C                                                                     C2202800
C                                                                     C2202900
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RBLO(10)         C2203000
C                                                                     C2203100
C                                                                     C2203200
      COMMON /INTG/    AWORK (200)   ,                                C2203300
     * CT1             , CT2           , CT3           , CT4         ,  C2203400
     * CT5             , FCON  (33)    , JCON  (10)    , LCON  (22)  ,  C2203500
     * MC    (10)      , NB1           , NBOD          , NCTC        ,  C2203600
     * NFER            , NFKC          , NFRC          , NLOR        ,  C2203700
     * NMV             , NMO           , NMOAXB OUT32  , NSVP        ,  C2203800
     * NSVC            , PCON  (11)    , SO            , SFR   (33)  ,  C2203900
     * SG              , SI    (55)    , SIG           , SL          ,  C2204000
     * SLK   (33)      , SMA   (10)    , SOK   (11)    , SQF   (11)  ,  C2204100
     * SGL   (11)      , SMV           , SR            , SSCN        ,  C2204200
     * SSIX            , SVA           , SVB           , SVD         ,  C2204300
     * SVI             , SVM           , SVP   (22)    , SVQ   (33)  ,  C2204400
```

```
     *  SXM    (3,10)  , SXT             , TORQ   (97)     , SMAL           ,  C2204500
     *  SEU            , NTQ             , SC     (33)     , SCG            ,  C2204600
     *  NFLXB          , SFLX            , SFXM   (10)     , NMCDS          ,  C2204700
     *  SFCC           , SCC    (10)                                           C2204800
C                                                                              C2204900
C                                                                              C2205000
      COMMON /INTGZ/                                                           C2205100
     *  SCNDUM         , SCN    (9)      , SCRDUM         , SCR    (9)      ,  C2205200
     *  SFKDUM         , SFK    (9)      , SIXDUM         , SIX    (9)      ,  C2205300
     *  SKDUM          , SK     (9)      , SPIDUM         , SPI    (9)      ,  C2205400
     *  SMCDUM         , SMC    (9)                                           C2205500
C                                                                              C2205600
C                                                                              C2205700
      COMMON /REAL/                                                            C2205800
     *  CA     (3,10)  , CAC    (3,10)   , CLM    (10)     , CCMC   (3,11)  ,  C2205900
     *  DCMC   (3,11)  , ETC    (3,11)   , ETM    (33)     , FOMC   (3,11)  ,  C2206000
     *  GAM    (3,66)  , H               , HM     (3,10)   , HMC    (3,10)  ,  C2206100
     *  HMOM   (10)    , PHI    (3,11)   , PLM    (10)     , QF     (3,33)  ,  C2206200
     *  QFC    (3,33)  , QL     (3,22)   , QLC    (3,22)   , ROMC   (3,11)  ,  C2206300
     *  T              ,                  THA    (33)     , THAD   (33)     ,  C2206400
     *  THADW  (10)    , THAW   (10)     , XDIC   (3,3,66) , XI     (3,3,10),  C2206500
     *  XIC    (3,3,10), XMAS   (10)     , XMN    (33,33)  , XMT    (3,3,10),  C2206600
     *  TLG    (33)    , FLA    (3,20)   , FLB    (3,20)   , FLC    (3,20)   ,  C2206700
     *  FLD    (3,3,20), FLJ    (3,3,20) , CAO    (3,10)   , XIC    (3,3,10),  C2206800
     *  FLIRC  (3,10)  , FLCRC  (3,10)   , FLAC   (3,20)   , FLQC   (3,20)   ,  C2206900
     *  FLCM   (20)    , ZETA   (20)     , FCF    (3,3,40) , FCK    (3,40)   ,  C2207000
     *  TIMEND                                                                  C2207100
C                                                                              C2207200
C                                                                              C2207300
      COMMON /REALZ/                                                           C2207400
     *  CEDUM  (1,3)   , CB     (3,10)   , CBCDUM(1,3)    , CBC    (3,10)   ,  C2207500
     *  XMCDUM(1,1,9)  , XMC    (3,3,10) , CBN(3)                             C2207600
C                                                                              C2207700
C                                                                              C2207800
      EQUIVALENCE (ETM(1),THADD(1))          ,(XMN(1,1),ANGD(1))      ,       C2207900
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,       C2208000
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,       C2208100
     *            (FLB(1,1),FLQ(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),        C2208200
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                                C2208300
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,       C2208400
     *            (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))     ,       C2208500
     *            (SCNDUM,IZINIT(1))                                          C2208600
C                                                                              C2208700
C                                                                              C2208800
      INTEGER TIK                                                              C2208900
      DIMENSION PHII(3,3)                                                      C2209000
      DIMENSION TOMC(3,11)                                                     C2209100
      DIMENSION SYSCM(3)                                                       C2209200
      DIMENSION SYSIN(3,3)                                                     C2209300
      DIMENSION HB(3,10),TK(10)                                                C2209400
      DIMENSION POS(3,10),VEL(3,10)                                            C2209500
      DIMENSION TEM1(3),TEM2(3),TEM3(3),DERV(3,11)                             C2209600
      DIMENSION TEM4(3)                                                        C2209700
      DIMENSION HBODY(3),          RINERT(3)                                   C2209800
      DIMENSION EPO(3,10),DHM(3,10)                                            C2209900
      DIMENSION EP(3),EI(3,3),EIC(3,3),FOD(3),FODC(3),TEM5(3,3)                C2210000
      REAL*8 LM(3,10),LMT(3)                                                   C2210100
      INTEGER SET(10),SFXMK                                                    C2210200
C                                                                              C2210300
C                                                                              C2210400
```

```
    1 CONTINUE                                                          C2210500
      PRINT 203, T                                                      C2210600
C                                                                       C2210700
C                                                                       C2210800
C     COMPUTE SYSTEM COMPOSITE CENTER OF MASS                           C2210900
      TOTM = C.DO                                                       C2211000
      DO 11  I=1,3                                                      C2211100
   11 TEM1(I) = 0.DO                                                    C2211200
      DO 15  K=1,NBOD                                                   C2211300
      KO = KTO(NB1,0,K)                                                 C2211400
      CALL SCLV(XMAS(K),CAM(1,KO),TEM2)                                 C2211500
      CALL VECADD(TEM1,TEM2,TEM1)                                       C2211600
   15 TOTM = TOTM + XMAS(K)                                             C2211700
      DO 16  I=1,3                                                      C2211800
   16 SYSCM(I) = TEM1(I)/TOTM                                           C2211900
      PRINT 200, (SYSCM(I),I=1,3)                                       C2212000
      PRINT 222, TOTM                                                   C2212100
C                                                                       C2212200
C                                                                       C2212300
C     COMPUTE SYSTEM INERTIA TENSOR ABOUT COMPOSITE CENTER OF MASS      C2212400
      CALL SUBCP(SYSCM,SYSCM,TOTM,SYSIN)                                C2212500
      DO 4   I=1,3                                                      C2212600
      DO 4   J=1,3                                                      C2212700
    4 SYSIN(I,J) = XDIC(I,J,1) - SYSIN(I,J)                             C2212800
      PRINT 211, (SYSIN(1,J),J=1,3)                                     C2212900
      PRINT 212, (SYSIN(2,J),J=1,3)                                     C2213000
      PRINT 211, (SYSIN(3,J),J=1,3)                                     C2213100
      PRINT 213                                                         C2213200
C                                                                       C2213300
C                                                                       C2213400
C     COMPUTE VECTOR FROM INERTIAL ORIGIN TO COMPOSITE SYSTEM           C2213500
C      CENTER OF MASS RELATIVE TO COMPUTING FRAME                       C2213600
      CALL VECADD(CBC(1,1),SYSCM,CBC(1,0))                              C2213700
C                                                                       C2213800
C                                                                       C2213900
C                                                                       C2214000
C     FLEXIBILITY RELATED PARAMETERS                                    C2214100
      MN = 0                                                            C2214200
      DO 30 K=1,NBOD                                                    C2214300
      DO 32 I=1,3                                                       C2214400
      EPD(I,K) = 0.                                                     C2214500
   32 DHM(I,K) = 0.                                                     C2214600
      IF(SFXM(K).EQ.0) GO TO 30                                         C2214700
      SFXMK = SFXM(K)                                                   C2214800
      DO 31 I=1,SFXMK                                                   C2214900
      MN = MN+1                                                         C2215000
      NFMN = NFER + MN                                                  C2215100
      CALL SCLV(THAD(NFMN),FLAC(1,MN),TEM1)                             C2215200
      CALL VECADD(EPD(1,K),TEM1,EPD(1,K))                               C2215300
      CALL SCLV(THAD(NFMN),FLQC(1,MN),TEM1)                             C2215400
      CALL VECADD(DHM(1,K),TEM1,DHM(1,K))                               C2215500
   31 CONTINUE                                                          C2215600
   30 CONTINUE                                                          C2215700
C                                                                       C2215800
C                                                                       C2215900
C                                                                       C2216000
C     COMPUTE INERTIAL ANGULAR MOMENTUM AND KINETIC ENERGY OF EACH      C2216100
C       BODY AND OF THE COMPOSITE SYSTEM                                C2216200
      TKIN = C.DO                                                       C2216300
      DO 7   I=1,3                                                      C2216400
```

```
      LMT(I) = 0.D0                                                    C2216500
    7 HBODY(I) = 0.D0                                                  C2216600
      DO 17 KKK=1,NBOD                                                 C2216700
      K=NBOD-(KKK-1)                                                   C2216800
      KK = K                                                           C2216900
      JK = JCON(K)                                                     C2217000
      IF(KK.NE.1) GO TO 28                                             C2217100
      DO 29  I=1,3                                                     C2217200
   29 TEM1(I) = ROMC(I,NB1)                                            C2217300
      GO TO 26                                                         C2217400
   28 CONTINUE                                                         C2217500
C     COMPUTE LINEAR VELOCITY OF CENTER OF MASS OF BODY K PUT IN TEM1  C2217600
      IF(RELB(K)) GO TO 19                                             C2217700
      CALL VECROS (FCMC(1,JK),CAC(1,K),TEM1)                           C2217800
      CALL VECADD(RCMC(1,K),TEM1,TEM1)                                 C2217900
      GO TO 24                                                         C2218000
   19 CALL VECROS (FCMC(1,K),CAC(1,K),TEM1)                            C2218100
   24 CALL VECADD(RCMC(1,NB1),TEM1,TEM1)                               C2218200
C     CHECK FOR END OF CHAIN                                           C2218300
   25 IF(JK.EQ.0) GO TO 26                                             C2218400
      CALL VECROS (FCMC(1,JK),CBC(1,KK),TEM2)                          C2218500
      CALL VECADD(TEM1,TEM2,TEM1)                                      C2218600
      KK = JK                                                          C2218700
      JK = JCON(KK)                                                    C2218800
      GO TO 25                                                         C2218900
   26 CONTINUE                                                         C2219000
C                                                                      C2219100
C     INERTIAL POSITION OF CENTER OF MASS IN TEM2                      C2219200
      KL = KTC(NB1,0,K)                                                C2219300
      CALL VECADD(CBC(1,1),GAM(1,KL),TEM2)                             C2219400
      DO 3  I=1,3                                                      C2219500
      POS(I,K) = TEM2(I)                                               C2219600
    3 VEL(I,K) = TEM1(I)                                               C2219700
C                                                                      C2219800
C     ADD RELATIVE VELOCITY OF CM, NON-ZERO IF BODY FLEXIBLE           C2219900
      CALL VECADD(VEL(1,K),EPD(1,K),VEL(1,K))                          C2220000
      DO 13 I=1,3                                                      C2220100
   13 TEM1(I) = VEL(I,K)                                               C2220200
C                                                                      C2220300
C                                                                      C2220400
C     START COMPUTATION OF ANGULAR MOMENTUM, LINEAR MOMENTUM AND KINETIC2220500
      CALL DYADTV(XIC(1,1,K),FCMC(1,K),HB(1,K))                        C2220600
      CALL VECCOT(FCMC(1,K),HE(1,K),TK(K))                             C2220700
      CALL VECCOT(TEM1,TEM1,TEM)                                       C2220800
      TK(K) = .5D0*(TK(K) + XMAS(K)*TEM)                               C2220900
      CALL VECROS (TEM2,TEM1,TEM3)                                     C2221000
      CALL SCLV(XMAS(K),TEM3,TEM3)                                     C2221100
      CALL VECADD(HB(1,K),TEM3,HB(1,K))                                C2221200
      IF(.NOT.RELB(K)) GO TO 27                                        C2221300
      IF(NMO.EQ.0) GO TO 5068                                          C2221400
      DO 27  M=1,NMO                                                   C2221500
      IF(MC(M).NE.K) GO TO 27                                          C2221600
      CALL SCLV(HMOM(M),HMC(1,M),TEM3)                                 C2221700
      CALL VECADD(HB(1,K),TEM3,HB(1,K))                                C2221800
      TK(K) = TK(K) + .5D0*HMOM(M)**2/PLM(M)                           C2221900
   27 CONTINUE                                                         C2222000
 5068 CONTINUE                                                         C2222100
C                                                                      C2222200
C                                                                      C2222300
C     ADD FLEXIBILITY ADDITIONS TO MOMENTUM AND ENERGY                 C2222400
```

```
      CALL VECADD(HB(1,K),DHM(1,K),HB(1,K))                              C2222500
      CALL VECDOT(FCMC(1,K),DHM(1,K),TEM)                                C2222600
      TK(K) = TK(K) + TEM                                                C2222700
C                                                                        C2222800
C                                                                        C2222900
C     ADD UP FOR SYSTEM ANGULAR MOMENTUM AND KINETIC ENERGY              C2223000
      TKIN = TKIN + TK(K)                                                C2223100
      CALL VECADD(HB(1,K),HBODY,HBODY)                                   C2223200
      CALL SCLV(XMAS(K),VEL(1,K),LM(1,K))                                C2223300
      CALL VECADD(LMT,LM(1,K),LMT)                                       C2223400
   17 CONTINUE                                                           C2223500
      CALL TRNSPS  (XMC(1,1,0))                                          C2223600
      CALL VECTRN (HBODY,XMC(1,1,0),HINERT)                              C2223700
      CALL TRNSPS  (XMC(1,1,0))                                          C2223800
      IF(.NOT.INERF) GO TO 2                                             C2223900
      CALL TRNSPS  (XMC(1,1,1))                                          C2224000
      CALL VECTRN (HINERT,XMC(1,1,1),HBODY)                              C2224100
      CALL TRNSPS  (XMC(1,1,1))                                          C2224200
    2 CONTINUE                                                           C2224300
      HMG = DSQRT(HINERT(1)**2 + HINERT(2)**2 + HINERT(3)**2)            C2224400
      PRINT 209, HMG                                                     C2224500
      PRINT 201, (HBODY(I),I=1,3)                                        C2224600
      PRINT 202, (HINERT(I),I=1,3)                                       C2224700
      A = DSQRT(LMT(1)**2 + LMT(2)**2 + LMT(3)**2)                       C2224800
      PRINT 217, A                                                       C2224900
      PRINT 219, (LMT(I),I=1,3)                                          C2225000
      PRINT 215, TKIN                                                    C2225100
C                                                                        C2225200
C                                                                        C2225300
C     COMPUTE INERTIAL ACCELERATIONS                                     C2225400
      DO 8  I=1,3                                                        C2225500
    8 TOMC(I,1) = DOMC(I,1)                                              C2225600
      IF(NECD.EQ.1) GO TO 5001                                           C2225700
      DO 14  K=2,NBOD                                                    C2225800
      CALL VECROS (FCMC(1,K),RCMC(1,K),TEM1)                             C2225900
   14 CALL VECSUB(DOMC(1,K),TEM1,TCMC(1,K))                              C2226000
 5001 CONTINUE                                                           C2226100
      CALL VECROS (FOMC(1,1),RCMC(1,NB1),TEM1)                           C2226200
      CALL VECSUB(DOMC(1,NB1),TEM1,TCMC(1,NB1))                          C2226300
      M = 1                                                              C2226400
      DO 20  K=1,NB1                                                     C2226500
      IF(K.EQ.1) GO TO 21                                                C2226600
      M = M+3-PCON(K-1)                                                  C2226700
   21 DO 22  I=1,3                                                       C2226800
   22 TEM1(I) = 0                                                        C2226900
      MMTERM=M+2-PCON(K)                                                 C2227000
      IF(M.GT.MMTERM) GO TO 20                                           C2227100
      DO 23 MM=M,MMTERM                                                  C2227200
      CALL SCLV(THADD(MM),QFC(1,MM),TEM2)                                C2227300
   23 CALL VECADD(TEM1,TEM2,TEM1)                                        C2227400
   20 CALL VECADD(TEM1,TCMC(1,K),DERV(1,K))                              C2227500
C                                                                        C2227600
C                                                                        C2227700
      DO 5 K=1,NBOD                                                      C2227800
      PRINT 204, K,(ROMC(I,K),I=1,3),(FOMC(I,K),I=1,3)                   C2227900
    5 PRINT 207, (DERV(I,K),I=1,3)                                       C2228000
      PRINT 223, (ETC(I,K),I=1,3),(PHI(I,K),I=1,3)                       C2228100
      PRINT 205, (CAC(I,K),I=1,3),(CDC(I,K),I=1,3)                       C2228200
      PRINT 216, (PCS(I,K),I=1,3),(VEL(I,K),I=1,3)                       C2228300
      PRINT 206, ((XMC(I,J,K),J=1,3),(XIC(I,J,K),J=1,3),I=1,3)           C2228400
```

```
      PRINT 214, (HB(J,K),J=1,3),TK(K)                                  C2228500
      PRINT 21E, (LM(J,K),J=1,3)                                        C2228600
    5 CONTINUE                                                          C2228700
      PRINT 20E, (FOMC(I,NB1),I=1,3),(DERV(I,NB1),I=1,3)                C2228800
      PRINT 223, (ETC(I,NB1),I=1,3),(PHI(I,NB1),I=1,3)                  C2228900
      PRINT 224, (CBC(I,0),I=1,3)                                       C2229000
      PRINT 221, ((XMC(I,J,0),J=1,3),I=1,3)                             C2229100
      PRINT 213                                                         C2229200
      IF(NMO.EQ.0) GO TO 41                                             C2229300
      DO 6 M=1,NMO                                                      C2229400
      PRINT 21C, M,HMOM(M),M,CLM(M)                                     C2229500
    6 CONTINUE                                                          C2229600
   41 CONTINUE                                                          C2229700
      PRINT 213                                                         C2229800
      IF(NMODS.EQ.0) GO TO 40                                           C2229900
      CALL UNPAC(SET,NSET,SFLX)                                         C2230000
      MN = 0                                                            C2230100
      DO 33 KK=1,NSET                                                   C2230200
      K = SET(NSET+1-KK)                                                C2230300
      DO 34 I=1,3                                                       C2230400
      EP(I) = 0                                                         C2230500
      FQD(I) = 0                                                        C2230600
      FQDC(I) = 0                                                       C2230700
      DO 34 J=1,3                                                       C2230800
      EI(I,J) = 0                                                       C2230900
      EID(I,J) = 0                                                      C2231000
   34 CONTINUE                                                          C2231100
      SFXMK = SFXM(K)                                                   C2231200
      DO 35 M=1,SFXMK                                                   C2231300
      MN = MN+1                                                         C2231400
      NFMN = NFER+MN                                                    C2231500
      CALL SCLV(THA(NFMN),FLA(1,MN),TEM1)                               C2231600
      CALL VECADD(EP,TEM1,EP)                                           C2231700
      CALL SCLV(THAD(NFMN),FLQ(1,MN),TEM1)                              C2231800
      CALL VECADD(FQD,TEM1,FQD)                                         C2231900
      CALL SCLD(THA(NFMN),FLE(1,1,MN),TEM5)                             C2232000
      CALL DYADD(EI,TEM5,EI)                                            C2232100
      CALL SCLD(THAD(NFMN),FLE(1,1,MN),TEM5)                            C2232200
      CALL DYADD(EID,TEM5,EID)                                          C2232300
   35 CONTINUE                                                          C2232400
      PRINT 226, (EP(I),I=1,3),(EPD(I,K),I=1,3)                         C2232500
      PRINT 226, K,(FQD(I),I=1,3)                                       C2232600
      PRINT 229, ((EI(I,J),J=1,3),(EID(I,J),J=1,3),I=1,3)               C2232700
      PRINT 230, (FLIRC(I,K),I=1,3),(FLCRC(I,K),I=1,3)                  C2232800
      PRINT 231,(DHM(I,K),I=1,3)                                        C2232900
   33 CONTINUE                                                          C2233000
      PRINT 213                                                         C2233100
   40 CONTINUE                                                          C2233200
      DO 9 I=1,NFER                                                     C2233300
    9 PRINT 220, (I,THA(I),I,THAD(I),I,THADD(I),I,(QFC(J,I),J=1,3))     C2233400
      N1 = NFER+1                                                       C2233500
      N2 = NFER+MN                                                      C2233600
      IF(MN.EQ.0) RETURN                                                C2233700
      DO 12 I=N1,N2                                                     C2233800
   12 PRINT 225,  I,THA(I),I,THAD(I),I,THADD(I)                         C2233900
  200 FORMAT (' CENTER OF MASS =',3D17.8,/)                             C2234000
  201 FORMAT (/,3X,'HBODY     = ',3D17.8)                               C2234100
  202 FORMAT (3X,'HINERT    = ',3D17.8)                                 C2234200
  203 FORMAT ('1 TIME = ',D15.8,/)                                      C2234300
  204 FORMAT(/,3X,'BODY ',I2,4X,'FOMC =',3D17.8,3X,'FCMC =',3D17.8)     C2234400
```

```
 205 FORMAT (14X,'CAC = ',3D17.8,3X,'CBC = ',3D17.8)                    C2234500
 206 FORMAT (14X,'XMC = ',3D17.8,3X,'XIC = ',3D17.8)                    C2234600
 207 FORMAT (14X,'ACC = ',3D17.8)                                       C2234700
 208 FORMAT (/,3X,'ORIGIN',4X,'FCMC = ',3D17.8,3X,'ACC = ',3D17.8)      C2234800
 209 FORMAT ('  ANGULAR MOMENTUM = ',D20.8)                            C2234900
 210 FORMAT (3X,'HMOM(',I2,') = ',D17.8,10X,'CLM(',I2,') = ',D17.8)     C2235000
 211 FORMAT (25X,3D17.8)                                                C2235100
 212 FORMAT ('  SYSTEM INERTIA TENSOR =',3D17.8)                        C2235200
 213 FORMAT ('  ')                                                      C2235300
 214 FORMAT(14X,' MB = ',3D17.8,3X,' TK = ',D17.8)                      C2235400
 215 FORMAT (/,'  KINETIC ENERGY = ',D20.8)                            C2235500
 216 FORMAT (14X,'PCS = ',3D17.8,3X,'VEL = ',3D17.8)                    C2235600
 217 FORMAT (/,'  LINEAR MOMENTUM = ',D20.8)                           C2235700
 218 FORMAT (14X,' LM = ',3D17.8)                                       C2235800
 219 FORMAT (/,3X,'LBODY     = ',3D17.8)                                C2235900
 220 FORMAT (3X,'THA(',I2,') =',D13.6,3X,'THAD(',I2,') =',D13.6,3X,'THAC2236000
    *CD(',I2,') =',D13.6,3X,'GFC(',I2,') =',3D13.6)                     C2236100
 221 FORMAT (14X,'XMC = ',3D17.8)                                       C2236200
 222 FORMAT ('   TOTAL SYSTEM MASS =',D17.8,/)                          C2236300
 223 FORMAT (14X,'ETC = ',3D17.8,3X,'PHI = ',3D17.8)                    C2236400
 224 FORMAT (14X,'CBC = ',3D17.8)                                       C2236500
 225 FORMAT (3X,'THA(',I2,') =',D13.6,3X,'THAD(',I2,') =',D13.6,3X,'THAC2236600
    *CD(',I2,') =',D13.6)                                               C2236700
 226 FORMAT (/,3X,'FLEXIBLE',3X,' EP = ',3D17.8,3X,'EPP = ',3D17.8)     C2236800
 228 FORMAT (3X,' BODY',I2,4X,' CD = ',3D17.8)                          C2236900
 229 FORMAT (14X,' EI = ',3D17.8,3X,'EID = ',3D17.8)                    C2237000
 230 FORMAT (14X,'FIR = ',3D17.8,3X,'FCR = ',3D17.8)                    C2237100
 231 FORMAT (14X,'DHM = ',3D17.8)                                       C2237200
     RETURN                                                             C2237300
     END                                                                C2237400
```

```
C                                                                       C2300000
      SUBROUTINE OUTPSP                                                 C2300100
C     *GENERAL TYPE OUTPUT FOR NO PARTICULAR SATELLITE                  C2300200
C                                                                       C2300300
C                                                                       C2300400
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2300500
      LOGICAL* FG1, FG2, FG3, FG4, FG5, INERF, RBLO, LEQU, LINIT(1)     C2300600
C                                                                       C2300700
C                                                                       C2300800
      INTEGER                                                           C2300900
     * AWORK , CT1   , CT2   , CT3   , CT4   , CT5   , FCCN  , PCCN  ,  C2301000
     * SCNDUM, SCN   , SCRDUM, SCR   , SFKDUM, SFK   , SFR   , SG    ,  C2301100
     * SI     , SIG   , SIXDUM, SIX   , SKDUM , SK    , SL    , SLK   ,  C2301200
     * SMA    , SMCDUM, SMC   , SMV   , SQK   , SPIDUM, SRI   , SQF   ,  C2301300
     * SCL    , SR    , SSCN  , SSIX  , SVA   , SVB   , SVD   , SVI   ,  C2301400
     * SVM    , SVP   , SVQ   , SXM   , SXT   , TORQ  , SMAL  , SEU   ,  C2301500
     * SC     , SCG   , NFLXB , SFLX  , SFXM  , NMODS , SFCC  , SCC   ,  C2301600
     * IINIT(1)       , IZINIT(1)      , SD                              C2301700
C                                                                       C2301800
C                                                                       C2301900
C                                                                       C2302000
      REAL*8                                                            C2302100
     * ANGC  (33)     , CNF   (3,10)  , ETIC  (3,10)  , ETMC  (3,10)  ,  C2302200
     * FLQ   (3,20)   , FLE   (3,3,20), FLH   (3,3,20),                 C2302300
     * THACD (33)     , YMCD  (3,2,11), RINIT (1)     , RZINIT(1)       C2302400
```

```
C                                                                       C2302500
C                                                                       C2302600
C                                                                       C2302700
C                                                                       C2302800
C                                                                       C2302900
      COMMON /LOGIC/ FG1, FG2, FG3, FG4, FG5, INERF, RELC(10)           C2303000
C                                                                       C2303100
C                                                                       C2303200
      COMMON /INTG/     AWORK (200)     ,                               C2303300
     * CT1             , CT2             , CT3             , CT4       ,C2303400
     * CT5             , FCON  (33)      , JCON  (10)      , LCON (22) ,C2303500
     * MC    (10)      , NEI             , NBOD            , NCTC      ,C2303600
     * NFER            , NFKC            , NFRC            , NLOR      ,C2303700
     * NMV             , NMO             , NMOA            , NSVP      ,C2303800
     * NSVC            , PCON  (11)      , SC              , SFR  (33) ,C2303900
     * SG              , SI    (55)      , SIG             , SL        ,C2304000
     * SLK   (33)      , SMA   (10)      , SCK   (11)      , SQF  (11) ,C2304100
     * SGL   (11)      , SMV             , SR              , SSCN      ,C2304200
     * SSIX            , SVA             , SVB             , SVC       ,C2304300
     * SVI             , SVM             , SVP   (22)      , SVQ  (33) ,C2304400
     * SXM   (3,10)    , SXT             , TORQ  (97)      , SMAL      ,C2304500
     * SEU             , NTQ             , SC    (33)      , SCG       ,C2304600
     * NFLXB           , SFLX            , SFXM  (10)      , NMODS     ,C2304700
     * SFCC            , SCC   (10)                                      C2304800
C                                                                       C2304900
C                                                                       C2305000
      COMMON /INTGZ/                                                    C2305100
     * SCNDUM          , SCN   (9)       , SCRDUM          , SCR  (9)  ,C2305200
     * SFKDUM          , SFK   (9)       , SIXDUM          , SIX  (9)  ,C2305300
     * SKDUM           , SK    (9)       , SPIDUM          , SPI  (9)  ,C2305400
     * SMCDUM          , SMC   (9)                                       C2305500
C                                                                       C2305600
C                                                                       C2305700
      COMMON /REAL/                                                     C2305800
     * CA    (3,10)    , CAC   (3,10)    , CLM   (10)      , CDMC (3,11),C2305900
     * DCMC  (3,11)    , ETC   (3,11)    , ETM   (33)      , FCMC (3,11),C2306000
     * GAM   (3,66)    , H               , HM    (3,10)    , HMC  (3,10),C2306100
     * HMCM  (10)      , PHI   (3,11)    , PLM   (10)      , QF   (3,33),C2306200
     * QFC   (3,33)    , QL    (3,22)    , QLC   (3,22)    , RDMC (3,11),C2306300
     * T               ,                   THA   (33)      , THAD (33)  ,C2306400
     * THADM (10)      , THAM  (10)      , XDIC  (3,3,66), XI   (3,3,10),C2306500
     * XIC   (3,3,10), XMAS  (10)      , XMN   (33,33)   , XMT  (3,3,10),C2306600
     * TLG   (33)      , FLA   (3,20)    , FLB   (3,20)    , FLC  (3,20) ,C2306700
     * FLD   (3,3,20), FLJ   (3,3,20), CAO   (3,10)    , XIC  (3,3,10),C2306800
     * FLIRC (3,10)    , FLCRC (3,10)    , FLAC  (3,20)    , FLQC (3,20) ,C2306900
     * FLCM  (20)      , ZETA  (20)      , FCF   (3,3,40), FCK  (3,40)  ,C2307000
     * TIMEND                                                            C2307100
C                                                                       C2307200
C                                                                       C2307300
      COMMON /REALZ/                                                    C2307400
     * CEDUM (1,3)     , CE    (3,10)    , CBCDUM(1,3)     , CBC  (3,10),C2307500
     * XMCDUM(1,1,9) , XMC   (3,3,10), CBN(3)                           C2307600
C                                                                       C2307700
C                                                                       C2307800
      EQUIVALENCE (ETM(1),THADC(1))          ,(XMN(1,1),ANGD(1))      ,C2307900
     *            (XMN(1,3),YMCD(1,1,1))     ,(XMN(1,6),CNF(1,1))     ,C2308000
     *            (XMN(1,8),ETIC(1,1))       ,(XMN(1,10),ETMC(1,1))   ,C2308100
     *            (FLB(1,1),FLQ(1,1))        ,(FLE(1,1,1),FLD(1,1,1)),C2308200
     *            (FLH(1,1,1),FLJ(1,1,1))    ,                        C2308300
     *            (FG1,LINIT(1))             ,(CA(1,1),RINIT(1))      ,C2308400
```

```
     *              (CBDUM(1,1),RZINIT(1))     ,(AWORK(1),IINIT(1))          C2308500
     *              (SCNDUM,IZINIT(1))                                       C2308600
C                                                                             C2308700
C                                                                             C2308800
C                                                                             C2308900
      DIMENSION PHII(3,3)                                                     C2309000
      DIMENSION TCMC(3,11)                                                    C2309100
      DIMENSION SYSCM(3)                                                      C2309200
      DIMENSION SYSIN(3,3)                                                    C2309300
      DIMENSION HB(3,10),TK(10)                                               C2309400
      DIMENSION POS(3,10),VEL(3,10)                                           C2309500
      DIMENSION TEM1(3),TEM2(3),TEM3(3),DERV(3,11)                            C2309600
      DIMENSION TEM4(3)                                                       C2309700
      DIMENSION HBODY(3),          PINERT(3)                                  C2309800
      DIMENSION EPD(3,10),DHM(3,10)                                           C2309900
      DIMENSION EP(3),EI(3,3),EID(3,3),FQD(3),FQDC(3),TEM5(3,3)               C2310000
      INTEGER SET(10),SFXMK                                                   C2310100
      REAL*8 LM(3,10),LMT(3)                                                  C2310200
      INTEGER TIK                                                             C2310300
C                                                                             C2310400
C                                                                             C2310500
    1 CONTINUE                                                                C2310600
      PRINT 203, T                                                            C2310700
C                                                                             C2310800
C                                                                             C2310900
C     COMPUTE SYSTEM COMPOSITE CENTER OF MASS                                 C2311000
      TOTM = 0.D0                                                             C2311100
      DO 11  I=1,3                                                            C2311200
   11 TEM1(I) = 0.D0                                                          C2311300
      DO 15  K=1,NBOD                                                         C2311400
      KO = KTC(NB1,0,K)                                                       C2311500
      CALL SCLV(XMAS(K),GAM(1,KC),TEM2)                                       C2311600
      CALL VECADD(TEM1,TEM2,TEM1)                                             C2311700
   15 TOTM = TOTM + XMAS(K)                                                   C2311800
      DO 16  I=1,3                                                            C2311900
   16 SYSCM(I) = TEM1(I)/TOTM                                                 C2312000
      PRINT 200, (SYSCM(I),I=1,3)                                             C2312100
      PRINT 222, TOTM                                                         C2312200
C                                                                             C2312300
C                                                                             C2312400
C     COMPUTE VECTOR FROM INERTIAL ORIGIN TO COMPOSITE SYSTEM                 C2312500
C       CENTER OF MASSRELATIVE TO COMPUTING FRAME                             C2312600
      CALL VECADD(CBC(1,1),SYSCM,CBC(1,0))                                    C2312700
C                                                                             C2312800
C                                                                             C2312900
C     COMPUTE SYSTEM INERTIA TENSOR ABOUT COMPOSITE CENTER OF MASS            C2313000
      CALL SUBOP(SYSCM,SYSCM,TOTM,SYSIN)                                      C2313100
      DO 4   I=1,3                                                            C2313200
      DO 4   J=1,3                                                            C2313300
    4 SYSIN(I,J) = XDIC(I,J,1) - SYSIN(I,J)                                   C2313400
      PRINT 211, (SYSIN(1,J),J=1,3)                                           C2313500
      PRINT 212, (SYSIN(2,J),J=1,3)                                           C2313600
      PRINT 211, (SYSIN(3,J),J=1,3)                                           C2313700
      PRINT 213                                                               C2313800
C                                                                             C2313900
C                                                                             C2314000
C     FLEXIBILITY RELATED PARAMETERS                                          C2314100
      MN = 0                                                                  C2314200
      DO 30 K=1,NBOD                                                          C2314300
      DO 32 I=1,3                                                             C2314400
```

```
      EPD(I,K) = 0.                                                    C2314500
   32 DHM(I,K) = 0.                                                    C2314600
      IF(SFXM(K).EQ.0) GO TO 30                                        C2314700
      SFXMK = SFXM(K)                                                  C2314800
      DO 31 I=1,SFXMK                                                  C2314900
      MN = MN+1                                                        C2315000
      NFMN = NFER + MN                                                 C2315100
      CALL SCLV(THAD(NFMN),FLAC(1,MN),TEM1)                            C2315200
      CALL VECADD(EPD(1,K),TEM1,EPD(1,K))                              C2315300
      CALL SCLV(THAD(NFMN),FLOC(1,MN),TEM1)                            C2315400
      CALL VECADD(DHM(1,K),TEM1,DHM(1,K))                              C2315500
   31 CONTINUE                                                         C2315600
   30 CONTINUE                                                         C2315700
C                                                                      C2315800
C                                                                      C2315900
C                                                                      C2316000
C     COMPUTE INERTIAL ANGULAR MOMENTUM AND KINETIC ENERGY OF EACH     C2316100
C        BODY AND OF THE COMPOSITE SYSTEM                              C2316200
      TKIN = 0.D0                                                      C2316300
      DO 7  I=1,3                                                      C2316400
      LMT(I) = 0.D0                                                    C2316500
    7 HBODY(I) = 0.D0                                                  C2316600
      DO 17 KKK=1,NBOD                                                 C2316700
      K=NBOD-(KKK-1)                                                   C2316800
      KK = K                                                           C2316900
      JK = JOIN(K)                                                     C2317000
      IF(KK.NE.1) GO TO 28                                             C2317100
      DO 29  I=1,3                                                     C2317200
   29 TEM1(I) = ROMC(I,NB1)                                            C2317300
      GO TO 26                                                         C2317400
   28 CONTINUE                                                         C2317500
C     COMPUTE LINEAR VELOCITY OF CENTER OF MASS OF BODY K PUT IN TEM1  C2317600
      IF(RELG(K)) GO TO 19                                             C2317700
      CALL VECROS (FOMC(1,JK),CAC(1,K),TEM1)                           C2317800
      CALL VECADD(ROMC(1,K),TEM1,TEM1)                                 C2317900
      GO TO 24                                                         C2318000
   19 CALL VECROS (FOMC(1,K),CAC(1,K),TEM1)                            C2318100
   24 CALL VECADD(ROMC(1,NB1),TEM1,TEM1)                               C2318200
C     CHECK FOR END OF CHAIN                                           C2318300
   25 IF(JK.EQ.0) GO TO 26                                             C2318400
      CALL VECROS (FOMC(1,JK),CBC(1,KK),TEM2)                          C2318500
      CALL VECADD(TEM1,TEM2,TEM1)                                      C2318600
      KK = JK                                                          C2318700
      JK = JOIN(KK)                                                    C2318800
      GO TO 25                                                         C2318900
   26 CONTINUE                                                         C2319000
C                                                                      C2319100
C     INERTIAL POSITION OF CENTER OF MASS IN TEM2                      C2319200
      KL = KTO(NB1,0,K)                                                C2319300
      CALL VECADD(CBC(1,1),GAM(1,KL),TEM2)                             C2319400
      DO 3  I=1,3                                                      C2319500
      POS(I,K) = TEM2(I)                                               C2319600
    3 VEL(I,K) = TEM1(I)                                               C2319700
C                                                                      C2319800
C                                                                      C2319900
C     ADD RELATIVE VELOCITY OF CM, NON-ZERO IF BODY FLEXIBLE,          C2320000
      CALL VECADD(VEL(1,K),EPD(1,K),VEL(1,K))                          C2320100
      DO 13 I=1,3                                                      C2320200
   13 TEM1(I) = VEL(I,K)                                               C2320300
C                                                                      C2320400
```

```
C     START COMPUTATION OF ANGULAR MOMENTUM, LINEAR MOMENTUM AND KINETIC C2320500
      CALL DYDCTV(XIC(1,1,K),FCMC(1,K),HB(1,K))                          C2320600
      CALL VECDOT(FCMC(1,K),HB(1,K),TK(K))                               C2320700
      CALL VECDOT(TEM1,TEM1,TEM)                                         C2320800
      TK(K) = .5D0*(TK(K) + XMAS(K)*TEM)                                 C2320900
      CALL VECROS (TEM2,TEM1,TEM3)                                       C2321000
      CALL SCLV(XMAS(K),TEM3,TEM3)                                       C2321100
      CALL VECADD(HE(1,K),TEM3,HB(1,K))                                  C2321200
      IF(.NOT.RBLO(K)) GO TO 27                                          C2321300
      IF(NMC.EQ.0) GO TO 5068                                            C2321400
      DO 27  M=1,NMO                                                     C2321500
      IF(MC(M).NE.K) GO TO 27                                            C2321600
      CALL SCLV(HMOM(M),HMC(1,M),TEM3)                                   C2321700
      CALL VECADD(HB(1,K),TEM3,HB(1,K))                                  C2321800
      TK(K) = TK(K) + .5D0*HMC(M)**2/PLM(M)                              C2321900
   27 CONTINUE                                                           C2322000
 5068 CONTINUE                                                           C2322100
C                                                                        C2322200
C                                                                        C2322300
C     ADD FLEXIBILITY ADDITIONS TO MOMENTUM AND ENERGY                   C2322400
      CALL VECADD(HB(1,K),DHM(1,K),HB(1,K))                              C2322500
      CALL VECDOT(FCMC(1,K),DHM(1,K),TEM)                                C2322600
      TK(K) = TK(K) + TEM                                                C2322700
C                                                                        C2322800
C                                                                        C2322900
C     ADD UP FOR SYSTEM ANGULAR MOMENTUM AND KINETIC ENERGY              C2323000
      TKIN = TKIN + TK(K)                                                C2323100
      CALL VECADD(HB(1,K),HBODY,HBODY)                                   C2323200
      CALL SCLV(XMAS(K),VEL(1,K),LM(1,K))                                C2323300
      CALL VECADD(LMT,LM(1,K),LMT)                                       C2323400
   17 CONTINUE                                                           C2323500
      CALL TRNSPS  (XMC(1,1,C))                                          C2323600
      CALL VECTRN (HBODY,XMC(1,1,C),HINERT)                              C2323700
      CALL TRNSPS  (XMC(1,1,C))                                          C2323800
      IF(.NOT.INERF) GO TO 2                                         V   C2323900
      CALL TRNSPS  (XMC(1,1,1))                                          C2324000
      CALL VECTRN (HINERT,XMC(1,1,1),HBODY)                              C2324100
      CALL TRNSPS  (XMC(1,1,1))                                          C2324200
    2 CONTINUE                                                           C2324300
      HMG = DSQRT(HINERT(1)**2 + HINERT(2)**2 + HINERT(3)**2)            C2324400
      PRINT 209, HMG                                                     C2324500
      PRINT 201, (HBODY(I),I=1,3)                                        C2324600
      PRINT 202, (HINERT(I),I=1,3)                                       C2324700
      A = DSQRT(LMT(1)**2 + LMT(2)**2 + LMT(3)**2)                       C2324800
      PRINT 217, A                                                       C2324900
      PRINT 215, (LMT(I),I=1,3)                                          C2325000
      PRINT 215, TKIN                                                    C2325100
C                                                                        C2325200
C                                                                        C2325300
C     COMPUTE INERTIAL ACCELERATIONS                                     C2325400
      DO 8  I=1,3                                                        C2325500
    8 TDMC(I,1) = DOMC(I,1)                                              C2325600
      IF(NECD.EQ.1) GO TO 5001                                           C2325700
      DO 14  K=2,NBOD                                                    C2325800
      CALL VECROS (FCMC(1,K),RCMC(1,K),TEM1)                             C2325900
   14 CALL VECSUB(DOMC(1,K),TEM1,TDMC(1,K))                              C2326000
 5001 CONTINUE                                                           C2326100
      CALL VECROS (FCMC(1,1),RCMC(1,NB1),TEM1)                           C2326200
      CALL VECSUB(DOMC(1,NB1),TEM1,TDMC(1,NB1))                          C2326300
      M = 1                                                              C2326400
```

```
      DO 20  K=1,NB1                                                    C2326500
      IF(K.EQ.1) GO TO 21                                               C2326600
      M = M+3-PCON(K-1)                                                 C2326700
   21 DO 22  I=1,3                                                      C2326800
   22 TEM1(I) = 0                                                       C2326900
      MMTERM=M+2-PCON(K)                                                C2327000
      IF(M.GT.MMTERM) GO TO 20                                          C2327100
      DO 23 MM=M,MMTERM                                                 C2327200
      CALL SCLV(THADD(MM),QFC(1,MM),TEM2)                               C2327300
   23 CALL VECADD(TEM1,TEM2,TEM1)                                       C2327400
   20 CALL VECADD(TEM1,TCMC(1,K),CERV(1,K))                             C2327500
C                                                                       C2327600
C                                                                       C2327700
      DO 5 K=1,NBOD                                                     C2327800
      PRINT 204, K,(ROMC(I,K),I=1,3),(FCMC(I,K),I=1,3)                  C2327900
   10 PRINT 207, (DERV(I,K),I=1,3)                                      C2328000
      PRINT 223, (ETC(I,K),I=1,3),(PHI(I,K),I=1,3)                      C2328100
      PRINT 205, (CAC(I,K),I=1,3),(CBC(I,K),I=1,3)                      C2328200
      PRINT 216, (POS(I,K),I=1,3),(VEL(I,K),I=1,3)                      C2328300
      PRINT 206, ((XMC(I,J,K),J=1,3),(XIC(I,J,K),J=1,3),I=1,3)          C2328400
      PRINT 214, (HB(J,K),J=1,3),TK(K)                                  C2328500
      PRINT 215, (LM(J,K),J=1,3)                                        C2328600
    5 CONTINUE                                                          C2328700
      PRINT 206, (FCMC(I,NB1),I=1,3),(DERV(I,NB1),I=1,3)                C2328800
      PRINT 223, (ETC(I,NB1),I=1,3),(PHI(I,NB1),I=1,3)                  C2328900
      PRINT 224, (CBC(I,C),I=1,3)                                       C2329000
      PRINT 221, ((XMC(I,J,0),J=1,3),I=1,3)                             C2329100
      PRINT 213                                                         C2329200
      IF(NMO.EQ.0) GO TO 41                                             C2329300
      DO 6 M=1,NMO                                                      C2329400
      PRINT 210, M,HMOM(M),M,CLM(M)                                     C2329500
    6 CONTINUE                                                          C2329600
      PRINT 213                                                         C2329700
   41 CONTINUE                                                          C2329800
      IF(NMODS.EQ.0) GO TO 40                                           C2329900
      CALL UNPAC(SET,NSET,SFLX)                                         C2330000
      MN = C                                                            C2330100
      DO 33 KK=1,NSET                                                   C2330200
      K = SET(NSET+1-KK)                                                C2330300
      DO 34 I=1,3                                                       C2330400
      EP(I) = C                                                         C2330500
      FQD(I) = 0                                                        C2330600
      FQDC(I) = 0                                                       C2330700
      DO 34 J=1,3                                                       C2330800
      EI(I,J) = 0                                                       C2330900
      EID(I,J) = 0                                                      C2331000
   34 CONTINUE                                                          C2331100
      SFXMK = SFXM(K)                                                   C2331200
      DO 35 M=1,SFXMK                                                   C2331300
      MN = MN+1                                                         C2331400
      NFMN = NFER+MN                                                    C2331500
      CALL SCLV(THA(NFMN),FLA(1,MN),TEM1)                               C2331600
      CALL VECADD(EP,TEM1,EP)                                           C2331700
      CALL SCLV(THAD(NFMN),FLQ(1,MN),TEM1)                              C2331800
      CALL VECADD(FQD,TEM1,FQD)                                         C2331900
      CALL SCLC(THA(NFMN),FLE(1,1,MN),TEM5)                             C2332000
      CALL CYADD(EI,TEM5,EI)                                            C2332100
      CALL SCLC(THAD(NFMN),FLE(1,1,MN),TEM5)                            C2332200
      CALL CYADD(EID,TEM5,EID)                                          C2332300
   35 CONTINUE                                                          C2332400
```

```
      PRINT 226, (EP(I),I=1,3),(EPD(I,K),I=1,3)                          C2332500
      PRINT 228, K,(FUD(I),I=1,3)                                        C2332600
      PRINT 229, ((EI(I,J),J=1,3),(EID(I,J),J=1,3),I=1,3)                C2332700
      PRINT 230, (FLIRC(I,K),I=1,3),(FLCRC(I,K),I=1,3)                   C2332800
      PRINT 231,(DHM(I,K),I=1,3)                                         C2332900
   32 CONTINUE                                                           C2333000
      PRINT 213                                                          C2333100
   40 CONTINUE                                                           C2333200
      DO 9 I=1,NPER                                                      C2333300
    9 PRINT 220, (I,THA(I),I,THAD(I),I,THADD(I),I,(QFC(J,I),J=1,3))      C2333400
      N1 = NPER+1                                                        C2333500
      N2 = NPER+MN                                                       C2333600
      IF(MN.EQ.0) RETURN                                                 C2333700
      DO 12 I=N1,N2                                                      C2333800
   12 PRINT 228,  I,THA(I),I,THAD(I),I,THADD(I)                          C2333900
 200 FORMAT (' CENTER OF MASS =',3D17.8,/)                               C2334000
 201 FORMAT (/,3X,'RBODY    = ',3D17.8)                                  C2334100
 202 FORMAT (3X,'RINERT   = ',3D17.8)                                    C2334200
 203 FORMAT ('1 TIME = ',D15.5,/)                                        C2334300
 204 FORMAT(/,3X,'BODY ',I2,4X,'RCMC =',3D17.8,3X,'FCMC =',3D17.8)       C2334400
 205 FORMAT (14X,'CAC = ',3D17.8,3X,'CBC = ',3D17.8)                     C2334500
 206 FORMAT (14X,'XMC = ',3D17.8,3X,'XIC = ',3D17.8)                     C2334600
 207 FORMAT (14X,'ACC = ',3D17.8)                                        C2334700
 208 FORMAT (/,3X,'ORIGIN',4X,'RCMC = ',3D17.8,3X,'ACC = ',3D17.8)       C2334800
 209 FORMAT ('  ANGULAR MOMENTUM = ',D20.8)                              C2334900
 210 FORMAT (3X,'HMCM(',I2,') = ',D17.8,10X,'CLM(',I2,') = ',D17.8)      C2335000
 211 FORMAT (25X,3D17.8)                                                 C2335100
 212 FORMAT ('  SYSTEM INERTIA TENSOR =',3D17.8)                         C2335200
 213 FORMAT ('  ')                                                       C2335300
 214 FORMAT(14X,' HB = ',3D17.8,3X,' TK = ',D17.8)                       C2335400
 215 FORMAT (/,'  KINETIC ENERGY = ',D20.8)                             C2335500
 216 FORMAT (14X,'PCS = ',3D17.8,3X,'VEL = ',3D17.8)                     C2335600
 217 FORMAT (/,'  LINEAR MOMENTUM = ',D20.8)                            C2335700
 218 FORMAT (14X,' LM = ',3D17.8)                                        C2335800
 219 FORMAT (/,3X,'LBODY    = ',3D17.8)                                  C2335900
 220 FORMAT (3X,'THA(',I2,') =',D13.6,3X,'THAD(',I2,') =',D13.6,3X,'THA  C2336000
    *DD(',I2,') =',D13.6,3X,'QFC(',I2,') =',3D13.6)                      C2336100
 221 FORMAT (14X,'XMC = ',3D17.8)                                        C2336200
 222 FORMAT ('   TOTAL SYSTEM MASS =',D17.8,/)                           C2336300
 223 FORMAT (14X,'ETC = ',3D17.8,3X,'PHI = ',3D17.8)                     C2336400
 224 FORMAT (14X,'CBC = ',3D17.8)                                        C2336500
 225 FORMAT (3X,'THA(',I2,') =',D13.6,3X,'THAD(',I2,') =',D13.6,3X,'THA  C2336600
    *DD(',I2,') =',D13.6)                                                C2336700
 226 FORMAT (/,3X,'FLEXIBLE',3X,' EP = ',3D17.8,3X,'EPP = ',3D17.8)      C2336800
 228 FORMAT (3X,' BODY',I2,4X,' QD = ',3D17.8)                           C2336900
 229 FORMAT (14X,' EI = ',3D17.8,3X,'EID = ',3D17.8)                     C2337000
 230 FORMAT (14X,'FIR = ',3D17.8,3X,'FCR = ',3D17.8)                     C2337100
 231 FORMAT (14X,'DHM = ',3D17.8)                                        C2337200
      RETURN                                                             C2337300
      END                                                                C2337400
```

```
C                                                                        C2400000
      SUBROUTINE SIMQ(D,C,N,NC)                                          C2400100
      IMPLICIT REAL*8(A-H,O-Z,$)                                         C2400200
      LOGICAL LEQU                                                       C2400300
      LOGICAL         LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,       C2400400
```

```
      *                  LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,      C2400500
      *                  LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C2400600
C                                                                           C2400700
      DIMENSION D(1),C(1)                                                   C2400800
      DIMENSION DD(33,33),CC(33)                                            C2400900
C     USED TO OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS     C2401000
C                       D*X = C                                             C2401100
C                                                                           C2401200
C      DESCRIPTION OF PARAMETERS                                            C2401300
C             D = COEFFICIENT MATRIX STORED COLUMN-WISE DESTROYED DURING    C2401400
C                 COMPUTATION                                             C2401500
C             N = NUMBER OF EQUATIONS                                       C2401600
C            ND = DIMENSION OF ARRAYS D AND C IN THE CALLING SUBROUTINE     C2401700
C             C = VECTOR OF ORIGINAL CONSTANTS OF LENGTH N, THESE ARE       C2401800
C                 REPLACED BY THE SOLUTION VECTOR                           C2401900
C                                                                           C2402000
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,         C2402100
      *                  LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,      C2402200
      *                  LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ         C2402300
      EQUIVALENCE (LSIMQ,LEQU)                                              C2402400
      IF(LEQU) PRINT 107                                                    C2402500
C                                                                           C2402600
C     TAKE ARRAY D FROM DIMENSION NDXND AS DEFINED IN CALLING SUBROUTINE    C2402700
C        TO AN NXN ARRAY                                                    C2402800
      IJ = 0                                                                C2402900
      DO 14   J=1,N                                                         C2403000
      J3 = (J-1)*ND                                                         C2403100
      DO 15    I=1,N                                                        C2403200
      IJ = IJ + 1                                                           C2403300
      D(IJ) = D(I+J3)                                                       C2403400
      IF(LEQU) DD(I,J) = D(IJ)                                              C2403500
   15 CONTINUE                                                              C2403600
      IF(LEQU) CC(J) = C(J)                                                 C2403700
   14 CONTINUE                                                              C2403800
C                                                                           C2403900
C     FORWARD SOLUTION                                                      C2404000
C                                                                           C2404100
      TOL = 0.D0                                                            C2404200
      KS = 0                                                                C2404300
      JJ = -N                                                               C2404400
      DO 65 J=1,N                                                           C2404500
      JY = J+1                                                              C2404600
      JJ = JJ+N+1                                                           C2404700
      BIGA = 0.D0                                                           C2404800
      IT = JJ-J                                                             C2404900
      DO 33   I=J,N                                                         C2405000
C                                                                           C2405100
C     SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                              C2405200
C                                                                           C2405300
      IJ = IT+I                                                             C2405400
      IF(DABS(BIGA) - DABS(D(IJ))) 21,33,33                                 C2405500
   21 BIGA = D(IJ)                                                          C2405600
      IMAX = I                                                              C2405700
   33 CONTINUE                                                              C2405800
C                                                                           C2405900
C     TEST FOR PIVOT LESS THAN TOLERANCE                                    C2406000
C                                                                           C2406100
      IF(DABS(BIGA) - TOL) 35,35,40                                         C2406200
   35 KS =1                                                                 C2406300
      PRINT 103                                                             C2406400
```

```
      STOP                                                              C2406500
      RETURN                                                            C2406600
C                                                                       C2406700
C     INTERCHANGE ROWS IF NECESSARY                                     C2406800
C                                                                       C2406900
   40 I1 = J + N*(J-2)                                                  C2407000
      IT = IMAX - J                                                     C2407100
      DO 50  K=J,N                                                      C2407200
      I1 = I1 + N                                                       C2407300
      I2 = I1 + IT                                                      C2407400
      SAVE = D(I1)                                                      C2407500
      D(I1) = D(I2)                                                     C2407600
      D(I2) = SAVE                                                      C2407700
C                                                                       C2407800
C     DIVIDE EQUATION BY LEADING COEFFICIENT                            C2407900
C                                                                       C2408000
   50 D(I1) = D(I1)/BIGA                                                C2408100
      SAVE  = C(IMAX)                                                   C2408200
      C(IMAX) = C(J)                                                    C2408300
      C(J) = SAVE/BIGA                                                  C2408400
C                                                                       C2408500
C     ELIMINATE NEXT VARIABLE                                           C2408600
C                                                                       C2408700
      IF(J-N) 55,70,55                                                  C2408800
   55 IQS = N*(J-1)                                                     C2408900
      DO 65  IX=JY,N                                                    C2409000
      IXJ = IQS + IX                                                    C2409100
      IT = J - IX                                                       C2409200
      DO 60  JX = JY,N                                                  C2409300
      IXJX = N*(JX-1) + IX                                              C2409400
      JJX = IXJX + IT                                                   C2409500
   60 D(IXJX) = D(IXJX) -(D(IXJ)*D(JJX))                                C2409600
   65 C(IX) = C(IX) - (C(J)*D(IXJ))                                     C2409700
C                                                                       C2409800
C     BACK SOLUTION                                                     C2409900
C                                                                       C2410000
   70 NY = N-1                                                          C2410100
      IT = N*N                                                          C2410200
      DO 80  J=1,NY                                                     C2410300
      IA = IT - J                                                       C2410400
      IB = N - J                                                        C2410500
      IC = N                                                            C2410600
      DO 80  K=1,J                                                      C2410700
      C(IB) = C(IB) - D(IA)*C(IC)                                       C2410800
      IA = IA-N                                                         C2410900
   80 IC = IC-1                                                         C2411000
C                                                                       C2411100
C                                                                       C2411200
      IF(.NOT. LEQU) GO TO 1000                                         C2411300
      TEST1 = 0.D0                                                      C2411400
      DO 1 I=1,N                                                        C2411500
      TEST2 = 0.D0                                                      C2411600
      DO 2  J=1,N                                                       C2411700
    2 TEST2 = TEST2 + DD(I,J)*C(J)                                      C2411800
      TEST2 = TEST2 - CC(I)                                             C2411900
      PRINT 100, I,TEST2                                                C2412000
      TEST1 = TEST1 + TEST2**2                                          C2412100
    1 CONTINUE                                                          C2412200
      TEST1 = DSQRT(TEST1)                                              C2412300
      PRINT 105, TEST1                                                  C2412400
```

```
 100C CONTINUE                                                          C2412500
C                                                                       C2412600
  1CC FORMAT ( 6D15.5)                                                  C2412700
  103 FORMAT (' MATRIX IS SINGULAR GARBAGE FOLLOWS ')                   C2412800
  104 FORMAT (' D(',I4,') = XMN(',I2,',',I2,') = ',D20.10)              C2412900
  105 FORMAT (' ')                                                      C2413000
  106 FORMAT (' C(',I2,') = ',D20.10)                                   C2413100
  107 FORMAT ('1  SUBROUTINE SIMQ ENTERED ')                            C2413200
  108 FORMAT (' ERROR IN ROW',I2,' = ',D17.8)                           C2413300
  109 FORMAT (' NORM OF ERROR VECTOR = ',D17.8)                         C2413400
      RETURN                                                            C2413500
      END                                                               C2413600


C                                                                       C2500000
      SUBROUTINE RUNGE(T,H,Y,YD,N1,N2,TEM)                              C2500100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2500200
      LOGICAL LEQU                                                      C2500300
      LOGICAL          LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2500400
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2500500
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ      C2500600
      COMMON /LDEBUG/ LRUNGE , LTRNSI , LVDIV  , LEQUIV , LTRAN  ,     C2500700
     *                 LTRANV , LRATE  , LXDY   , LETA   , LTORQU ,     C2500800
     *                 LQFDOT , LDCT   , LANGLE , LSETUP , LSIMQ      C2500900
      EQUIVALENCE (LRUNGE,LEQU)                                         C2501000
      DIMENSION Y(1),YD(1),TEM(2,1)                                     C2501100
C     ACCEPTS SYSTEM STATE AT TIME T                                    C2501200
C     RETURNS SYSTEM STATE AT TIME T+H                                  C2501300
C     SYSTEM STATE Y(N)                                                 C2501400
C     DERIVATIVE OF Y(N) IS YD(N)                                       C2501500
C     SUBROUTINE DYN COMPUTES YD(N) AS REQUIRED BY RUNGE KUTTA          C2501600
C     NUMBER OF STATE VARIABLES IS N1 + N2                              C2501700
C     N1 = NUMBER OF DYNAMIC EQUATIONS                                  C2501800
C     N2 = NUMBER OF EQUATIONS FROM TORQUE                              C2501900
C     TEMPORARY STORAGE AREA TEM(2,N) NOT TO BE USED IN DYN             C2502000
      N = N1 + N2                                                       C2502100
      IF(LEQU)PRINT 202                                                 C2502200
      IF(LEQU)PRINT 200,  (I,Y(I),I,YD(I),I,TEM(1,I),I,TEM(2,I),I=1,N)  C2502300
      K = 0                                                             C2502400
      DO 1  I=1,N                                                       C2502500
      TEM(1,I) = Y(I)                                                   C2502600
    1 TEM(2,I) = YD(I)                                                  C2502700
      CD = H/2                                                          C2502800
      A  = CD                                                           C2502900
    2 T = T + CD                                                        C2503000
    3 DO 4  I=1,N                                                       C2503100
    4 Y(I) = TEM(1,I)+A*YD(I)                                           C2503200
      IF(LEQU)PRINT 201,  (I,Y(I),I,YD(I),I,TEM(1,I),I,TEM(2,I),I=1,N)  C2503300
      CALL DYN(Y,YD,N1)                                                 C2503400
      IF(LEQU)PRINT 200,  (I,Y(I),I,YD(I),I,TEM(1,I),I,TEM(2,I),I=1,N)  C2503500
      K=K+1                                                             C2503600
      IF(K.EQ.3) GO TO 7                                                C2503700
      DO 5  I=1,N                                                       C2503800
    5 TEM(2,I) = TEM(2,I) + 2*YD(I)                                     C2503900
      IF(K.EQ.1) GO TO 3                                                C2504000
      A = H                                                             C2504100
      GO TO 2                                                           C2504200
```

```
    7 A = A/6                                                              C2504300
      DO 6  I=1,N                                                          C2504400
      TEM(2,I) = A*(TEM(2,I) + YD(I))                                      C2504500
    6 Y(I) = TEM(1,I) + TEM(2,I)                                           C2504600
      IF(LEQU)PRINT 201,  (I,Y(I),I,YD(I),I,TEM(1,I),I,TEM(2,I),I=1,N)     C2504700
      CALL DYN(Y,YD,N1)                                                    C2504800
      IF(LEQU)PRINT 200,  (I,Y(I),I,YD(I),I,TEM(1,I),I,TEM(2,I),I=1,N)     C2504900
  200 FORMAT ('  IN RUNGE Y(',I2,') = ',D18.8,' YD(',I2,') = ',D18.8,' TC2505000
     *EM(1,',I2,') = ',D18.8,' TEM(2,',I2,') = ',D18.8)                    C2505100
  201 FORMAT (' OUT RUNGE Y(',I2,') = ',D18.8,' YD(',I2,') = ',D18.8,' TC2505200
     *EM(1,',I2,') = ',D18.8,' TEM(2,',I2,') = ',D18.8)                    C2505300
  202 FORMAT ('1')                                                         C2505400
      RETURN                                                               C2505500
      END                                                                  C2505600
```

```
C                                                                          C2600000
      SUBROUTINE UNCAGE(SCG,SC,T,TUG)                                      C2600100
      IMPLICIT REAL*8(A-H,O-Z,$)                                           C2600200
      INTEGER SCG,SC(1),SCG1                                               C2600300
      DIMENSION TUG(1)                                                     C2600400
C      TEST TO SEE IF TIME TO UNCAGE ANY DEGREE OF FREEDOM                 C2600500
C        IF YES DO SO AND RENUMBER SC ARRAY                                C2600600
      SCG1 = SCG                                                           C2600700
      DO 6 I=1,SCG1                                                        C2600800
      IF(T.LT.TUG(SC(I))) GO TO 6                                          C2600900
      SCG = SCG-1                                                          C2601000
      PRINT 100, SC(I),T,SCG                                               C2601100
      SC(I) = 0                                                            C2601200
    6 CONTINUE                                                             C2601300
      IF(SCG.EQ.SCG1) RETURN                                               C2601400
      J=0                                                                  C2601500
      DO 7 I=1,SCG1                                                        C2601600
      IF(SC(I).EQ.0) GO TO 7                                               C2601700
      J=J+1                                                                C2601800
      SC(J)=SC(I)                                                          C2601900
      IF(I.EQ.J) GO TO 7                                                   C2602000
      SC(I) = 0                                                            C2602100
    7 CONTINUE                                                             C2602200
  100 FORMAT ('   MOTION ABOUT FREE VECTOR ',I2,' UNCAGED AT T =',E15.5,   C2602300
     *' MOTION STILL CAGED ABOUT ',I2,' FREE VECTORS ')                     C2602400
      RETURN                                                               C2602500
      END                                                                  C2602600
```

```
      SUBROUTINE COMPRS(XMN,THADD,N,SC,SCG,LG)                             C2602700
      IMPLICIT REAL*8(A-H,O-Z,$)                                           C2602800
      DIMENSION XMN(33,1),THADD(1)                                         C2602900
      INTEGER SC(1),SCG                                                    C2603000
      LOGICAL LG(1)                                                        C2603100
      SET UP LOGIC FLAGS                                                   C2603200
      NS = N + SCG                                                         C2603300
      DO 7 I=1,NS                                                          C2603400
    7 LG(I) = .TRUE.                                                       C2603500
```

```
      DO 8 I=1,SCG                                                    C2E03600
    8 LG(SC(I)) = .FALSE.                                             C2E03700
      II = 0                                                          C2E03800
      DO 9 I=1,NS                                                     C2E03900
      IF(.NOT.LG(I)) GO TO 9                                          C2E04000
      II = II+1                                                       C2E04100
      THADD(II) = THADD(I)                                            C2E04200
      JJ = 0                                                          C2E04300
      DO 10 J=1,NS                                                    C2E04400
      IF(.NOT.LG(J)) GO TO 10                                         C2E04500
      JJ = JJ+1                                                       C2E04600
      XMN(II,JJ) = XMN(I,J)                                           C2E04700
   10 CONTINUE                                                        C2E04800
    9 CONTINUE                                                        C2E04900
      RETURN                                                          C2E05000
      END                                                             C2E05100
```

```
      SUBROUTINE UNPRS(THADD,N,SCG,LG)                                C2E05200
      IMPLICIT REAL*8(A-H,O-Z,$)                                      C2E05300
      DIMENSION THADD(1)                                              C2E05400
      INTEGER SCG                                                     C2E05500
      LOGICAL LG(1)                                                   C2E05600
      I2 = N                                                          C2E05700
      NN = N+SCG                                                      C2E05800
      DO 11 I=1,NN                                                    C2E05900
      I1 = NN+1-I                                                     C2E06000
      IF(LG(I1)) GO TO 12                                             C2E06100
      THADD(I1) = 0.0                                                 C2E06200
      GO TO 11                                                        C2E06300
   12 THADD(I1) = THADD(I2)                                           C2E06400
      I2 = I2-1                                                       C2E06500
   11 CONTINUE                                                        C2E06600
      RETURN                                                          C2E06700
      END                                                             C2E06800
```

```
      SUBROUTINE COMPAC(SET,NSET,S)                                   C2E06900
      IMPLICIT REAL*8(A-H,O-Z,$)                                      C2E07000
      INTEGER  SET(NSET),A(24),S,AB                                   C2E07100
      DATA  A/ Z1       , Z2       , Z4       , Z8       ,           C2E07200
     *         Z10      , Z20      , Z40      , Z80      ,           C2E07300
     *         Z100     , Z200     , Z400     , Z800     ,           C2E07400
     *         Z1000    , Z2000    , Z4000    , Z8000    ,           C2E07500
     *         Z10000   , Z20000   , Z40000   , Z80000   ,           C2E07600
     *         Z100000  , Z200000  , Z400000  , Z800000/             C2E07700
      DATA AB/ Z1000000/                                              C2E07800
C00851953                                                             C2E07900
C     TAKES THE SET OF INTEGERS STORED IN SET(NSET) AND COMPACTS      C2E08000
C     THEN INTO THE SINGLE CODED INTEGER WORD S.  THE SET OF          C2E08100
C     INTEGERS IN ARRAY SET MUST BE DISTINCT FROM EACH OTHER          C2E08200
C     AND LIE BETWEEN 1 AND 24 INCLUSIVE.                             C2E08300
      S= NSET * AB                                                    C2E08400
      IF(NSET.EQ.0)GO TO 2                                            C2E08500
```

```
      DO 1  K=1,NSET                                                    C2608600
    1 S= S + A(SET(K))                                                  C2608700
      RETURN                                                            C2608800
    2 SET(1)=0                                                          C2608900
      RETURN                                                            C2609000
      END                                                               C2609100


      SUBROUTINE UNPAC(SET,NSET,S)                                      C2609200
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2609300
      INTEGER  SET(1),A(24),S,AB,TS                                     C2609400
      DATA  A/ Z1       , Z2       , Z4       , Z8       ,            C2609500
     *         Z10      , Z20      , Z40      , Z80      ,            C2609600
     *         Z100     , Z200     , Z400     , Z800     ,            C2609700
     *         Z1000    , Z2000    , Z4000    , Z8000    ,            C2609800
     *         Z10000   , Z20000   , Z40000   , Z80000   ,            C2609900
     *         Z100000  , Z200000  , Z400000  , Z800000/              C2610000
      DATA AB/ Z1000000/                                                C2610100
C                                                                       C2610200
C     DECODES THE CODED WORD S TO OBTAIN THE ELEMENTS OF SET(NSET)      C2610300
C     ELEMENTS OF SET RETURNED IN DECREASING ORDER OF MAGNITUDE         C2610400
C        SET(1).GT.SET(2).GT. .... .GT. SET(NSET)                       C2610500
      NSET = S/AB                                                       C2610600
      IF(NSET.EQ.0) GO TO 5                                             C2610700
      I=0                                                               C2610800
      TS= S-NSET*AB                                                     C2610900
      KN=25                                                             C2611000
      DO 1 K=1,24                                                       C2611100
      IF( TS-A(KN-K))  1,3,2                                            C2611200
    2 I=I+1                                                             C2611300
      SET(I) = KN-K                                                     C2611400
      TS = TS-A(KN-K)                                                   C2611500
    1 CONTINUE                                                          C2611600
    3 I=I+1                                                             C2611700
      SET(I)=KN-K                                                       C2611800
    4 RETURN                                                            C2611900
    5 SET(1)=0                                                          C2612000
      RETURN                                                            C2612100
      END                                                               C2612200


      INTEGER FUNCTION KTO(N,J,K)                                       C2612300
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2612400
      IF(K.GE.J) GO TO 1                                                C2612500
      KTO=K*(N-1)+J+1-K*(K-1)/2                                         C2612600
      GO TO 2                                                           C2612700
    1 KTO = J*(N-1) + K + 1 - J*(J-1)/2                                 C2612800
    2 RETURN                                                            C2612900
      END                                                               C2613000
```

```
      INTEGER FUNCTION KT1(N,J,K)                                       C2613100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2613200
      IF(K.GE.J) GO TO 1                                                C2613300
      KT1=(K-1)*(N-1)+J-(K-1)*(K-2)/2                                   C2613400
      GO TO 2                                                           C2613500
    1 KT1 = (J-1)*(N-1) + K - (J-1)*(J-2)/2                             C2613600
    2 RETURN                                                            C2613700
      END                                                               C2613800
```

```
      LOGICAL FUNCTION CTAIN(I,S,N)                                     C2613900
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2614000
      INTEGER S(N)                                                      C2614100
C                                                                       C2614200
C        CTAIN = .TRUE.  IF BODY LABEL I CONTAINED IN SET S(N)          C2614300
C              = .FALSE. IF NOT                                         C2614400
C     ELEMENTS IN SET(N) MUST BE POSITIVE NON-ZERO INTEGERS IN          C2614500
C        EITHER ASCENDING OR DECENDING ORDER OF MAGNITUDE               C2614600
C                                                                       C2614700
      IF(N.EQ.0.OR.I.EQ.0) GO TO 1                                      C2614800
      N1 = N+1                                                          C2614900
      IF(S(N) - S(1)) 2,2,3                                             C2615000
    2 K = N                                                             C2615100
      L = -1                                                            C2615200
      GO TO 4                                                           C2615300
    3 K = 1                                                             C2615400
      L = 1                                                             C2615500
    4 IF(S(K)-I) 5,6,1                                                  C2615600
    5 K = K+L                                                           C2615700
      IF(K.EQ.0.OR.K.EQ.N1) GO TO 1                                     C2615800
      GO TO 4                                                           C2615900
    1 CTAIN = .FALSE.                                                   C2616000
      RETURN                                                            C2616100
    6 CTAIN = .TRUE.                                                    C2616200
      RETURN                                                            C2616300
      END                                                               C2616400
```

```
C                                                                       C2700000
      SUBROUTINE VECTRN (VA,XMT,VAD)                                    C2700100
C     PERFORMS TRANSFORMATION OF COORDINATES FOR VECTORS                 C2700200
C        XMT = MATRIX, TRANSFORMS VECTORS TO COMPUTING FRAME COORDINATES C2700300
C                                      FROM BODY LAMBA FIXED COORDINAT   C2700400
C        VA = VECTOR INPUTED RELATIVE TO BODY LAMBA FIXED COORDINATES    C2700500
C       VAD = VECTOR COMPONENTS COMPUTED RELATIVE TO COMPUTING FRAME     C2700600
C                                                                       C2700700
C     CORRECT EQUATION SOLVED IS                                        C2700800
C                                                                       C2700900
C                          VAD = (XMT) * VA                             C2701000
C                                                                       C2701100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2701200
      DIMENSION VA(3),XMT(3,3),VAD(3)                                   C2701300
      DO 1 I=1,3                                                        C2701400
      VAD(I)=0                                                          C2701500
```

```
      CO 1 J=1,3                                                        C2701600
    1 VAC(I)=VAD(I)+XMT(I,J)*VA(J)                                      C2701700
      RETURN                                                            C2701800
      ENC                                                               C2701900


      SUBROUTINE TENTRN(XI,XMT,XID)                                     C2702000
C     TRANSFORM 3X3 TENSORS WITH CHECK FOR SYMMETRY                     C2702100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2702200
      DIMENSION XI(3,3),XMT(3,3),XID(3,3)                               C2702300
      LOGICAL FLAG                                                      C2702400
      FLAG = .TRUE.                                                     C2702500
      IF(      XI(1,2).EQ.XI(2,1)                                       C2702600
     *   .AND.XI(1,3).EQ.XI(3,1)                                        C2702700
     *   .AND.XI(2,3).EQ.XI(3,2)) GO TO 2                               C2702800
      FLAG = .FALSE.                                                    C2702900
    2 CO 1 I=1,3                                                        C2703000
      II = I                                                            C2703100
      IF(FLAG) GO TO 3                                                  C2703200
      II = 1                                                            C2703300
    3 CO 1 J=II,3                                                       C2703400
      XID(I,J) = 0                                                      C2703500
      CO 1  L=1,3                                                       C2703600
      CO 1  M=1,3                                                       C2703700
      XID(I,J) = XID(I,J) + XMT(I,L)*XI(L,M)*XMT(J,M)                   C2703800
    1 CONTINUE                                                          C2703900
      IF(.NOT.FLAG) RETURN                                              C2704000
      XID(2,1) = XID(1,2)                                               C2704100
      XID(3,1) = XID(1,3)                                               C2704200
      XID(3,2) = XID(2,3)                                               C2704300
      RETURN                                                            C2704400
      ENC                                                               C2704500


      SUBROUTINE VECNRM (V)                                             C2704600
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2704700
      DIMENSION V(3)                                                    C2704800
C      USED TO NORMALIZE VECTORS TO UNITY                               C2704900
      A = V(1)**2 + V(2)**2 + V(3)**2                                   C2705000
      IF(A.NE.0) GO TO 1                                                C2705100
      PRINT 100                                                         C2705200
  100 FORMAT (' GIMBAL LOCK CONDITION. NORMALIZATION SKIPPED ERRORS FRO C2705300
     *EABLY FOLLOW ')                                                   C2705400
      RETURN                                                            C2705500
    1 A = DSQRT(A)                                                      C2705600
      CO 2  I=1,3                                                       C2705700
    2 V(I) = V(I)/A                                                     C2705800
      RETURN                                                            C2705900
      ENC                                                               C2706000
```

```
      SUBROUTINE MATMUL (A,B,C,N)                                       C2706100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2706200
      DIMENSION A(N,N),B(N,N),C(N,N)                                    C2706300
C     COMPUTS THE STANDARD MATRIX PRODUCT OF TWO NXN MATRICES           C2706400
C                   A*B = C                                             C2706500
      DO 1  I=1,N                                                       C2706600
      DO 1  J=1,N                                                       C2706700
      C(I,J) = 0                                                        C2706800
      DO 2  L=1,N                                                       C2706900
    2 C(I,J) = C(I,J) + A(I,L)*B(L,J)                                   C2707000
    1 CONTINUE                                                          C2707100
      RETURN                                                            C2707200
      END                                                               C2707300
```

```
      SUBROUTINE TRNSPS   (XMT)                                         C2707400
C     USED TO COMPUTE THE TRANSPOSE OF A 3X3 MATRIX                     C2707500
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2707600
      DIMENSION XMT(3,3),TEMP(3,3)                                      C2707700
      DO 1 I=1,3                                                        C2707800
      DO 1 J=1,3                                                        C2707900
    1 TEMP(I,J) = XMT(I,J)                                              C2708000
      DO 2 I=1,3                                                        C2708100
      DO 2 J=1,3                                                        C2708200
    2 XMT(I,J) = TEMP(J,I)                                              C2708300
      RETURN                                                            C2708400
      END                                                               C2708500
```

```
      SUBROUTINE ROT(A,C,JA,TME)                                        C2708600
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2708700
C     GENERAL EULER ANGLE ROTATION MATRIX                               C2708800
      DIMENSION TME(3,3),V(3),W(3)                                      C2708900
      S = C/DABS(C)                                                     C2709000
      DO 24 N=1,3                                                       C2709100
      W(N) = S                                                          C2709200
   24 V(N) = .0D0                                                       C2709300
      JJA = IABS(JA)                                                    C2709400
      V(JJA) = A*JA/JJA                                                 C2709500
      W(JJA) = 1.0D0                                                    C2709600
      TME(1,1) = DSQRT(1.D0 - V(2)**2 - V(3)**2)*W(1)                   C2709700
      TME(2,2) = DSQRT(1.D0 - V(1)**2 - V(3)**2)*W(2)                   C2709800
      TME(3,3) = DSQRT(1.D0 - V(1)**2 - V(2)**2)*W(3)                   C2709900
      TME(1,2) = -V(3)                                                  C2710000
      TME(2,1) =  V(3)                                                  C2710100
      TME(1,3) = V(2)                                                   C2710200
      TME(3,1) = -V(2)                                                  C2710300
      TME(2,3) = -V(1)                                                  C2710400
      TME(3,2) =  V(1)                                                  C2710500
      RETURN                                                            C2710600
      END                                                               C2710700
```

```
C                                                                       C2800000
      SUBROUTINE VECADD(V1,V2,S)                                        C2800100
C     ADDS VECTOR V1 TO V2 RESULT IN S                                  C2800200
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2800300
      DIMENSION V1(3),V2(3), S(3)                                       C2800400
      S(1) = V1(1) + V2(1)                                              C2800500
      S(2) = V1(2) + V2(2)                                              C2800600
      S(3) = V1(3) + V2(3)                                              C2800700
      RETURN                                                            C2800800
      END                                                               C2800900


      SUBROUTINE VECSUB(V1,V2,D)                                        C2801000
C     SUBTRACTS VECTORS V1-V2=D                                         C2801100
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2801200
      DIMENSION V1(3),V2(3),D(3)                                        C2801300
      D(1) = V1(1) - V2(1)                                              C2801400
      D(2) = V1(2) - V2(2)                                              C2801500
      D(3) = V1(3) - V2(3)                                              C2801600
      RETURN                                                            C2801700
      END                                                               C2801800


      SUBROUTINE SCLV(SC,V,P)                                           C2801900
C     SCALAR * VECTOR                                                   C2802000
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2802100
      DIMENSION V(3),P(3)                                               C2802200
      P(1) = SC*V(1)                                                    C2802300
      P(2) = SC*V(2)                                                    C2802400
      P(3) = SC*V(3)                                                    C2802500
      RETURN                                                            C2802600
      END                                                               C2802700


      SUBROUTINE VECDOT(V1,V2,D)                                        C2802800
C     VECTOR DOT PRODUCT                                                C2802900
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2803000
      DIMENSION V1(3),V2(3)                                             C2803100
      D = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)                       C2803200
      RETURN                                                            C2803300
      END                                                               C2803400


      SUBROUTINE VECROS (V1,V2,C)                                       C2803500
C     VECTOR CROSS PRODUCT  C = V1 X V2                                 C2803600
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2803700
      DIMENSION V1(3),V2(3),C(3)                                        C2803800
      C(1) = V1(2)*V2(3)- V1(3)*V2(2)                                   C2803900
```

```
      C(2) = V1(3)*V2(1)- V1(1)*V2(3)                                   C2804000
      C(3) = V1(1)*V2(2)- V1(2)*V2(1)                                   C2804100
      RETURN                                                            C2804200
      END                                                               C2804300



      SUBROUTINE TRIPVP(V1,V2,V)                                        C2804400
C                                                                       C2804500
C     COMPUTES STANDARD VECTOR TRIPLE PRODUCT                           C2804600
C                                                                       C2804700
C                  V = V1X(V1XV2)                                       C2804800
C                    = V1*(V1.V2) - V2*(V1.V1)                          C2804900
C                                                                       C2805000
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2805100
      DIMENSION V1(3),V2(3),V(3)                                        C2805200
      A = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)                       C2805300
      B = V1(1)*V1(1) + V1(2)*V1(2) + V1(3)*V1(3)                       C2805400
      V(1) = V1(1)*A - V2(1)*B                                          C2805500
      V(2) = V1(2)*A - V2(2)*B                                          C2805600
      V(3) = V1(3)*A - V2(3)*B                                          C2805700
      RETURN                                                            C2805800
      END                                                               C2805900



      SUBROUTINE DYADD(D1,D2,D)                                         C2806000
C     ADDS TWO DYADS                                                    C2806100
C           C = D1 + D2                                                 C2806200
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2806300
      DIMENSION D1(3,3), D2(3,3), D(3,3)                                C2806400
      DO 1  I=1,3                                                       C2806500
      DO 1  J=1,3                                                       C2806600
    1 D(I,J) = D1(I,J) + D2(I,J)                                        C2806700
      RETURN                                                            C2806800
      END                                                               C2806900



      SUBROUTINE SCLD(A,D,T)                                            C2807000
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2807100
      DIMENSION D(3,3),T(3,3)                                           C2807200
C     MULTIPLY SCALAR BY A TENSOR                                       C2807300
      T(1,1) = A*D(1,1)                                                 C2807400
      T(2,1) = A*D(2,1)                                                 C2807500
      T(3,1) = A*D(3,1)                                                 C2807600
      T(1,2) = A*D(1,2)                                                 C2807700
      T(2,2) = A*D(2,2)                                                 C2807800
      T(3,2) = A*D(3,2)                                                 C2807900
      T(1,3) = A*D(1,3)                                                 C2808000
      T(2,3) = A*D(2,3)                                                 C2808100
      T(3,3) = A*D(3,3)                                                 C2808200
      RETURN                                                            C2808300
      END                                                               C2808400
```

```
      SUBROUTINE DYDOTV(A,V,D)                                          C2808500
C     SCALAR DOT PRODUCT OF DYAD AND VECTOR                             C2808600
C                 D = A.V                                               C2808700
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2808800
      DIMENSION D(3),A(3,3),V(3)                                        C2808900
      DO 1  I=1,3                                                       C2809000
      D(I) = 0                                                          C2809100
      DO 1  J=1,3                                                       C2809200
    1 D(I) = D(I) + A(I,J)*V(J)                                         C2809300
      RETURN                                                            C2809400
      END                                                               C2809500


      SUBROUTINE VXDYOV(V1,DY,V)                                        C2809600
C     COMPUTES   VECTOR X (DYAD . VECTOR)                               C2809700
C              V = V1 X (DY . V1)                                       C2809800
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2809900
      DIMENSION V1(3),V2(3),V(3),DY(3,3)                                C2810000
      DO 1 K=1,3                                                        C2810100
      V2(K) = 0.D0                                                      C2810200
      DO 1  J=1,3                                                       C2810300
    1 V2(K) = V2(K) + DY(K,J)*V1(J)                                     C2810400
      V(1)= V1(2)*V2(3) - V1(3)*V2(2)                                   C2810500
      V(2) = V1(3)*V2(1) - V1(1)*V2(3)                                  C2810600
      V(3) = V1(1)*V2(2) - V1(2)*V2(1)                                  C2810700
      RETURN                                                            C2810800
      END                                                               C2810900


      SUBROUTINE DYTOV (D,X1,X)                                         C2811000
C     USE TO TAKE SCALAR DOT PRODUCT OF TRANSPOSE OF                    C2811100
C      TENSOR D WITH VECTOR X1                                          C2811200
C     NEEDED SINCE TENSORS IN SYMMERIC MATRIX OF INERTIA TENSORS ARE IN C2811300
C       NON SYMMETRIC                                                   C2811400
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2811500
      DIMENSION D(3,3),X1(3),X(3)                                       C2811600
      DO 1  I=1,3                                                       C2811700
      X(I) = 0                                                          C2811800
      DO 1  J=1,3                                                       C2811900
      X(I) = X(I) + D(J,I)*X1(J)                                        C2812000
    1 CONTINUE                                                          C2812100
      RETURN                                                            C2812200
      END                                                               C2812300


      SUBROUTINE VDDYOV(V1,DY,V2,X)                                     C2812400
C     COMPUTES THE SCALAR TRIPLE PRODUCT                                C2812500
C                 VECTOR . (DYAD . VECTOR)                            C2812600
C                    V1.(DY.V2)                                       C2812700
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2812800
      DIMENSION V1(3),DY(3,3),V2(3),TEM(3)                              C2812900
```

```
      DO 1  K=1,3                                                       C2813000
      TEM(K) = 0.D0                                                     C2813100
      DO 1  J=1,3                                                       C2813200
    1 TEM(K) = TEM(K) + DY(K,J)*V2(J)                                   C2813300
      Y = C                                                             C2813400
      DO 2  J=1,3                                                       C2813500
    2 Y = Y + V1(J)*TEM(J)                                              C2813600
      RETURN                                                            C2813700
      END                                                               C2813800


      SUBROUTINE DYCP(V,D)                                              C2813900
C     TRANSFORMS VECTOR V1 INTO SKEW DYAD                               C2814000
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2814100
      DIMENSION  V(3),D(3,3)                                            C2814200
      D(1,1) = 0                                                        C2814300
      D(1,2) = V(3)                                                     C2814400
      D(1,3) = -V(2)                                                    C2814500
      D(2,1) = -V(3)                                                    C2814600
      D(2,2) = 0                                                        C2814700
      D(2,3) = V(1)                                                     C2814800
      D(3,1) = V(2)                                                     C2814900
      D(3,2) = -V(1)                                                    C2815000
      D(3,3) = 0                                                        C2815100
      RETURN                                                            C2815200
      END                                                               C2815300


      SUBROUTINE SUEOP(V1,V2,XM,D)                                      C2815400
      IMPLICIT REAL*8(A-H,O-Z,$)                                        C2815500
      DIMENSION V1(3),V2(3),D(3,3)                                      C2815600
C     USED TO COMPUTE THE PSUEDO INERTIA TENSOR                         C2815700
C      OF BODY LAMBA WITH RESPECT TO THE ORIGIN OF NEST K-1 AND         C2815800
C      THE HINGE POINT I-1 WHICH IS ON THE TOPOLOGICAL PATH FROM        C2815900
C      BODY 1 TO BODY LAMBA                                             C2816000
C          BLOCK 6 SUPPER GAMBA,SUE K-1,I-1 EQUATION 2-55 OF X-732-71-70 C2816100
C                      D = XM*((V1.V2)*1 - V2 V1)                       C2816200
C                                                                       C2816300
C        XM - SCALAR                                                    C2816400
C        V1 - VECTOR                                                    C2816500
C        V2 - VECTOR                                                    C2816600
C         1 - UNIT DYAD                                                 C2816700
C         * - SCALAR MULTIPLICATION                                     C2816800
C         . - VECTOR SCALAR MULTIPLICATION                              C2816900
C       BLANK - TENSOR MULTIPLICATION                                   C2817000
C      NOTE THAT IN GENERAL THE PSUEDO INERTIA TENSOR IS NON SYMMETRIC   C2817100
C                                                                       C2817200
      D(1,1) = XM*(V1(2)*V2(2) + V1(3)*V2(3))                           C2817300
      D(1,2) =-XM*V2(1)*V1(2)                                           C2817400
      D(1,3) =-XM*V2(1)*V1(3)                                           C2817500
      D(2,1) =-XM*V2(2)*V1(1)                                           C2817600
      D(2,2) = XM*(V2(1)*V1(1) + V2(3)*V1(3))                           C2817700
      D(2,3) =-XM*V2(2)*V1(3)                                           C2817800
      D(3,1) =-XM*V2(3)*V1(1)                                           C2817900
```

```
      D(3,2) =-XM*V2(3)*V1(2)                                          C2818000
      D(3,3) = XM*(V2(1)*V1(1) + V2(2)*V1(2))                          C2818100
      RETURN                                                           C2818200
      END                                                              C2818300


C                                                                      C2900000
      SUBROUTINE QUTMUL (Q1,Q2,P)                                      C2900100
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C2900200
      DIMENSION Q1(4),Q2(4),P(4)                                       C2900300
C     MULTIPLIES TWO QUATENIONS AND PUTS PRODUCT IN P                  C2900400
C             P=Q1*Q2                                                  C2900500
C     WHERE                                                            C2900600
C             * = QUATERNION MULTIPLICATION                            C2900700
      A0 = Q1(1)                                                       C2900800
      A1 = Q1(2)                                                       C2900900
      A2 = Q1(3)                                                       C2901000
      A3 = Q1(4)                                                       C2901100
      B0 = Q2(1)                                                       C2901200
      B1 = Q2(2)                                                       C2901300
      B2 = Q2(3)                                                       C2901400
      B3 = Q2(4)                                                       C2901500
      P(1) = A0*B0 - A1*B1 - A2*B2 - A3*B3                             C2901600
      P(2) = A0*B1 + A1*B0 + A2*B3 - A3*B2                             C2901700
      P(3) = A0*B2 - A1*B3 + A2*B0 + A3*B1                             C2901800
      P(4) = A0*B3 + A1*B2 - A2*B1 + A3*B0                             C2901900
      RETURN                                                           C2902000
      END                                                              C2902100


      SUBROUTINE QUATOP(QF,THA,ZT)                                     C2902200
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C2902300
      DIMENSION QF(3),ZT(4)                                            C2902400
C     COMPUTES ROTATION QUATERNION FROM EIGENVECTOR QF AND ROTATION    C2902500
C       ANGLE THA, EQUATION A-11 OF X-732-71-89                        C2902600
      ZT(1) = DCOS(THA/2)                                              C2902700
      S     = DSIN(THA/2)                                              C2902800
      DO 1  J=2,4                                                      C2902900
    1 ZT(J) = QF(J-1)*S                                                C2903000
      RETURN                                                           C2903100
      END                                                              C2903200


      SUBROUTINE TRANSO(GML,SL)                                        C2903300
      IMPLICIT REAL*8(A-H,O-Z,$)                                       C2903400
      DIMENSION GML(4),SL(3,3)                                         C2903500
C                                                                      C2903600
C     MATRIX OPERATOR CONSTRUCTS THE QUATERNION TRANSFORMATION MATRIX SL C2903700
C       FROM THE COMPONENTS OF THE QUATERNION GML(K)                   C2903800
C     THE EQUATION EVALUATE IS THE TRANSPOSE OF A-18 OF X-732-71-89    C2903900
C       THAT IS, LET                                                   C2904000
```

```
C                    GML(K) = (E0,E1,E2,E3)                                  C2904100
C        THEN                                                                C2904200
C                                                                            C2904300
C   E0**2+E1**2-E2**2-E3**2     2(E0*E3+E1*E2)          2(E1*E3-E0*E2)      C2904400
C      2(E1*E2-E0*E3)        E0**2-E1**2+E2**2-E3**2     2(E2*E3+E0*E1)      C2904500
C      2(E1*E3+E0*E2)           2(E2*E3-E0*E1)       E0**2-E1**2-E2**2+E3**2C2904600
C                                                                            C2904700
C                    SL(1,1)   SL(1,2)   SL(1,3)                             C2904800
C                =   SL(2,1)   SL(2,2)   SL(2,3)                             C2904900
C                    SL(3,1)   SL(3,2)   SL(3,3)                             C2905000
C                                                                            C2905100
C                                                                            C2905200
      E00 = GML(1)*GML(1)                                                    C2905300
      E01 = GML(1)*GML(2)                                                    C2905400
      E02 = GML(1)*GML(3)                                                    C2905500
      E03 = GML(1)*GML(4)                                                    C2905600
      E11 = GML(2)*GML(2)                                                    C2905700
      E12 = GML(2)*GML(3)                                                    C2905800
      E13 = GML(2)*GML(4)                                                    C2905900
      E22 = GML(3)*GML(3)                                                    C2906000
      E23 = GML(3)*GML(4)                                                    C2906100
      E33 = GML(4)*GML(4)                                                    C2906200
      SL(1,1) = E00 + E11 - E22 - E33                                        C2906300
      SL(1,2) = 2*(E03 + E12)                                                C2906400
      SL(1,3) = 2*(E13 - E02)                                                C2906500
      SL(2,1) = 2*(E12 - E03)                                                C2906600
      SL(2,2) = E00 - E11 + E22 - E33                                        C2906700
      SL(2,3) = 2*(E23 + E01)                                                C2906800
      SL(3,1) = 2*(E13 + E02)                                                C2906900
      SL(3,2) = 2*(E23 - E01)                                                C2907000
      SL(3,3) = E00 - E11 - E22 + E33                                        C2907100
      RETURN                                                                 C2907200
      END                                                                    C2907300
```

APPENDIX B

SUPPORT PROGRAMS FOR N-BOD2

FLEXIBLE BODY CAPABILITY

APPENDIX B

SUPPORT PROGRAMS FOR N-BOD2 FLEXIBLE BODY CAPABILITY

The flexible body options available in N-BOD2 require the user to obtain several resultant mode dependent parameters. These parameters will, for most applications, be computed from lumped parameter data obtained via the standard in-house finite element program. NASTRAN is the standard program at NASA/GSFC; however, other installations make use of other programs. Therefore, as a guide for the interested user, the following two programs are submitted and can either be used directly or be extensively modified to suit the situation existing at the particular computer installation that N-BOD2 will be applied.

The first program is actually the NASTRAN Executive Control Deck for a specially written DMAP program. This DMAP program will output all lumped parameter data necessary for the generation of the required N-BOD2 resultant mode dependent parameters. The program DISCOS referred to in the DMAP listing is another flexible body program being prepared by H. P. Frisch for release at GSFC. It requires similar data but in a completely different format.

This DMAP program is operational on the Macneal-Schwendler version 38 of NASTRAN and the Level 15.5 NASA version with one trivial change in the functional module GP4. (Built-in NASTRAN error messages will point out the change.)

The DMAP program was written so that, in addition to modes and frequencies (real), NASTRAN could be forced to output modal mass, stiffness, and damping matrices. Several print and punch options were also built in so that the user could see more clearly some of the resultant mass stiffness and damping matrices internally generated by NASTRAN. The interested user is referred to a NASTRAN Level 15 User's Manual (Reference 1) and Programmer's Manual (Reference 2) for a detailed description of the DMAP program. Basically it is a combination of rigid formats 3, 9, and 12.

```
  ID      FRISCH, DISCOS N-BOD2 NASTRAN
  DIAG    14
TIME    6000
  APP     DMAP
  BEGIN   DISCOS AND N-BOD2 DATA VIA NASTRAN $
  FILE    $
  GP1     GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/ C,N,123/
          V,N,NOGPDT $
  SAVE    LUSET,NOGPDT $
  CHKPNT  GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL  $
  PARAM   //C,N,NOP/V,Y,PRTGD=1 $
  COND    LBL10,PRTGD $
  TABPT      GPL,EQEXIN,GPDT,BGPDT,SIL// $
  TABPRT  GPL//C,N,GPL $
  TABPRT  GPDT//C,N,GPDT $
  TABPRT  BGPDT//C,N,BGPDT $
  LABEL   LBL10 $
  PURGE   USET,GM,GO,KAA,BAA,MAA,K4AA,PST,KFS,QP,EST/NOGPDT $
  CHKPNT  USET,GM,GO,KAA,BAA,MAA,K4AA,PST,KFS,QP,EST  $
  GP2     GEOM2,EQEXIN/ECT $
  CHKPNT  ECT $
  TA1,    ,ECT,EPT,BGPDT,SIL,,CSTM/EST,,GEI,ECPT,GPCT/V,N,LUSET/ C,N,
          123/V,N,NOSIMP/C,N,0/V,N,NOGENL/V,N,GENEL $
  SAVE    NOGENL,NOSIMP,GENEL $
  CHKPNT  EST,GEI,ECPT,GPCT $
  PURGE   K4GG,GPST,OGPST,MGG,BGG, K4NN,K4FF,K4AA,MNN,MFF,MAA,BNN,BFF,
          BAA,KGGX/NOSIMP/ OGPST/GENEL $
  CHKPNT  K4GG,GPST,OGPST,MGG,BGG,K4NN,K4FF,K4AA,MNN,MFF,MAA,BNN,BFF,
          BAA,KGGX  $
  SMA1    CSTM,MPT,ECPT,GPCT, DIT/KGGX,K4GG,GPST/V,N,NOGENL/ V,N,NOK4GG$
  SAVE    NOK4GG $
  CHKPNT  KGGX,K4GG,GPST $
  PARAM   //C,N,NOP/V,Y,PRTKG=-1 $
  COND    LBL14,PRTKG $
  MATPRN  KGGX,,,,// $
  LABEL   LBL14 $
  PURGE   K4NN,K4GG,K4FF,K4AA/NOK4GG $
  CHKPNT  K4NN,K4GG,K4FF,K4AA  $
  SMA2    CSTM,MPT,ECPT,GPCT,DIT/MGG,BGG/V,Y,WTMASS=1.0/V,N,NOMGG/ V,N,
          NOBGG/V,Y,COUPMASS=-1 $
  SAVE    NOMGG,NOBGG $
  CHKPNT  MGG,BGG  $
  PARAM   //C,N,NOP/V,Y,PRTMG=-1  $
  COND    LBL11,PRTMG $
  MATPRN  MGG,,,,// $
  LABEL   LBL11 $
  PARAM   //C,N,NOP/V,Y,PRTMGD=1 $
  COND    LBL15,PRTMGD $
  DIAGONAL MGG/MMG/C,Y,OPT=COLUMN/V,Y,POWER=1. $
  MATPRN  MMG,,,,// $
  LABEL   LBL15 $
  PURGE   BNN,BFF,BAA,BGG/NOBGG $
  CHKPNT  BNN,BFF,BAA,BGG $
  COND    LBL1,GRDPNT $
  GPWG    BGPDT,CSTM,EQEXIN,MGG/OGPWG/V,Y,GRDPNT=-1/V,Y,WTMASS $
  CHKPNT  OGPWG  $
  OFP     OGPWG,,,,,//V,N,CARDNO $
  SAVE    CARDNO $
  LABEL   LBL1 $
  PARAM   //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 $
```

```
GP4     CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPDT,CSTM/RG,,USET,/V,N,LUSET/
        V,N,MPCF1/V,N,MPCF2/V,N,SINGLE/V,N,OMIT/V,N,REACT/V,N,NSKIP/
        V,N,REPEAT/V,N,NOSET/V,N,NOL/V,N,NOA/C,N,0 $
SAVE    MPCF1,SINGLE,OMIT,NOSET,REACT,MPCF2,NSKIP,REPEAT,NOL,NOA $
CHKPNT  RG,USET $
PURGE   GM,GMD/MPCF1/GO,KOOB,LOO,UOO,MOOB,MOAB,GOD/OMIT/KFS,PST,OP/
        SINGLE $
CHKPNT  GM,GMD,GO,KOOB,LOO,UOO,MOOB,MOAB,GOD,KFS,PST,OP $
EQUIV   KGGX,KNN/MPCF1/MGG,MNN/MPCF1/ BGG,BNN/MPCF1/K4GG,K4NN/MPCF1 $
CHKPNT  KNN,MNN,BNN,K4NN,OGPST $
GPSP    GPL,GPST,USET,SIL/OGPST $
OFP     OGPST,,,,,//V,N,CARDNO $
SAVE    CARDNO $
COND    LBL2,MPCF1 $
MCE1    USET,RG/GM $
MCE2    USET,GM,KGGX,MGG,BGG,K4GG/KNN,MNN,BNN,K4NN $
CHKPNT  GM,KNN,MNN,BNN,K4NN $
LABEL   LBL2 $
EQUIV   KNN,KFF/SINGLE/MNN,MFF/SINGLE/BNN,BFF/SINGLE/K4NN,K4FF/SINGLE $
CHKPNT  KFF,MFF,BFF,K4FF  $
COND    LBL3,SINGLE $
SCE1    USET,KNN,MNN,BNN,K4NN/KFF,KFS, ,MFF,BFF,K4FF $
LABEL   LBL3 $
EQUIV   KFF,KAA/OMIT/ MFF,MAA/OMIT/BFF,BAA/OMIT/K4FF,K4AA/OMIT $
CHKPNT  KAA,MAA,BAA,K4AA  $
COND    LBL20,OMIT $
SMP1    USET,KFF,,BFF,K4FF/GO,KAA,KOOB,LOO,UOO,,,,BAA,K4AA $
CHKPNT  GO,KAA,KOOB,LOO,UOO,BAA,K4AA $
SMP2    USET,GO,MFF/MAA $
CHKPNT  MAA $
LABEL   LBL20 $
PURGE   KRR,KLR,DM,MLR,MR/REACT/CM/MPCF1/GO/OMIT/KFS/SINGLE/OG/NOSET $
COND    LBL21,REACT $
RBMG1   USET,KAA,MAA/KLL,KLR,KRR,MLL,MLR,MRR $
CHKPNT  KLL,KLR,KRR,MLL,MLR,MRR $
RBMG2   KLL/LLL,ULL $
CHKPNT  ULL,LLL $
RBMG3   LLL,ULL,KLR,KRR/DM $
CHKPNT  DM $
RBMG4   DM,MLL,MLR,MRR/MR $
CHKPNT  MR $
LABEL   LBL21 $
DPD     DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,,,,,,EED,EQDYN/V,N,
        LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NODLT/V,N,NOPSDL/V,N,NOFRL/V,N,
        NOMLFT/V,N,NOTRL/V,N,NOEED/C,N,123/V,N,NOUE $
SAVE    NOEED,NOUE $
CHKPNT  GPLD,SILD,USETD,EED,EQDYN $
COND    ERR02,NOEED
READ    KAA,MAA,MR,DM,EED,USET,CASECC/LAMA,PHIA,MI,OEIGS/C,N,MODES/V,
        N,NEIGV $
SAVE    NEIGV$
CHKPNT  LAMA,PHIA,MI,OEIGS $
ADD     MI,/OI/C,N,(1.-10,0.0) $
OFP     LAMA,OEIGS,,,,//V,N,CARDNO $
COND    FINIS,NEIGV $
SDR1    USET,,PHIA,,,GO,GM,,KFS,,/PHIG,,OG/C,N,1/C,N,REIG $
CHKPNT  PHIG,OG  $
PARAM   //C,N,NOP/V,Y,PRTMD=1 $
COND    LBL12,PRTMD $
MATPRN     PHIG,,,,// $
```

```
  LABEL   LBL12 $
  SDR2    CASECC,CSTM,MPT,DIT,EQEXIN,SIL,,,BGPDT,LAMA,OG,PHIG,EST,/,OQG1,
          OPHIG,OES1,OEF1,PPHIG/C,N,REIG $
  CHKPNT  OQG1,OPHIG,OES1,OEF1,PPHIG $
  OFP     OPHIG,OQG1,OEF1,OES1,,//V,N,CARDNO $
  SAVE    CARDNO $
  EQUIV   GO,GOD/NOUE/GM,GMD/NOUE $
  CHKPNT  GOD,GMD $
  MTRXIN  CASECC,MATPOOL,EQDYN,,/K2PP,M2PP,B2PP/V,N,LUSETD/V,N,NOK2PP/V,
          N,NOM2PP/V,N,NOB2PP $
  SAVE    NOK2PP,NOM2PP,NOB2PP $
  CHKPNT  K2PP,M2PP,B2PP $
  PARAM   //C,N,AND/V,N,KDEKA/V,N,NOUE/V,N,NOK2PP $
  PARAM   //C,N,AND/V,N,MDEMA/V,N,NOUE/V,N,NOM2PP $
  PARAM   //C,N,AND/V,N,KDEK2/V,N,NOGENL/V,N,NOSIMP $
  PURGE   K2DD/NOK2PP/M2DD/NOM2PP/B2DD/NOB2PP $
  EQUIV   M2PP,M2DD/NOA/B2PP,B2DD/NOA/K2PP,K2DD/NOA/MAA,MDD/MDEMA/ KAA,
          KDD/KDEKA
  GKAD    USETD,GM,GO,KAA,BAA,MAA,K4AA,K2PP,M2PP,B2PP/KDD,BDD,MDD,GMD,
          GOD,K2DD,M2DD,B2DD/C,N,TRANRESP/C,N,DISP/C,N,DIRECT/C,Y,G=0.0/
          C,Y,W3=0.0/C,Y,W4=0.0/V,N,NOK2PP/V,N,NOM2PP/V,N,NOB2PP/ V,N,
          MPCF1/V,N,SINGLE/V,N,OMIT/V,N,NOUE/V,N,NOK4GG/V,N,NOBGG/ V,N,
          KDEK2/C,N,-1 $
  CHKPNT  M2DD,B2DD,K2DD,MDD,KDD,BDD,GMD,GOD $
  GKAM    USETD,PHIA,OI,LAMA,DIT,MDD,BDD,KDD,CASECC/MHH,BHH,KHH,
          PHIDH/V,N,NOUE/C,Y,LMODES=0/C,Y,LFREQ=0.0/ C,Y,HFREQ=0.0/C,N,
          1/C,N,1/C,N,1/V,N,NONCPU/V,N,FMODE $
  CHKPNT  MHH,BHH,KHH,PHIDH  $
  PARAM   //C,N,NOP/V,Y,PRTMM=1 $
  COND    LBL13,PRTMM $
  MATPRN  KHH,BHH,MHH,PHIDH,// $
  LABEL   LBL13 $
  PARAM  //C,N,NOP/V,Y,PCHGD=-1 $
  COND    LBL6,PCHGD $
  TABPCH  GPL,BGPDT,,,//C,N,GP/C,N,BG $
  LABEL   LBL6 $
  PARAM  //C,N,NOP/V,Y,PCHMG=-1 $
  COND    LBL7,PCHMG $
  OUTPUT3 MGG,,,,//C,N,0/C,Y,N1=MMG $
  LABEL   LBL7 $
  PARAM  //C,N,NOP/V,Y,PCHMD=-1 $
  COND    LBL8,PCHMD $
  OUTPUT3 PHIG,,,,//C,N,0/C,Y,N1=PHG $
  LABEL   LBL8 $
  PARAM  //C,N,NOP/V,Y,PCHMM=-1 $
  COND    LBL9,PCHMM $
  OUTPUT3 KHH,BHH,MHH,,//C,N,0/C,Y,N1=KKH/C,Y,N2=BBH/C,Y,N3=MMH $
  LABEL   LBL9 $
  JUMP    FINIS$
  LABEL   ERR02 $
  PRTPARM //C,N,-2/C,N,MODES $
  LABEL   FINIS $
  END     $
CEND
```

NASTRAN is extremely inflexible in allowing the user control over output data formatting. Consequently, a preprocessor program must be available to read the NASTRAN generated data. The next program is designed to not only read the data but also to process the data and compute the input data required by both N-BOD2 and DISCOS in respectively acceptable format.

For N-BOD2 users, the resultant mode dependent parameters are outputted on the line printer and annotated with the same acronyms used in the coding of the read data statements in subroutine INOPT.

```
C
C
C     DISCOS, N-BOD2 PREPROCESSOR OF NASTRAN GENERATED DATA
C
C        PURPOSE:
C                 1) READ ONE OR MORE NASTRAN GENERATED TAPES
C                 2) WRITE ONE INPUT TAPE FOR DISCOS PROGRAM
C                     CONTAINING FLEXIBLE BODY DATA FOR ALL
C                     FLEXIBLE BODIES IN SIMULATION
C                 3) COMPUTE RESULTANT MODE DEPENDENT PARAMETERS
C                     FOR GSFC MULTI-FLEXIBLE BODY PROGRAM
C                     N-BOD2, THIS SECTION OF NO INTEREST TO
C                     DISCOS USERS
C
C
C        DEFINITIONS:
C
C     DATA SET  1 = SCRATCH FILE USED TO PROCESS NASTRAN DATA
C
C     DATA SET  2 = DISCOS INPUT TAPE (THE OUTPUT TAPE OF THIS PROGRAM)
C        THE CARD OUTPUT OF THIS PROGRAM ASSUMES THAT DATA SET 14 IN
C        DISCOS WILL BE RESERVED FOR THE FLEXIBLE BODY DATA
C
C     DATA SET  3 = SCRATCH FILE USED TO REDUCE CORE REQUIREMENTS
C
C     DATA SET 11 = LOCATION IN SYSTEM OF NASTRAN TAPE ASSOCIATED WITH
C                   THE FLEXIBLE BODY IN DISCOS HAVING LOWEST MAGNITUDE
C                   INTEGER LABEL
C
C     DATA SET 12 = LOCATION IN SYSTEM OF NASTRAN TAPE ASSOCIATED WITH
C                   THE NEXT FLEXIBLE BODY IN THE DISCOS MODEL
C           .
C           .
C           .
C         ETC.      PROGRAM DIMENSIONED FOR A MAXIMUM OF 6 NASTRAN TAPES
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DATA IDL/1H$/
      DATA ISTR,IBLK/1H*,1H /
      DATA  IGPL,IBGP,IMGG     /4HGPL ,4HBGPD,4HMGG /
```

```
      DATA ID,IT,II/1HD,1HT,1HI/
      DATA IR,IE,IS/1HR,1HE,1HS/
C
      INTEGER  INPT(80), IGRID(500), IA(20), IB(20),
     *         ICOT(5),  IY(4)
      INTEGER MSAVE(100),MSTOTL,HINGE
C
      REAL*8   X(500),   Y(500),   Z(500),
     *         INERTA(6,6,500),    PHI(6,500,12),
     *         KHH(12,12),         BHH(12,12),
     *         MHH(12,12),         XX(4),          XY(4)
C
      REAL*8 AMD(500,72)
C
      COMMON /WORK/  AMD
C
      EQUIVALENCE (AMD(1,1),PHI(1,1,1))
      EQUIVALENCE (AMD(1,1),X(1))
      EQUIVALENCE (AMD(1,2),Y(1))
      EQUIVALENCE (AMD(1,3),Z(1))
      EQUIVALENCE (AMD(1,4),INERTA(1,1,1))
C
      LOGICAL  LECHO,LPUNCH,LCHECK,LNAST
      LOGICAL LRSTRT
C
C
  100 FORMAT (I5,3L5,2I5,L5)
  101 FORMAT (4A4,7I8,2A4)
  102 FORMAT (2A4,8I8)
  103 FORMAT (4A4,5I8,E16.9)
  104 FORMAT (2A4,2E16.9,I16,E16.9)
  105 FORMAT (2A4,2E16.9)
  106 FORMAT (I10)
  108 FORMAT (4A4,I8,2I16,E16.8)
  109 FORMAT (10E13.5)
  110 FORMAT (6E15.5)
  111 FORMAT (12I5)
  112 FORMAT ('1',I5,3L5,2I5,L5)
  113 FORMAT (1X,20I5)
  114 FORMAT ('1',5X,'******  MODAL DATA SELECTED FROM NASTRAN INPUT TAI
     *E',I4,' FOR PROCESSING  ******',//)
  115 FORMAT (12X,'MODE',I4,' LISTED BELOW IS MODE',I4,' ON THE NASTRAN
     *DATA TAPE')
  200 FORMAT (' NJOINT =',I5,' NUMBER OF GRID POINT LOCATIONS TO BE REAC
     * =',I5,' THESE SHOULD BE EQUAL')
  201 FORMAT ('      NUMBER    GRID POINT     X-LOCATION     Y-LOCATION
     * Z-LOCATION ')
  202 FORMAT (6X,I5,7X,I5,2X,3E15.5)
  300 FORMAT (80A1)
  302 FORMAT (4A4,I8,2I16,E16.8,2A4)
  303 FORMAT (2A4,4E16.8,2A4)
  400 FORMAT (2X,80A1)
  500 FORMAT (' ANOMOLOUS EOF MARK ON TAPE, LISTING FOLLOWS ')
  501 FORMAT (/,' INTERNAL GRID POINT ',I5,' EXTERNAL GRID POINT',I5,'
     *N THE NASTRAN BULK DATA DECK')
  502 FORMAT (' X,Y,Z COORDINATES')
  503 FORMAT (6E20.10)
  504 FORMAT (' LUMPED INERTIA MATRIX')
  505 FORMAT (' PHI (6 MODAL COMPONENTS FOR EACH SELECTED MODE)')
```

```
  506 FORMAT (//,' KHH ')
  507 FORMAT ('1',5X,'****** LUMPED PARAMETER AND MODAL DATA TO BE PROCE
     *SSED FOR DISCOS AND N-BOD2 INPUT DATA ******',///)
  508 FORMAT (//,' BHH ')
  509 FORMAT (//,' MHH ')
  510 FORMAT (/////)
  511 FORMAT (2X,'EXIT DECODE ',4(I8,E16.8))
  512 FORMAT (2X,'ENTER DECODE',2I8,4E16.8)
  600 FORMAT (I5,3E20.8)
C
C
C                    READ TWO INPUT CARDS PER NASTRAN TAPE
C                      NTAPE = DATA SET WHERE TAPE IS TO
C                              BE FOUND NTAPE=11,12,...
C                      LECHO = .TRUE. PRINT ECHO OF PROCESSED DATA
C                            = .FALSE. PRINT ONLY IF ERROR FOUND ON TAPE
C                      LPUNCH = .TRUE. PUNCH PLOT DATA
C                             = .FALSE. DON'T
C                      LCHECK = .TRUE. PRINT DATA IN/OUT OF DECODE
C                             = .FALSE. DON'T
C                      HINGE = GRID POINT NUMBER OF HINGE POINT AS
C                              DEFINED BY USER IN NASTRAN BULK DATA
C                      MSTOTL = TOTAL NUMBER OF FLEXIBLE BODY MODES TO
C                               PROCESS FOR DISCOS AND N-BOD2 DATA
C                      LNAST = .TRUE. PRINT NASTRAN DATA INPUT
C                              .FALSE. DON'T
C                      MSAVE  = MODE SAVE ARRAY
C                               MSAVE(K) = 0 SKIP MODE K DATA
C                                          K PROCESS MODE K DATA
C
      REWIND 2
    1 CONTINUE
      READ(5,100,END=4) NTAPE,LECHO,LPUNCH,LCHECK,HINGE,MSTOTL,LNAST
      WRITE(6,112)      NTAPE,LECHO,LPUNCH,LCHECK,HINGE,MSTOTL,LNAST
      READ(5,111) (IA(I),I=1,MSTOTL)
      DO 11 I=1,100
   11 MSAVE(I) = 0
      DO 12 I=1,MSTOTL
   12 MSAVE(IA(I)) = I
      WRITE(6,113) MSAVE
      REWIND NTAPE
C                        LRSTRT = .FALSE.  NO RESTART CARDS ON NTAPE
C                               = .TRUE.   RSTART CARDS ON NTAPE
      ICNTRS = 0
      LRSTRT = .FALSE.
    9 READ (NTAPE,300,END=8) INPT
      IF(LNAST) WRITE(6,400) INPT
C     CHECK FOR RESTART CARDS
      IF(IR.NE.INPT(1)) GO TO 9
      IF(IE.NE.INPT(2)) GO TO 9
      IF(IS.NE.INPT(3)) GO TO 9
      IF(IT.NE.INPT(4)) GO TO 9
C        COME HERE FOR RESTART CARDS, COUNT THEM AND SET FLAG
      LRSTRT = .TRUE.
      ICNTRS = ICNTRS+1
   13 READ (NTAPE,300,END=8) INPT
      IF(LNAST) WRITE(6,400) INPT
      IF(ID.EQ.INPT(1).AND.IT.EQ.INPT(2).AND.II.EQ.INPT(3)) GO TO 9
      ICNTRS =ICNTRS+1
      GO TO 13
```

```
    8 CONTINUE
      REWIND NTAPE
      REWIND 1
      REWIND 3
      ICNT = 0
      IBLK = 0
      DO 7 I=1,5
    7 ICOT(I) = 0
C
C               READ ALL DTI CARDS FROM NASTRAN TAPE NTAPE TO OBTAIN
C                   1) GRID POINT NUMBERING SEQUENCE SET UP BY NASTRAN
C                   2) GRID POINT LOCATIONS
C
C           **** NOTE ****
C
C               NASTRAN WILL RENUMBER GRID POINTS DEFINED BY USER
C               INTO A SEQUENCIAL SET, THIS IS USED FOR INTERNAL NASTRAN
C               COMPUTATION AND WILL BE USED BY DISCOS
C
C     SKIP RESTART CARDS IN FRONT OF GOOD DATA
      IF(.NOT.LRSTRT) GO TO 2
      DO 14 I=1,ICNTRS
   14 READ (NTAPE,300,END=3) INPT
    2 READ (NTAPE,101,END=3) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
C          CHECK FOR TABLE HEADING
      IF(IA(3).EQ.IGPL) GO TO 50
      IF(IA(3).EQ.IBGP) GO TO 60
      IF(IA(3).EQ.IMGG) GO TO 70
      GO TO 2
C
C       ERROR ON TAPE
    3 CONTINUE
      WRITE(6,500)
    6 REWIND NTAPE
    5 READ (NTAPE,300,END=4) INPT
      WRITE(6,400) INPT
      GO TO 5
    4 STOP
C
C                   COME HERE TO PROCESS GPL TABLE DATA
   50 CONTINUE
      ICNT = ICNT + 1
      IF(ICNT.NE.1) GO TO 51
      NJOINT = IB(2)
C                                 NJOINT = NUMBER OF GRID POINTS IN MODEL
      GO TO 2
   51 IF(ICNT.NE.2)GO TO 52
      DO 53 J=1,6
   53 IGRID(J) = IB(J+1)
      NC = (NJOINT-6)/8
      DO 54 J=1,NC
      JJ = 7 + 8*(J-1)
      JJ7 = JJ+7
   54 READ(NTAPE,102) (IA(I),I=1,2),(IGRID(I),I=JJ,JJ7)
      JJ = 6 + 8*NC
      JJ1 = JJ+1
      IF(JJ.EQ.NJOINT) GO TO 2
      READ(NTAPE,102) (IA(I),I=1,2),(IGRID(I),I=JJ1,NJOINT)
      GO TO 2
C                                 IGRID(I) =USER DEFINED GRID POINT NUMBER
```

```
C                                                  WHICH NASTRAN HAS RELABLED TO
C                                                  BE INTERNAL NUMBER I
   52 ICNT = 0
      NC = (NJOINT-3)/4
      IF(NJOINT.EQ.4*NC+3) GO TO 16
      NC = NC + 1
   16 DO 15 I=1,NC
   15 READ (NTAPE,300) INPT
      GO TO 2
C
C
C                  COME HERE TO PROCESS BGPDT CARDS
   60 CONTINUE
      IF(IB(2).EQ.NJOINT) GO TO 61
C     ERROR
      WRITE(6,200) NJOINT,IB(2)
      GO TO 6
   61 READ (NTAPE,300) INPT
      READ (NTAPE,103) (IA(I),I=1,4),(IB(I),I=1,5),X(1)
      DO 62 J=2,NJOINT
   62 READ (NTAPE,104) (IA(I),I=1,2),Y(J-1),Z(J-1),IB(1),X(J)
      J = NJOINT
      READ (NTAPE,105) (IA(I),I=1,2),Y(J),Z(J)
      GO TO 2
C        SAVE LOCATION DATA, WRITE IT ON TAPE 3 AFTER INERTIA DATA
C
C              X(J) = X-COORDINATE OF GRID POINT IGRID(J)
C              Y(J) = Y-COORDINATE OF GRID POINT IGRID(J)
C              Z(J) = Z-COORDINATE OF GRID POINT IGRID(J)
C
C
C                  COME HERE TO PROCESS ALL DMI CARDS
C        STEP 1:
C                 COUNT CARDS IN EACH GROUP
C                 ASCERTAIN FORMAT OF EACH CARD
C                 REWRITE ON SCRATCH FILE ALL CARDS
C                  PRECEEDED BY THEIR FORMAT CODE
C        STEP 2:
C                 REWIND SCRATCH FILE
C                 READ CARDS OFF SCRATCH FILE WITH KNOWN FORMAT
C                 LOAD ALL ARRAYS, SKIPPING ALL UNWANTED MODES
   70 CONTINUE
      REWIND 1
      IFORM = 301
      WRITE(1,106) IFORM
      WRITE(1,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      IBLK = 1
      ICOT(IBLK) = 1
      IF(IB(2).NE.6.OR.IB(6).NE.IB(7).OR.IB(6).NE.NJOINT*6) GO TO 71
      DO 76 I=1,NJOINT
      DO 76 J1=1,6
      DO 76 J2=1,6
   76 INERTA(J1,J2,I) =0.0
      GO TO 72
   71 CONTINUE
C     ERROR
      WRITE(6,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      GO TO 6
   72 READ(NTAPE,300,END=80) INPT
      IF(INPT(4).NE.ISTR) GO TO 73
```

```
      ICOT(IBLK) = ICOT(IBLK) + 1
      IFORM = 302
      GO TO 74
   73 IF(INPT(1).EQ.ISTR) GO TO 75
C                END OF DATA BLOCK IBLK
C        CHECK FOR 'END OF CHECKPOINT DICTIONARY'
      IF(INPT(1).EQ.IDL) GO TO 72
      IBLK = IBLK + 1
      ICOT(IBLK) = ICOT(IBLK) + 1
      IFORM = 301
      GO TO 74
   75 CALL INTERP(INPT,IFORM)
      ICOT(IBLK) = ICOT(IBLK) + 1
   74 WRITE(1,106) IFORM
      WRITE(1,300) INPT
      GO TO 72
   80 END FILE 1
C
C                    ALL DMI CARDS READ, FORMAT DEDUCED AND
C                    REWRITTEN ON SCRATCH FILE 1
C     FORMAT CODES:
C           301 = FORMAT (4A4,7I8,2A4)
C           302 = FORMAT (4A4,I8,2I16,E16.8,2A4)
C           303 = FORMAT (2A4,4E16.8,2A4)
C           304 = FORMAT (2A4,3E16.8,I16,2A4)
C           305 = FORMAT (2A4,2E16.8,I16,E16,8,2A4)
C           306 = FORMAT (2A4,E16.8,I16,2E16.8,2A4)
C           307 = FORMAT (2A4,E16.8,I16,E16.8,I16,2A4)
C           308 = FORMAT (2A4,I16.8,3E16.8,2A4)
C           309 = FORMAT (2A4,I16.8,2E16.8,I16.8,2A4)
C           310 = FORMAT (2A4,I16,E16.8,I16,E16.8,2A4)
C
C
C                          START TO PROCESS DMI CARDS
      REWIND 1
      IF(IBLK.LE.5) GO TO 20
C        ERROR IBLK MUST BE LESS THAN OR EQUAL TO 5
   81 REWIND 1
   83 READ (1,300,END=90) INPT
      WRITE(6,400) INPT
      GO TO 83
   90 STOP
   20 CONTINUE
      DO 22 IBK=1,IBLK
      ICBLK = ICOT(IBK)
      DO 21 JJ =1,ICBLK
      IF(JJ.NE.1) GO TO 23
C          READ FIRST CARD OF DATA BLOCK
      READ (1,106) IFORM
      READ (1,101) (IA(I),I=1,4),(IB(I),I=1,7),(IA(I),I=5,6)
      IF(IBK.NE.1) GO TO 40
C
C                    INERTIA MATRIX
      IF(IB(2).NE.6.OR.IB(6).NE.IB(7).OR.IB(6).NE.NJOINT*6) GO TO 81
C          SO FAR SO GOOD
      GO TO 21
   40 IF(IBK.NE.2) GO TO 41
C
C                    MODE SHAPES
      IF(IB(6).NE.6*NJOINT) GO TO 81
```

```
      NMODES = IB(7)
C                                  NMODES = NUMBER OF MODES
C                                  MSTOTL = NUMBER OF MODES TO PROCESS
      DO 42 I1=1,6
      DO 42 I2=1,NJOINT
      DO 42 I3=1,MSTOTL
   42 PHI(I1,I2,I3) = 0.0
      GO TO 21
C
C              MODAL MASS, STIFFNESS AND DAMPING MATRICES
   41 IF(IB(6).NE.IB(7).OR.IB(7).NE.NMODES) GO TO 71
      DO 43 I1=1,MSTOTL
      DO 43 I2=1,MSTOTL
      IF(IBK.NE.3) GO TO 44
      KHH(I1,I2) = 0.0
      GO TO 43
   44 IF(IBK.NE.4) GO TO 45
      BHH(I1,I2) = 0.0
      GO TO 43
   45 MHH(I1,I2) = 0.0
   43 CONTINUE
      GO TO 21
C
C                    READ IN DATA CARDS
   23 CONTINUE
      READ (1,106) IFORM
      IF(IFORM.NE.302) GO TO 24
      READ (1,302) (IA(I),I=1,4),IB(1),NNR,NNC,XX(1),(IA(I),I=5,6)
      NSR = MSAVE(NNR)
      NSC = MSAVE(NNC)
      IF(IBK.NE.1) GO TO 25
      N1T = (NNC-1)/6 + 1
      N2T = (NNR-1)/6 + 1
      IF(N1T.EQ.N2T) GO TO 26
C      ERROR
      WRITE(6,302) (IA(I),I=1,4),IB(1),NNR,NNC,XX(1),(IA(I),I=5,6)
      GO TO 81
   26 NT = N1T
      NC = MOD(NNC-1,6) + 1
      NR = MOD(NNR-1,6) + 1
      INERTA(NR,NC,NT) = XX(1)
      GO TO 21
   25 IF(IBK.NE.2) GO TO 27
      NT = (NNC-1)/6 + 1
      NE = MOD(NNC-1,6) + 1
      IF(NSR.EQ.0) GO TO 21
      PHI(NE,NT,NSR) = XX(1)
      GO TO 21
   27 IF(IBK.NE.3) GO TO 28
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      KHH(NSR,NSC) = XX(1)
      GO TO 21
   28 IF(IBK.NE.4) GO TO 29
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      BHH(NSR,NSC) = XX(1)
      GO TO 21
   29 CONTINUE
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 21
      MHH(NSR,NSC) = XX(1)
      GO TO 21
```

```
C
   24 CONTINUE
C          FORMAT 4E16.8 FOR XX WILL READ INTEGER I WHICH IS RIGHT
C          JUSTIFIED AS I * 1.0E-08
C          THIS IS EASILY UNDONE IN DECODE
      READ (1,303) (IA(I),I=1,2),(XX(I),I=1,4),(IA(I),I=3,4)
      IF(LCHECK) WRITE (6,512) IFORM,NNC,(XX(I),I=1,4)
      CALL DECODE(IFORM,XX,NNC,XY,IY,NY)
      IF(LCHECK) WRITE (6,511) (IY(I),XY(I),I=1,NY)
      DO 30 I=1,NY
      IF(IBK.NE.1) GO TO 31
      NC = MOD(IY(I)-1,6) + 1
      INERTA(NR,NC,NT) = XY(I)
      GO TO 30
   31 IF(IBK.NE.2) GO TO 32
      NT = (IY(I)-1)/6 + 1
      NE = MOD(IY(I)-1,6) + 1
      IF(NSR.EQ.0)  GO TO 30
      PHI(NE,NT,NSR) = XY(I)
      GO TO 30
   32 IF(IBK.NE.3) GO TO 33
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      KHH(NSR,NSC) = XY(I)
      GO TO 30
   33 IF(IBK.NE.4) GO TO 34
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      BHH(NSR,NSC) = XY(I)
      GO TO 30
   34 CONTINUE
      NSC = MSAVE(IY(I))
      IF(NSR.EQ.0.OR.NSC.EQ.0) GO TO 30
      MHH(NSR,NSC) = XY(I)
   30 CONTINUE
   21 CONTINUE
C
C
      IF(IBK.NE.1) GO TO 63
      REWIND 3
      WRITE (3) (((INERTA(I,J,N),I=1,6),J=1,6),N=1,NJOINT)
      WRITE (3) (X(N),Y(N),Z(N),N=1,NJOINT)
      IF(.NOT.LPUNCH) GO TO 10
      WRITE (7,600) (IGRID(N),X(N),Y(N),Z(N),N=1,NJOINT)
   10 CONTINUE
      WRITE(6,507)
      DO 64 N=1,NJOINT
      WRITE (6,501) N,IGRID(N)
      WRITE (6,502)
      WRITE (6,503) X(N),Y(N),Z(N)
      WRITE (6,504)
      WRITE (6,503) ((INERTA(I,J,N),J=1,6),I=1,6)
   64 CONTINUE
      WRITE (6,510)
      GO TO 22
   63 IF(IBK.NE.2) GO TO 65
      WRITE (3) (((PHI(I,N,J),I=1,6),J=1,MSTOTL),N=1,NJOINT)
      IF(.NOT.LPUNCH) GO TO 67
      DO 68 J=1,MSTOTL
      WRITE (7,600) (IGRID(N),(PHI(I,N,J),I=1,3),N=1,NJOINT)
```

```
   68 CONTINUE
   67 CONTINUE
      WRITE (6,114) NTAPE
      DO 17 I=1,NMODES
      IF(MSAVE(I).EQ.0) GO TO 17
      WRITE (6,115) MSAVE(I),I
   17 CONTINUE
      WRITE (6,510)
      DO 66 N=1,NJOINT
      WRITE (6,501) N,IGRID(N)
      WRITE (6,505)
      WRITE (6,503) ((PHI(I,N,J),I=1,6),J=1,MSTOTL)
   66 CONTINUE
      WRITE (6,510)
      GO TO 22
   65 IF(IBK.NE.3) GO TO 69
      WRITE (3) ((KHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,506)
      WRITE (6,503) ((KHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
      GO TO 22
   69 IF(IBK.NE.4) GO TO 77
      WRITE (3) ((BHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,508)
      WRITE (6,503) ((BHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
      GO TO 22
   77 IF(IBK.NE.5) GO TO 22
      WRITE (3) ((MHH(I,J),I=1,MSTOTL),J=1,MSTOTL)
      WRITE (6,509)
      WRITE (6,503) ((MHH(I,J),J=1,MSTOTL),I=1,MSTOTL)
      WRITE (6,510)
   22 CONTINUE
C
C                    DATA READY FOR DISCOS PROCESSING
      CALL DISCOS(IGRID,NJOINT,MSTOTL,ITAPE,LECHO)
      CALL NBOD2 (IGRID,NJOINT,MSTOTL,ITAPE,LECHO,HINGE)
      IF(ITAPE.LE.6) GO TO 1
      STOP
      END



      SUBROUTINE DISCOS(IGRID,NJ,NM,ICNT,LECHO)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      LOGICAL LECHO
C
      REAL*8   X(500),Y(500),Z(500),INERTA(6,6,500),IRUNNO
      REAL*8   PHI(6,500,12),KHH(12,12),BHH(12,12),LOC(6)
      REAL*8   AMS(6),JMASS(500,1),JINER(500,6),AIN(6),SMM(6)
      REAL*8   SMSM(500,3),LOCA(500,3),MOD(6,6),AMD(500,13),KH(6),BH(6)
      REAL*8 AMP(500,72)
      COMMON /WORK/  AMP
      EQUIVALENCE (AMP(1,1),X(1))
      EQUIVALENCE (AMP(1,2),Y(1))
      EQUIVALENCE (AMP(1,3),Z(1))
```

```
      EQUIVALENCE (AMP(1,4),INERTA(1,1,1))
      EQUIVALENCE (AMP(1,1),PHI(1,1,1))
C
      INTEGER  IGRID(500)
C
      COMMON /WORK2/ AMD
      EQUIVALENCE (AMD(1,1),JMASS(1,1)),     (AMD(1,1),JINER(1,1))
      EQUIVALENCE (AMD(1,7),SMSM(1,1)),      (AMD(1,10),LOCA(1,1))
      EQUIVALENCE (AMD(1,13),KHH(1,1)), (AMD(145,13),BHH(1,1))
C
      DATA      IFST/0/
      DATA      AMS/6HJMASS1,6HJMASS2,6HJMASS3,6HJMASS4,6HJMASS5,
     *             6HJMASS6/
      DATA     FLX/6HFLXDAT/
      DATA     AIN/6HINERT1,6HINERT2,6HINERT3,6HINERT4,6HINERT5,
     *             6HINERT6/
      DATA      SMM/6HSTMSM1,6HSTMSM2,6HSTMSM3,6HSTSMS4,6HSTMSM5,
     *             6HSTMSM6/
      DATA      LOC/6HLOCAT1,6HLOCAT2,6HLOCAT3,6HLOCAT4,6HLOCAT5,
     *             6HLOCAT6/
      DATA  MOD/6HXDIS1 ,6HYDIS1 ,6HZDIS1 ,6HTHA11 ,6HTHA21 ,6HTHA31 ,
     *          6HXDIS2 ,6HYDIS2 ,6HZDIS2 ,6HTHA12 ,6HTHA22 ,6HTHA32 ,
     *          6HXDIS3 ,6HYDIS3 ,6HZDIS3 ,6HTHA13 ,6HTHA23 ,6HTHA33 ,
     *          6HXDIS4 ,6HYDIS4 ,6HZDIS4 ,6HTHA14 ,6HTHA24 ,6HTHA34 ,
     *          6HXDIS5 ,6HYDIS5 ,6HZDIS5 ,6HTHA15 ,6HTHA25 ,6HTHA35 ,
     *          6HXDIS6 ,6HYDIS6 ,6HZDIS6 ,6HTHA16 ,6HTHA26 ,6HTHA36 /
      DATA   KH/6HKHH1  ,6HKHH2  ,6HKHH3  ,6HKHH4  ,6HKHH5  ,6HKHH6  /
      DATA   BH/6HBHH1  ,6HBHH2  ,6HBHH3  ,6HBHH4  ,6HBHH5  ,6HBHH6  /
      DATA IRUNNO/6HPREPRS/
C
  700 FORMAT (A6,'   0   14PREPRS ')
C
C               INITIALIZE TAPE, START COUNTER
      REWIND 3
      IF(IFST.NE.0) GO TO 1
      CALL INTAPE(2,FLX)
      ICNT = 0
      IFST = 1
    1 CONTINUE
      ICNT = ICNT+1
C
C                    PROCESS MASS MATRIX DATA
      READ  (3) (((INERTA(I,J,N),I=1,6),J=1,6),N=1,NJ)
      READ  (3) (X(N),Y(N),Z(N),N=1,NJ)
      DO 2  I=1,NJ
      IF(INERTA(1,1,I).NE.INERTA(2,2,I)) GO TO 3
      IF(INERTA(2,2,I).NE.INERTA(3,3,I)) GO TO 3
      IF(INERTA(3,3,I).LT.0) GO TO 3
      JMASS(I,1) = INERTA(3,3,I)
      GO TO 2
    3 WRITE(6,500) I,(INERTA(II,II,I),II=1,3)
  500 FORMAT (' ERROR MASS MATRIX POINT ',I5,' JMASS =',3E20.10)
      JMASS(I,1) = (INERTA(1,1,I)+INERTA(2,2,I)+INERTA(3,3,I))/3.0D0
    2 CONTINUE
C                    PUNCH READ MATRIX INPUT CARD
C                    PUT MASS MATRIX ON TAPE
      WRITE(7,700) AMS(ICNT)
      IF(LECHO) CALL WRITE(JMASS,NJ,1,AMS(ICNT),500)
      CALL WTAPE(JMASS,NJ,1,AMS(ICNT),500,2)
C
```

```
C                    PROCESS INERTIA DYAD DATA
      DO 4  I=1,NJ
      JINER(I,1) =  INERTA(4,4,I)
      JINER(I,2) =  INERTA(5,5,I)
      JINER(I,3) =  INERTA(6,6,I)
      JINER(I,4) =-INERTA(4,5,I)
      JINER(I,5) =-INERTA(4,6,I)
    4 JINER(I,6) =-INERTA(5,6,I)
      WRITE(7,700) AIN(ICNT)
      IF(LECHO) CALL WRITE(JINER,NJ,6,AIN(ICNT),500)
      CALL WTAPE(JINER,NJ,6,AIN(ICNT),500,2)
C
C                    PROCESS STATIC MASS MOMENT DATA
C                    PROCESS JOINT LOCATION DATA
      DO 5  I=1,NJ
      SMSM(I,1) =  INERTA(2,6,I)
      SMSM(I,2) = -INERTA(1,6,I)
      SMSM(I,3) =  INERTA(1,5,I)
      LOCA(I,1) =  X(I)
      LOCA(I,2) =  Y(I)
    5 LOCA(I,3) =  Z(I)
      WRITE(7,700)  SMM(ICNT)
      IF(LECHO) CALL WRITE(SMSM ,NJ,3,SMM(ICNT),500)
      CALL WTAPE(SMSM,NJ,3,SMM(ICNT),500,2)
      WRITE(7,700)  LOC(ICNT)
      IF(LECHO) CALL WRITE(LOCA ,NJ,3,LOC(ICNT),500)
      CALL WTAPE(LOCA,NJ,3,LOC(ICNT),500,2)
C
C                    PROCESS MODAL DATA
      READ  (3) (((PHI(I,N,J),I=1,6),J=1,NM),N=1,NJ)
      DO 6 K=1,6
      DO 7 I=1,NJ
      DO 7 M=1,NM
    7 AMD(I,M) = PHI(K,I,M)
      WRITE(7,700)  MOD(K,ICNT)
      IF(LECHO) CALL WRITE(AMD,NJ,NM,MOD(K,ICNT),500)
      CALL WTAPE(AMD,NJ,NM,MOD(K,ICNT),500,2)
    6 CONTINUE
C
C                    MODAL MATRICES
      READ  (3) ((KHH(I,J),I=1,NM),J=1,NM)
      READ  (3) ((BHH(I,J),I=1,NM),J=1,NM)
      WRITE(7,700)  KH(ICNT)
      IF(LECHO) CALL WRITE(KHH,NM,NM,KH(ICNT),12)
      CALL WTAPE(KHH,NM,NM,KH(ICNT),12,2)
      WRITE(7,700)  BH(ICNT)
      WRITE(6,100)
      IF(LECHO) CALL WRITE(BHH,NM,NM,BH(ICNT),12)
      CALL WTAPE(BHH,NM,NM,BH(ICNT),12,2)
C
      CALL LTAPE(2)
      CALL RTAPE(IRUNNO,AMS(ICNT),AMD,NJ,1,500,12,2)
      CALL WRITE(AMD,NJ,1,AMS(ICNT),500)
      CALL RTAPE(IRUNNO,AIN(ICNT),AMD,NJ,6,500,12,2)
      CALL WRITE(AMD,NJ,6,AIN(ICNT),500)
      CALL RTAPE(IRUNNO,SMM(ICNT),AMD,NJ,3,500,12,2)
      CALL WRITE(AMD,NJ,3,SMM(ICNT),500)
      CALL RTAPE(IRUNNO,LOC(ICNT),AMD,NJ,3,500,12,2)
      CALL WRITE(AMD,NJ,3,LOC(ICNT),500)
      DO 8 K=1,6
```

```
      CALL RTAPE(IRUNNO,MOD(K,ICNT),AMD,NJ,NM,500,12,2)
    8 CALL WRITE(AMD,NJ,NM,MOD(K,ICNT),500)
      CALL RTAPE(IRUNNO,KH(ICNT),KHH,NM,NM,12,12,2)
      CALL WRITE(KHH,NM,NM,KH(ICNT),12)
      CALL RTAPE(IRUNNO,BH(ICNT),BHH,NM,NM,12,12,2)
      CALL WRITE(BHH,NM,NM,BH(ICNT),12)
  100 FORMAT ('1',5X,'****** PROCESSED LUMPED PARAMETER AND MODAL DATA N
     *OW ON DATA SET 2 (DISCOS INPUT TAPE) ******',///)
      RETURN
      END




      SUBROUTINE NBOD2 (IGRID,NJ,NM,ICNT,LECHO,HINGE)
C     COMPUTE RESULTANT MODE DEPENDENT PARAMETERS FOR N-BOD2
      IMPLICIT REAL*8 (A-H,O-Z)
C
      LOGICAL LECHO
C
      COMMON /WORK/  AMP
      REAL*8 AMP(500,72)
      COMMON /WORK2/ AMD
      REAL*8 AMD(500,13)
C
      REAL*8   AMS(6),FLX,AIN(6),SMM(6),LOC(6),MOD(6,6),KH(6),BH(6)
      REAL*8   JMASS(500), JLOC(3,500), CMLOC(3), JINERT(3,3,500),
     *         PHIT(3,500,12), PHIR(3,500,12),  SMO(3,500),
     *         FLOM(12), ZETA(12), BDINER(3,3),
     *         FLA(3,12), FLB(3,12), FLC(3,12), FLD(3,3,12),
     *         FLJ(3,3,12),
     *         FCF(3,3,12,12), FCK(3,12,12),
     *         TEM(3), TEM1(3,3), ADM(500,12,6), TFM(3),
     *         KHH(12,12), BHH(12,12), IRUNNO
      INTEGER  IGRID(500)
      INTEGER  HINGE, INTH
C
      DATA     AMS/6HJMASS1,6HJMASS2,6HJMASS3,6HJMASS4,6HJMASS5,
     *           6HJMASS6/
      DATA    AIN/6HINERT1,6HINERT2,6HINERT3,6HINERT4,6HINERT5,
     *           6HINERT6/
      DATA     SMM/6HSTMSM1,6HSTMSM2,6HSTMSM3,6HSTSMS4,6HSTMSM5,
     *           6HSTMSM6/
      DATA     LOC/6HLOCAT1,6HLOCAT2,6HLOCAT3,6HLOCAT4,6HLOCAT5,
     *           6HLOCAT6/
      DATA  MOD/6HXDIS1 ,6HYDIS1 ,6HZDIS1 ,6HTHA11 ,6HTHA21 ,6HTHA31 ,
     *          6HXDIS2 ,6HYDIS2 ,6HZDIS2 ,6HTHA12 ,6HTHA22 ,6HTHA32 ,
     *          6HXDIS3 ,6HYDIS3 ,6HZDIS3 ,6HTHA13 ,6HTHA23 ,6HTHA33 ,
     *          6HXDIS4 ,6HYDIS4 ,6HZDIS4 ,6HTHA14 ,6HTHA24 ,6HTHA34 ,
     *          6HXDIS5 ,6HYDIS5 ,6HZDIS5 ,6HTHA15 ,6HTHA25 ,6HTHA35 ,
     *          6HXDIS6 ,6HYDIS6 ,6HZDIS6 ,6HTHA16 ,6HTHA26 ,6HTHA36 /
      DATA   KH/6HKHH1  ,6HKHH2  ,6HKHH3  ,6HKHH4  ,6HKHH5  ,6HKHH6  /
      DATA   BH/6HBHH1  ,6HBHH2  ,6HBHH3  ,6HBHH4  ,6HBHH5  ,6HBHH6  /
      DATA IRUNNO/6HPREPRS/
C
      EQUIVALENCE (AMP(1, 1),PHIT(1,1,1)),(AMP(1,37),PHIR(1,1,1)),
     *            (AMP(1, 1),ADM(1,1,1)),
     *            (AMP(1,13),ADM(1,1,2)),
```

```
     *             (AMP(1,25),ADM(1,1,3)),
     *             (AMP(1,37),ADM(1,1,4)),
     *             (AMP(1,49),ADM(1,1,5)),
     *             (AMP(1,61),ADM(1,1,6))
      EQUIVALENCE (AMD(1,1),JMASS(1)), (AMD(1,2),JLOC(1,1)),
     *            (AMD(1,5),SMO(1,1)), (AMD(1,5),JINERT(1,1,1))
C
C        FIND INTERNAL NASTRAN NUMBER ASSOCIATED
C        WITH THE HINGE POINT (GRID POINT 'HINGE')
      DO 1 N=1,NJ
      IF(IGRID(N).EQ.HINGE) GO TO 2
    1 CONTINUE
      WRITE(6,100) HINGE
      RETURN
    2 INTH = N
C
C                  INTH = INTERNAL NASTRAN NUMBER
C                         OF HINGE POINT
C
C         GET DATA OFF OF TAPE 2 TO COMPUTE CENTER OF MASS
C
      CALL RTAPE(IRUNNO,LOC(ICNT),AMP,NJ,3,500,72,2)
C
C              JLOC(I,N) = I-TH COORDINATE OF GRID POINT IGRID(N)
C                          VECTOR RELATIVE TO HINGE POINT HINGE
      DO 3 N=1,NJ
      DO 3 I=1,3
      JLOC(I,N) = AMP(N,I) - AMP(INTH,I)
    3 CONTINUE
C
      CALL RTAPE(IRUNNO,AMS(ICNT),AMP,NJ,1,500,72,2)
C
C             JMASS(N)  = LUMPED MASS AT GRID POINT IGRID(N)
      DO 4 N=1,NJ
      JMASS(N) = AMP(N,1)
    4 CONTINUE
C
      IF(.NOT.LECHO) GO TO 17
      WRITE(6,101)
      WRITE(6,102) (N,IGRID(N),N,(JLOC(I,N),I=1,3),N,JMASS(N),N=1,NJ)
   17 CONTINUE
C
C        COMPUTE CENTER OF MASS LOCATION
      TOTMAS = 0.0
      CMLOC(1) = 0.0
      CMLOC(2) = 0.0
      CMLOC(3) = 0.0
C
C         READ GRID POINT INERTIA TENSORS AND MASS MOMENTS
C
      CALL RTAPE(IRUNNO,SMM(ICNT),AMP,NJ,3,500,72,2)
      DO 11 N=1,NJ
```

```
      DO 11 I=1,3
      SMO(I,N) = AMP(N,I)
      IF(SMO(I,N).EQ.0.0) GO TO 11
      WRITE(6,107) (N,(SMO(I,N),I=1,3))
   11 CONTINUE
      IF(LECHO) CALL WRITE(SMO,3,NJ,6H SMO  ,3)
C
      CALL RTAPE(IRUNNO,AIN(ICNT),AMP,NJ,6,500,72,2)
      DO 10 N=1,NJ
      JINERT(1,1,N) =  AMP(N,1)
      JINERT(1,2,N) = -AMP(N,4)
      JINERT(1,3,N) = -AMP(N,5)
      JINERT(2,1,N) = -AMP(N,4)
      JINERT(2,2,N) =  AMP(N,2)
      JINERT(2,3,N) = -AMP(N,6)
      JINERT(3,1,N) = -AMP(N,5)
      JINERT(3,2,N) = -AMP(N,6)
      JINERT(3,3,N) =  AMP(N,3)
   10 CONTINUE
      IF(LECHO) CALL WRITE(JINERT,9,NJ,6HJINERT,9)
C
C
C
C     COMPUTE UNDEFORMED BODY INERTIA TENSOR
C       RELATIVE TO BODY CENTER OF MASS
C
      DO 20 I=1,3
      DO 20 J=1,3
   20 BDINER(I,J) = 0.0
      DO 19 N=1,NJ
      CALL VECSUB(JLOC(1,N),CMLOC,TEM)
      CALL SUEOP(TEM,TEM,JMASS(N),TEM1)
      CALL DYADD(BDINER,TEM1,BDINER)
      CALL DYADD(BDINER,JINERT(1,1,N),BDINER)
   19 CONTINUE
C
C
C          READ IN MODE SHAPES AND RESORT THEM
C
      DO 6 I=1,6
      CALL RTAPE(IRUNNO,MOD(I,ICNT),ADM(1,1,I),NJ,NM,500,12,2)
    6 CONTINUE
      REWIND 3
      WRITE (3) (((ADM(N,M,I),I=1,3),N=1,NJ),M=1,NM)
      WRITE (3) (((ADM(N,M,I),I=4,6),N=1,NJ),M=1,NM)
      REWIND 3
      READ  (3) (((PHIT(I,N,M),I=1,3),N=1,NJ),M=1,NM)
      READ  (3) (((PHIR(I,N,M),I=1,3),N=1,NJ),M=1,NM)
      REWIND 3
C
      IF(.NOT.LECHO) GO TO 18
      WRITE(6,104)
      DO 7 M=1,NM
      WRITE(6,105)
      WRITE(6,106)    (N,M,(PHIT(I,N,M),I=1,3),
     *                 N,M,(PHIR(I,N,M),I=1,3),N=1,NJ)
    7 CONTINUE
   18 CONTINUE
C
      WRITE(6,118)
```

```
      WRITE(6,103) TOTMAS
      WRITE(6,115)
      WRITE(6,119) (BDINER(1,J),J=1,3)
      WRITE(6,120) (BDINER(2,J),J=1,3)
      WRITE(6,121) (BDINER(3,J),J=1,3)
      WRITE(6,115)
      WRITE(6,122) (CMLOC(I),I=1,3)
      WRITE(6,115)
      WRITE (6,125) INTH, HINGE
      WRITE (6,115)
C
C          COMPUTE RESULTANT PARAMETERS
C
      DO 8 M=1,NM
      DO 8 I=1,3
      FLA(I,M) = 0.0
      FLB(I,M) = 0.0
      FLC(I,M) = 0.0
      DO 8 J=1,3
      FLJ(I,J,M) = 0.0
      FLD(I,J,M) = 0.0
    8 CONTINUE
      DO 9 M=1,NM
      DO 9 MM=1,NM
      DO 9 I=1,3
      FCK(I,M,MM) = 0.0
      DO 9 J=1,3
      FCF(I,J,M,MM) = 0.0
    9 CONTINUE
C
C
      AMASS = 1.0/TOTMAS
C
      DO 12 M=1,NM
      DO 13 J=1,NJ
C
      CALL SCLV(JMASS(J),PHIT(1,J,M),TEM)
      CALL VECADD(FLA(1,M),TEM,FLA(1,M))
C
      CALL VECSUB(JLOC(1,J),CMLOC,TEM)
      CALL SUEOP(PHIT(1,J,M),TEM,JMASS(J),TEM1)
      CALL DYADD(FLD(1,1,M),TEM1,FLD(1,1,M))
C
      CALL VECROS(JLOC(1,J),PHIT(1,J,M),TEM)
      CALL SCLV(JMASS(J),TEM,TEM)
      CALL VECADD(FLB(1,M),TEM,FLB(1,M))
C
      IF(JINERT(1,1,J).EQ.0.0.AND.
     *   JINERT(2,2,J).EQ.0.0.AND.
     *   JINERT(3,3,J).EQ.0.0) GO TO 13
C
      CALL DYDOTV(JINERT(1,1,J),PHIR(1,J,M),TEM)
      CALL VECADD(FLC(1,M),TEM,FLC(1,M))
C
      CALL VECXDY(PHIR(1,J,M),JINERT(1,1,J),TEM1)
      CALL DYADD(FLJ(1,1,M),TEM1,FLJ(1,1,M))
   13 CONTINUE
C
C        RESULTANT UNCOUPLED MODE DEPENDED PARAMETERS
C
```

```
      CALL SCLV(AMASS,FLA(1,M),FLA(1,M))
      CALL SCLV(AMASS,FLB(1,M),FLB(1,M))
   12 CONTINUE
C
C
C
C     READ MODAL STIFFNESS AND DAMPING DATA
C
      CALL RTAPE(IRUNNO,KH(ICNT),KHH,NM,NM,12,12,2)
      CALL RTAPE(IRUNNO,BH(ICNT),BHH,NM,NM,12,12,2)
C
      DO 16 M=1,NM
      WRITE (6,116) M
      FLOM(M) = DSQRT(KHH(M,M))
      ZETA(M) = BHH(M,M)/(2.0*FLOM(M))
      WRITE(6,115)
      WRITE(6,123) M,FLOM(M)
      WRITE(6,115)
      WRITE(6,124) M,ZETA(M)
      WRITE(6,115)
      WRITE(6,108)  (M,(FLA(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,109)  (M,(FLB(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,110)  (M,(FLC(I,M),I=1,3))
      WRITE(6,115)
      WRITE(6,126)   (M,(FLD(1,J,M),J=1,3))
      WRITE(6,111)   (M,(FLD(I,J,M),J=1,3),I=2,3)
      WRITE(6,115)
      WRITE(6,127)   (M,(FLJ(1,J,M),J=1,3))
      WRITE(6,112)   (M,(FLJ(I,J,M),J=1,3),I=2,3)
      WRITE(6,115)
   16 CONTINUE
      WRITE (6,105)
C
C        RESULTANT COUPLED MODE DEPENDENT PARAMETERS
C
      DO 14 NN=1,NM
      DO 14 MM=1,NM
C
      DO 15 J=1,NJ
      CALL SUEOP(PHIT(1,J,NN),PHIT(1,J,MM),JMASS(J),TEM1)
      CALL DYADD(FCF(1,1,MM,NN),TEM1,FCF(1,1,MM,NN))
C
      CALL DYDOTV(JINERT(1,1,J),PHIR(1,J,MM),TEM)
      CALL VECROS(TEM,PHIR(1,J,NN),TEM)
      HF = .5
      CALL SCLV(HF,TEM,TEM)
      CALL VECROS(PHIR(1,J,MM),PHIR(1,1,NN),TEM)
      CALL SCLV(JMASS(J),TEM,TEM)
      CALL VECADD(TEM,TEM,TEM)
      CALL VECADD(FCK(1,MM,NN),TEM,FCK(1,MM,NN))
   15 CONTINUE
      WRITE (6,117) MM,NN
      WRITE(6,115)
      WRITE(6,128)   (MM,NN,(FCF(1,J,MM,NN),J=1,3))
      WRITE(6,113)   (MM,NN,(FCF(I,J,MM,NN),J=1,3),I=2,3)
      WRITE(6,115)
      WRITE(6,114)   (MM,NN,(FCK(I,MM,NN),I=1,3))
   14 CONTINUE
```

```
C
  100 FORMAT (1X,'  ** ERROR **  CANNOT FIND HINGE POINT',I5,' IN GRID P
     *OINT TABLE, RETURN OUT OF SUBROUTINE NBOD2')
  101 FORMAT (////,'  GRID POINT LABELING, LOCATION AND MASS TABLES',//)
  102 FORMAT (' IGRID(',I4,') =',I5,9X,'JLOC(',I4,') =',3E15.7,9X,
     *        'JMASS(',I4,') =',E15.7)
    5 CONTINUE
      A = 1.0/TOTMAS
      CALL SCLV(A,CMLOC,CMLOC)
C
      DO 5 N=1,NJ
      TOTMAS = TOTMAS + JMASS(N)
      CALL SCLV(JMASS(N),JLOC(1,N),TEM)
      CALL VECADD(CMLOC,TEM,CMLOC)
    8 Y(I) = X(I+1)
      NY = 3
      NR = IY(3)
      RETURN
    7 IF(IFORM.NE.307) GO TO 9
      IY(1) = NR+1
      Y(1) = X(1)
      NY = 1
      NR = X(2)*FAC
      IF(NR.EQ.0) RETURN
      IY(2) = NR
      Y(2) = X(3)
      NY = 2
      NR = X(4)*FAC
      IF(NR.EQ.0) RETURN
      NR = NR-1
      RETURN
    9 IF(IFORM.NE.308) GO TO 10
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      DO 11 I=1,3
      IY(I) = NR-1+I
   11 Y(I)  = X(I+1)
      NY = 3
      NR = IY(3)
      RETURN
   10 IF(IFORM.NE.309) GO TO 12
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      DO 13 I=1,2
      IY(I) = NR-1+I
   13 Y(I)  = X(I+1)
      NY = 2
      NR = X(4)*FAC
      IF(NR.EQ.0) RETURN
      NR = NR-1
      RETURN
  103 FORMAT ('  XMAS =',E15.7,30X,'  (TOTAL MASS OF BODY)')
  104 FORMAT (////,'  MODE SHAPE TABLE (INTERNAL NUMBERING) ',//)
  105 FORMAT (///)
  106 FORMAT (' PHIT(',I4,',',I2,') =',3E15.7,5X,
     *        ' PHIR(',I4,',',I2,') =',3E15.7)
  107 FORMAT (///,' WARNING MASS MOMENT AT GRID POINT NON-ZERO, THAT IS
     * SMM ',I4,') =', 3E15.7,/,' THIS EFFECT IGNORED BY N-BOD2.  N-BOD2
     *.ASSUMES EACH GRID POINT AT ELEMENT MASS CENTER ')
```

```
  108 FORMAT ('  FLA(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND CENTER OF
     * MASS LOCATION AFTER DEFORMATION)')
  109 FORMAT ('  FLB(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND ANGULAR M
     *OMENTUM DUE TO DEFORMATION)')
  110 FORMAT ('  FLC(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND ANGULAR M
     *OMENTUM DUE TO DEFORMATION)')
  111 FORMAT ('  FLD(',I2,') = ',3E15.7)
  112 FORMAT ('  FLJ(',I2,') = ',3E15.7)
  113 FORMAT ('  FCF(',I2,',',I2,') =',3E15.7)
  114 FORMAT ('  FCK(',I2,',',I2,') =',3E15.7,' (COEFFICIENT FOR CORIOLI
     *S TORQUE DUE TO DEFORMATION)')
  115 FORMAT ('    ')
  116 FORMAT (/,3X,'FOR MODE',I3,' THE RESULTANT UNCOUPLED MODE DEPENDEN
     *T PARAMETERS REQUIRED FOR N-BOD2 INPUT ARE')
  117 FORMAT (/,3X,'FOR MODES',I3,' AND',I3,' THE RESULTANT MODE DEPENDE
     *NT CROSS COUPLING PARAMETERS REQUIRED FOR N-BOD2 INPUT ARE')
  118 FORMAT ('1',5X,'***** RESULTANT FLEXIBLE BODY DATA REQUIRED FOR IN
     *PUT TO N-BOD2 *****',///)
  119 FORMAT ('        ',3E15.7,'  (UNDEFORMED INERTIA TENSOR OF THE')
  120 FORMAT ('  XI    =',3E15.7,'   FLEXIBLE BODY IN BODY COORDINATES')
  121 FORMAT ('         ',3E15.7,'   RELATIVE TO BODY CENTER OF MASS)')
  122 FORMAT ('  CA    =',3E15.7,'  (CENTER OF MASS VECTOR, HINGE POINT T
     *O CM)')
  123 FORMAT ('  FLOM(',I2,') =',E15.7,' RAD/SEC  (MODAL FREQUENCY)')
  124 FORMAT ('  ZETA(',I2,') =',E15.7,'          (MODAL DAMPING ZETA)')
  125 FORMAT ('  CB    = VECTOR IN CONTIGUOUS BODY TO HINGE POINT DEFINED
     * AT INTERNAL GRID POINT',I5,' (EXTERNAL GRID POINT',I5,' IN BULK D
     *ATA)')
  126 FORMAT ('  FLD(',I2,') = ',3E15.7,' (COEFFICIENT TO FIND INERTIA T
     *ENSOR AFTER DEFORMATION)')
  127 FORMAT ('  FLJ(',I2,') = ',3E15.7,' (COEFFICIENT FOR CENTRIPITAL T
     *ORQUE DUE TO DEFORMATION)')
  128 FORMAT ('  FCF(',I2,',',I2,') =',3E15.7,' (COEFFICIENT FOR CENTRIP
     *ITAL TORQUE DUE TO DEFORMATION)')
      RETURN
      END



      SUBROUTINE INTERP(INPT,IMAT)
C
C       THIS ROUTINE SETS UP A CODE SEQUENCE WHICH WILL BE USED
C       TO READ MIXED INTEGER AND FLOATING POINT DATA FROM
C       THE DMI CARDS
C       BLANK FIELD ON LAST CARD WILL BE INTERPRETED AS AN INTEGER
C       BUT READ LATER IN AS ZERO, A CHECK FOR WHICH IS MADE
C
      INTEGER INPT(80)
      DATA IDOT,IEE/1H.,1HE/
      IF(INPT(12).EQ.IDOT.AND.INPT(21).EQ.IEE) GO TO 1
      IF(INPT(44).EQ.IDOT.AND.INPT(53).EQ.IEE) GO TO 2
      IMAT = 310
C                              310 - FORMAT  I,E,I,E
      RETURN
    2 IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 3
      IMAT = 309
C                              309 - FORMAT  I,E,E,I
```

```
      RETURN
    3 IMAT = 308
C                                 308 - FORMAT  I,E,E,E
      RETURN
    1 IF(INPT(28).EQ.IDOT.AND.INPT(37).EQ.IEE) GO TO 4
      IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 5
      IMAT = 307
C                                 307 - FORMAT  E,I,E,I  OR  E
      RETURN
    5 IMAT = 306
C                                 306 - FORMAT  E,I,E,E
      RETURN
    4 IF(INPT(44).EQ.IDOT.AND.INPT(53).EQ.IEE) GO TO 6
      IMAT = 305
C                                 305 - FORMAT  E,E,I,E  OR  E,E
      RETURN
    6 IF(INPT(60).EQ.IDOT.AND.INPT(69).EQ.IEE) GO TO 7
      IMAT = 304
C                                 304 - FORMAT  E,E,E,I  OR  E,E,E
      RETURN
    7 IMAT = 303
C                                 303 - FORMAT  E,E,E,E
      RETURN
      END



      SUBROUTINE DECODE(IFORM,X,NR,Y,IY,NY)
C
C***      ERROR IN TERMINOLOGY ELEMENTS COMING IN BY ROWS,
C           NR AND IY(I) REFER TO COLUMN NUMBER
C
C         OUTPUT OF DECODE
C              NY    = NUMBER OF REAL NUMBERS ON DMI CARD
C              IY(I) = ROW LOCATION FOR I-TH REAL NUMBER ON DMI CARD
C              Y(I)  = I-TH REAL NUMBER
C              NR    = NEXT REAL NUMBER TO BE PUT IN ROW NR+1 UNLESS
C                      FIELD 1 OF NEXT CARD CONTAINS AN INTEGER
C
      REAL*8 X(4), Y(4)
      INTEGER  IY(4)
      FAC = 1.00001D+08
      IF(IFORM.NE.303) GO TO 1
      DO 2 I=1,4
      IY(I) = NR+I
    2 Y(I)  = X(I)
      NY = 4
      NR = IY(4)
      RETURN
    1 IF(IFORM.NE.304) GO TO 3
      DO 4 I=1,3
      IY(I) = NR+I
    4 Y(I)  = X(I)
      NY = 3
      NR = X(4)*FAC-1
      RETURN
    3 IF(IFORM.NE.305) GO TO 5
```

```
      DO 6 I=1,2
      IY(I) = NR+I
    6 Y(I)  = X(I)
      NY = 2
      NR = X(3)*FAC
      IF(NR.EQ.0) RETURN
      IY(3) = NR
      Y(3) = X(4)
      NY = 3
      RETURN
    5 IF(IFORM.NE.306) GO TO 7
      IY(1) = NR+1
      Y(1) = X(1)
      NY = 1
      NR = X(2)*FAC
      IF(NR.EQ.0) RETURN
      DO 8 I=2,3
      IY(I) = NR-2+I
   12 IF(IFORM.NE.310) GO TO 14
      NR = X(1)*FAC
      IF(NR.EQ.0) GO TO 14
      IY(1) = NR
      Y(1) = X(2)
      NR = X(3)*FAC
      IF(NR.EQ.0) RETURN
      IY(2) = NR
      Y(2) = X(4)
      NY = 2
      RETURN
   14 WRITE(6,100) IFORM
  100 FORMAT (' IFORM =',I5,' ERROR')
      STOP
      END



      SUBROUTINE INTAPE (NTAPE,TAPEID)
      REAL*8 TAPEID,BUF,EOT
      DATA IZ1,BUF,EOT/1,0.D 0,3HEOT/
      DATA NOT / 6/
C
C  INITIALIZE TAPE FOR SUBROUTINE WTAPE.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE  = NUMBER OF TAPE. (E.G. 10).
C  TAPEID = TAPE IDENTIFICATION. (E.G. T1234). (A6 FORMAT).
C
 2001 FORMAT (//// 14H LOGICAL UNIT I2, 7H, TAPE A6,
     *             23H, HAS BEEN INITIALIZED.)
C
      REWIND NTAPE
      WRITE (NTAPE) TAPEID,IZ1,EOT,(BUF,I=1,16)
      REWIND NTAPE
```

```
      WRITE (NOT,2001) NTAPE,TAPEID
C
      RETURN
      END



      SUBROUTINE WTAPE  (A,NRA,NCA,ANAME,KR,NTAPE)
      REAL*8 A,ANAME,IRUNNO,DATE,
     *      BUF,EOT,DENSE,TAPEID,IEOTCK
      DIMENSION  A(KR,1)
      DATA IRUNNO,DATE/6HPREPRS,6HDATE  /
      DATA BUF,EOT,DENSE/0.D 0,3HEOT,5HDENSE/
      DATA NOT / 6/
C
C  WRITE MATRIX A ON TAPE.
C  INITIALIZE TAPE WITH SUBROUTINE INTAPE.
C  REWIND TAPE BEFORE FIRST USE OF THIS SUBROUTINE.
C  NOTE...THIS ROUTINE IS DESIGNED SPECIFICALLY FOR WRITING ON A DISK
C
C        (EG CDC-6400 DISK). USING THIS ROUTINE TO WRITE ON A PHYSICAL
C        TAPE DIRECTLY (IE WITHOUT USING THE DISK AS AN INTERMEDIARY)
C        WILL PROBABLY GIVE POOR RESULTS (DUE TO THE TOLERANCE
C        CHARACTERISTICS OF A TAPE DRIVE) AND SHOULD BE AVOIDED IF AT
C        ALL POSSIBLE.
C     ...THE CDC-6400 DISK IS AUTOMATICALLY ENDFILED AFTER EACH WRITE.
C  CODED BY   W A BENFIELD.  MARCH 1966.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C  REVISED BY  RF HRUDA.  NOVEMBER 1970.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  A       = MATRIX TO BE WRITTEN ON TAPE. SIZE(NRA,NCA).
C  NRA     = NUMBER OF ROWS OF MATRIX A.
C  NCA     = NUMBER OF COLS OF MATRIX A.
C  ANAME   = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR      = ROW DIMENSION OF A IN CALLING PROGRAM.
C  NTAPE   = NUMBER OF TAPE. (E.G. 10).
C
C     INTERNAL VARIABLES THAT ARE PUT ON TAPE (TRANSFERRED THRU COMMON).
C  IRUNNO IS RUN NUMBER OF PROBLEM. (A6 FORMAT).
C  DATE IS DATE. (A6 FORMAT).  FOR EXAMPLE  15FE65.
C
      IF (NRA .LT. 1 .OR. NCA .LT. 1) GO TO 999
C
C  SEARCH TAPE FOR END OF WRITTEN DATA.
   10 READ (NTAPE) TAPEID,LN,IEOTCK
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ (NTAPE)
      GO TO 10
C
C  END OF WRITTEN DATA HAS BEEN FOUND.
   20 BACKSPACE NTAPE
      WRITE (NTAPE) TAPEID,LN,BUF,IRUNNO,ANAME,NRA,NCA,DATE,DENSE,
     *              (BUF,I=1,10)
      WRITE (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      LN = LN + 1
      WRITE (NTAPE) TAPEID,LN,EOT,(BUF,I=1,16)
```

```
      BACKSPACE NTAPE
      RETURN
C
  999 WRITE (NOT,1000)
 1000 FORMAT (1H1,43HERROR IN SUBROUTINE WTAPE, PROGRAM STOPPED.)
      STOP
      END



      SUBROUTINE LTAPE (NTAPE)
      REAL*8 TAPEID,IRUNNO,ANAME,IEOTCK,DATE,ITYPE,ICHK,EOT,
     *    DENSE,SPARSE,SPART
          DATA NOT / 6/
          DATA EOT,DENSE,SPARSE,SPART/ 3HEOT,5HDENSE,6HSPARSE,5HSPART/
C
C  LIST HEADINGS OF MATRICES ON TAPE.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.   REVISED NOVEMBER 1970.
C  REVISED BY R A PHILIPPUS.  APRIL 1969.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE  = NUMBER OF TAPE. (E.G. 10).
C
 2001 FORMAT (//36X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6)
 2002 FORMAT (//30X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6,
     *        12H (CONTINUED))
 2003 FORMAT (27X69(1H-)/27X3HNO.3X7HRUN NO.4X4HNAME5X5HNROWS4X5HNCOLS4X
     *        4HDATE6X3HNNZ3X9HPARTITION/
     *          27X3H---3X6H------4X6H------4X5H-----
     *          4X5H-----3X6H------5X3H---3X9H--------- / )
 2004 FORMAT (25X15,3XA6,4XA6,3X15,4X15,4XA6,3X15,3X14,1H/I4)
 2005 FORMAT (/27X12HEND OF LIST.)
C
      REWIND NTAPE
      READ (NTAPE) TAPEID
      REWIND NTAPE
      L=0
C
   12 CONTINUE
      IF(L .EQ. 0)  WRITE (NOT,2001) NTAPE,TAPEID
      IF(L .NE. 0)  WRITE (NOT,2002) NTAPE,TAPEID
      WRITE (NOT,2003)
      NLINE=1
   13 L=L+1
      READ (NTAPE) TAPEID,LN,IEOTCK,IRUNNO,ANAME,NR,NC,DATE,ITYPE,NNZ,
     *             NP,NPT
      IF (L .EQ. 1)  ICHK = IRUNNO
      IF (ICHK .EQ. IRUNNO) GO TO 15
      NLINE=NLINE+1
      WRITE (NOT,2004)
      ICHK = IRUNNO
   15 IF (IEOTCK .EQ. EOT) GO TO 30
      READ  (NTAPE)
      IF (ITYPE .EQ.   DENSE ) WRITE (NOT,2004)
     *                         LN,IRUNNO,ANAME,NR,NC,DATE
```

```
      IF (ITYPE .EQ.   DENSE ) GO TO 20
      IF (ITYPE .EQ.   SPARSE) WRITE (NOT,2004)
     *                         LN,IRUNNO,ANAME,NR,NC,DATE,NNZ
      IF (ITYPE .EQ.   SPARSE) GO TO 20
      IF (ITYPE .EQ.   SPART ) WRITE (NOT,2004)
     *                         LN,IRUNNO,ANAME,NR,NC,DATE,NNZ,NP,NPT
      IF (ITYPE .EQ.   SPART ) GO TO 20
      WRITE (NOT,2004) LN,IRUNNO,ANAME,NR,NC,ITYPE
   20 NLINE=NLINE+1
      IF(NLINE.GT.43) GO TO 12
      GO TO 13
C
   30 WRITE (NOT,2004) LN,IEOTCK
      WRITE (NOT,2005)
      REWIND NTAPE
      RETURN
      END



      SUBROUTINE RTAPE  (IARUNO,IANAME, A,NRA,NCA, KR,KC,NTAPE)
      REAL*8 A,IARUNO,IANAME,TAPEID,IEOTCK,ITRUNO,ITNAME,
     *       DATE,ITYPE,DENSE,EOT
      DIMENSION A(KR,1)
      DATA NOT / 6 /
      DATA DENSE,EOT / 5HDENSE,3HEOT /
C
C  READ MATRIX A FROM TAPE BY IDENTIFICATION OF IARUNO,IANAME.
C  CALLS FORMA SUBROUTINES LTAPE,PAGEHD,ZZBOMB.
C  CODED BY WA BENFIELD.  JUNE 1966.
C  LAST REVISION BY R F HRUDA.  SEPTEMBER 1971.
C
C     SUBROUTINE ARGUMENTS
C  IARUNO = INPUT  RUN NUMBER OF MATRIX A. (A6 FORMAT).
C  IANAME = INPUT  MATRIX IDENTIFICATION.  (A6 FORMAT).
C  A      = OUTPUT MATRIX READ FROM TAPE. SIZE(NRA,NCA).
C  NRA    = OUTPUT NUMBER OF ROWS OF MATRIX A. WILL BE READ FROM TAPE.
C  NCA    = OUTPUT NUMBER OF COLS OF MATRIX A. WILL BE READ FROM TAPE.
C  KR     = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KC     = INPUT  COL DIMENSION OF A IN CALLING PROGRAM.
C  NTAPE  = INPUT  NUMBER OF TAPE. (E.G. 10).
C
 3001 FORMAT (29H1RTAPE CANNOT FIND   RUNNO = A6 /
     *        21X 8HANAME = A6 / 29X 6H------ )
C
      NTIME = 0
C  SEARCH TAPE FOR CORRECT HEADING.
    5 READ (NTAPE) TAPEID,LN,IEOTCK,ITRUNO,ITNAME,NRA,NCA,DATE,ITYPE,NNZ
      IF (ITRUNO .EQ. IARUNO .AND. ITNAME .EQ. IANAME) GO TO 10
      IF (IEOTCK .EQ.   EOT) GO TO 20
      READ  (NTAPE)
      GO TO 5
C
C  MATRIX HAS BEEN FOUND.
   10
```

```
                                                                NERROR=1
      IF (ITYPE .NE. DENSE .AND. NNZ .NE. 0) GO TO 999
                                                                NERROR=2
      IF (NRA.GT.KR .OR. NCA.GT.KC) GO TO 999
      READ (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
      RETURN
C
C  MATRIX CANNOT BE FOUND. SEARCH TAPE ONCE MORE.
   20 NTIME = NTIME+1
                                                                NERROR=3
      IF (NTIME .EQ. 2) GO TO 998
      REWIND NTAPE
      GO TO 5
C
  998 WRITE (NOT,3001) IARUNO,IANAME
  999 CALL LTAPE (NTAPE)
      WRITE (NOT,3002) NERROR
 3002 FORMAT (1H1,43HERROR IN SUBROUTINE RTAPE, PROGRAM STOPPED,
     *       10H NERROR = ,I3)
      STOP
      END




      SUBROUTINE WRITE (A,NR,NC,ANAME,KR)
      REAL*8 A,ANAME
      DIMENSION A(KR,1)
      DATA NOT / 6/
C
C  WRITE MATRIX OF REAL NUMBERS ON PAPER.
C  REQUIRES 123 COLUMN (MINIMUM) PRINTER.
C  UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RL WOHLEN.  DECEMBER 1968.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  A      = MATRIX TO BE PRINTED. SIZE(NR,NC).
C  NR     = NUMBER OF ROWS IN MATRIX A.
C  NC     = NUMBER OF COLS IN MATRIX A.
C  ANAME  = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR     = ROW DIMENSION OF A IN CALLING PROGRAM.
C
 2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
     *       10X,10(7X,1H( I2,1H))/)
 2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )
     *        3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)
 2030 FORMAT (1X,2I5,2X,1P10D11.3)
 2040 FORMAT (14HOEND OF WRITE.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      WRITE (NOT,2010) ANAME,NR,NC,(L,L=1,10)
      NLINE = 0
C
      DO 60 I=1,NR
      NZERO = 0
      JS = 1
   10 JE = JS+9
```

```
      IF (JE .GT. NC) JE=NC
C   SEE IF ELEMENTS ARE ZERO.
      DO 20 J=JS,JE
      IF (A(I,J) .NE. 0.D 0) GO TO 30
   20 CONTINUE
      GO TO 40
   30 NLINE = NLINE+1
      IF (NLINE .LE. 44) GO TO 35
      WRITE (NOT,2020) ANAME,NR,NC,(L,L=1,10)
      NLINE = 1
   35 WRITE (NOT,2030) I,JS,(A(I,J), J=JS,JE)
      NZERO = 1
   40 IF (JE .EQ. NC) GO TO 50
      JS = JS+10
      GO TO 10
C   SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS
C   AND SOMETHING HAS BEEN WRITTEN.
   50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR)  GO TO 60
      NLINE = NLINE+1
      WRITE (NOT,2030)
   60 CONTINUE
C
      WRITE (NOT,2040)
      RETURN
      END



      SUBROUTINE VECADD(V1,V2,S)
C     ADDS VECTOR V1 TO V2 RESULT IN S
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3), S(3)
      S(1) = V1(1) + V2(1)
      S(2) = V1(2) + V2(2)
      S(3) = V1(3) + V2(3)
      RETURN
      END



      SUBROUTINE VECSUB(V1,V2,D)
C     SUBTRACTS VECTORS V1-V2=D
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),D(3)
      D(1) = V1(1) - V2(1)
      D(2) = V1(2) - V2(2)
      D(3) = V1(3) - V2(3)
      RETURN
      END
```

```
      SUBROUTINE SCLV(SC,V,P)
C     SCALAR * VECTOR
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V(3),P(3)
      P(1) = SC*V(1)
      P(2) = SC*V(2)
      P(3) = SC*V(3)
      RETURN
      END



      SUBROUTINE VECDOT(V1,V2,D)
C     VECTOR DOT PRODUCT
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3)
      D = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
      RETURN
      END



      SUBROUTINE VECROS (V1,V2,C)
C     VECTOR CROSS PRODUCT  C = V1 X V2
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),C(3)
      C(1) = V1(2)*V2(3)- V1(3)*V2(2)
      C(2) = V1(3)*V2(1)- V1(1)*V2(3)
      C(3) = V1(1)*V2(2)- V1(2)*V2(1)
      RETURN
      END



      SUBROUTINE TRIPVP(V1,V2,V)
C
C      COMPUTES STANDARD VECTOR TRIPLE PRODUCT
C
C                  V = VIX(V1XV2)
C                    = V1*(V1.V2) - V2*(V1.V1)
C
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),V(3)
      A = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
      B = V1(1)*V1(1) + V1(2)*V1(2) + V1(3)*V1(3)
      V(1) = V1(1)*A - V2(1)*B
      V(2) = V1(2)*A - V2(2)*B
      V(3) = V1(3)*A - V2(3)*B
      RETURN
      END
```

```
      SUBROUTINE DYADD(D1,D2,D)
C     ADDS TWO DYADS
C             D = D1 + D2
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D1(3,3), D2(3,3), D(3,3)
      DO 1  I=1,3
      DO 1  J=1,3
    1 D(I,J) = D1(I,J) + D2(I,J)
      RETURN
      END



      SUBROUTINE SCLD(A,D,T)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3,3),T(3,3)
C     MULTIPLY SCALAR BY A TENSOR
      T(1,1) = A*D(1,1)
      T(2,1) = A*D(2,1)
      T(3,1) = A*D(3,1)
      T(1,2) = A*D(1,2)
      T(2,2) = A*D(2,2)
      T(3,2) = A*D(3,2)
      T(1,3) = A*D(1,3)
      T(2,3) = A*D(2,3)
      T(3,3) = A*D(3,3)
      RETURN
      END



      SUBROUTINE DYDOTV(A,V,D)
C     SCALAR DOT PRODUCT OF DYAD AND VECTOR
C                D = A.V
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3),A(3,3),V(3)
      DO 1  I=1,3
      D(I) = 0
      DO 1  J=1,3
    1 D(I) = D(I) + A(I,J)*V(J)
      RETURN
      END



      SUBROUTINE VXDYOV(V1,DY,V)
C     COMPUTES   VECTOR X (DYAD . VECTOR)
C             V = V1 X (DY . V1)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),V(3),DY(3,3)
      DO 1 K=1,3
```

```
      V2(K) = 0.D0
      DO 1  J=1,3
    1 V2(K) = V2(K) + DY(K,J)*V1(J)
      V(1)= V1(2)*V2(3) - V1(3)*V2(2)
      V(2) = V1(3)*V2(1) - V1(1)*V2(3)
      V(3) = V1(1)*V2(2) - V1(2)*V2(1)
      RETURN
      END



      SUBROUTINE DYTOV  (D,X1,X)
C     USE TO TAKE SCALAR DOT PRODUCT OF TRANSPOSE OF
C      TENSOR D WITH VECTOR X1
C     NEEDED SINCE TENSORS IN SYMMERIC MATRIX OF INERTIA TENSORS ARE IN
C        NON SYMMETRIC
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION D(3,3),X1(3),X(3)
      DO 1  I=1,3
      X(I) = 0
      DO 1  J=1,3
      X(I) = X(I) + D(J,I)*X1(J)
    1 CONTINUE
      RETURN
      END



      SUBROUTINE VODYOV(V1,DY,V2,X)
C      COMPUTES THE SCALAR TRIPLE PRODUCT
C                  VECTOR . (DYAD . VECTOR)
C                        V1.(DY.V2)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),DY(3,3),V2(3),TEM(3)
      DO 1  K=1,3
      TEM(K) = 0.D0
      DO 1  J=1,3
    1 TEM(K) = TEM(K) + DY(K,J)*V2(J)
      X = 0
      DO 2  J=1,3
    2 X = X + V1(J)*TEM(J)
      RETURN
      END



      SUBROUTINE DYOP(V,D)
C     TRANSFORMS VECTOR V1 INTO SKEW DYAD
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION  V(3),D(3,3)
      D(1,1) = 0
      D(1,2) = V(3)
```

```
      D(1,3) = -V(2)
      D(2,1) = -V(3)
      D(2,2) = 0
      D(2,3) = V(1)
      D(3,1) = V(2)
      D(3,2) = -V(1)
      D(3,3) = 0
      RETURN
      END



      SUBROUTINE SUEOP(V1,V2,XM,D)
      IMPLICIT REAL*8(A-H,O-Z,$)
      DIMENSION V1(3),V2(3),D(3,3)
C     USED TO COMPUTE THE PSUEDO INERTIA TENSOR
C      OF BODY LAMBA WITH RESPECT TO THE ORIGIN OF NEST K-1 AND
C      THE HINGE POINT I-1 WHICH IS ON THE TOPOLOGICAL PATH FROM
C      BODY 1 TO BODY LAMBA
C         BLOCK G SUPPER GAMBA,SUB K-1,I-1 EQUATION 2-55 OF X-732-71-70
C                   D = XM*((V1.V2)*1 - V2 V1)
C
C        XM - SCALAR
C        V1 - VECTOR
C        V2 - VECTOR
C         1 - UNIT DYAD
C         * - SCALAR MULTIPLICATION
C         . - VECTOR SCALAR MULTIPLICATION
C     BLANK - TENSOR MULTIPLICATION
C      NOTE THAT IN GENERAL THE PSUEDO INERTIA TENSOR IS NON SYMMETRIC
C
      D(1,1) = XM*(V1(2)*V2(2) + V1(3)*V2(3))
      D(1,2) =-XM*V2(1)*V1(2)
      D(1,3) =-XM*V2(1)*V1(3)
      D(2,1) =-XM*V2(2)*V1(1)
      D(2,2) = XM*(V2(1)*V1(1) + V2(3)*V1(3))
      D(2,3) =-XM*V2(2)*V1(3)
      D(3,1) =-XM*V2(3)*V1(1)
      D(3,2) =-XM*V2(3)*V1(2)
      D(3,3) = XM*(V2(1)*V1(1) + V2(2)*V1(2))
      RETURN
      END



      SUBROUTINE VECXDY(P,T,D)
      REAL*8  P(3),  T(3,3),  D(3,3)
C
C        COMPUTES VECTOR CROSS DYAD
C                   D = P X T
      D(1,1) = P(2)*T(3,1) - P(3)*T(2,1)
      D(1,2) = P(2)*T(3,2) - P(3)*T(2,2)
      D(1,3) = P(2)*T(3,3) - P(3)*T(2,3)
      D(2,1) = P(3)*T(1,1) - P(1)*T(3,1)
```

```
D(2,2) = P(3)*T(1,2) - P(1)*T(3,2)
D(2,3) = P(3)*T(1,3) - P(1)*T(3,3)
D(3,1) = P(1)*T(2,1) - P(2)*T(1,1)
D(3,2) = P(1)*T(2,2) - P(2)*T(1,2)
D(3,3) = P(1)*T(2,3) - P(2)*T(1,3)
RETURN
END
```

REFERENCES

1. McCormick, Caleb W., "The NASTRAN User's Manual," NASA Document SP-222, September 1970.

2. Douglas, Frank J., "The NASTRAN Programmer's Manual," NASA Document SP-223, September 1970.

BIBLIOGRAPHIC DATA SHEET

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-1145		

4. Title and Subtitle	5. Report Date
The N-BOD2 User's and Programmer's Manual	February 1978
	6. Performing Organization Code: 712

7. Author(s)	8. Performing Organization Report No.
Harold P. Frisch	G 7702 F-22

9. Performing Organization Name and Address	10. Work Unit No.
Goddard Space Flight Center Greenbelt, Maryland 20771	11. Contract or Grant No.
	13. Type of Report and Period Covered: Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A general purpose digital computer program has been developed entitled N-BOD2. The program is designed to aid in the analysis of spacecraft attitude dynamics; however it is applicable to a much broader class of problems. The program provides the analyst with the capability of automatically deriving and numerically solving the equations of motion of any system that can be modeled as a topological tree of coupled rigid bodies, flexible bodies, point masses, and symmetrical momentum wheels. Two modes of output are available; the composite system equations of motion may be outputted on a line printer in a symbolic form that may be easily translated into common vector-dyadic notation, or the composite system equations of motion may be solved numerically and any desirable set of system state-variables outputted as a function of time.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Spacecraft attitude dynamics, Stabilization and control, Rigid body dynamics, Flexible body dynamics, Applied mathematics	STAR Category 18 Unclassified--Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassifed	306	$11.75

*For sale by the National Technical Information Service, Springfield, Virginia 22161

*U.S. GOVERNMENT PRINTING OFFICE: 1978 - 735-078/77

National Aeronautics and Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

SPECIAL FOURTH CLASS MAIL
BOOK

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return